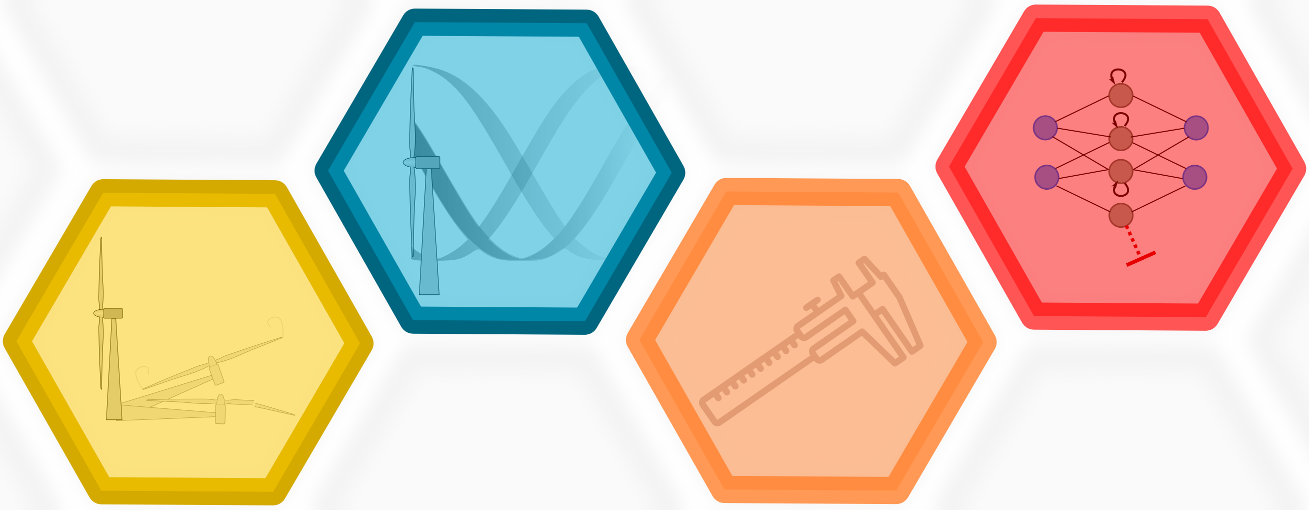
# Engineering Models and Data Science for Wind Energy



vorgelegt von
**MSc. MEng. Matthew Lennie**

an der Fakultät V – Verkehrs- und Maschinensysteme
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
- Dr.-Ing. -
genehmigte Dissertation

**Promotionsausschuss:**
**Vorsitzender:** Prof. Dr-Ing. J. Weiss
**Gutachter:** Prof. Dr-Ing. C. O. Paschereit
Prof. R. Martinuzzi
Associate Prof. A. Bianchini
Dr-Ing. G. Pechlivanoglou
**Tag der**
**wissenschaftlichen Aussprache:** 16.09.2019

Berlin. 2020

## Acknowledgments

To my family, friends, colleagues, mentors and to my wife Svenja.
Thank you for the support and companionship.

# Abstract

The development of wind turbines relies on solid numerical models. From the tips of the blades through to the design of wind parks, numerical models are crucial. Engineering models have been the solid back bone of wind turbine engineering. However, useful as these models are, they do face limitations. This study explored how a combination of engineering models and more recent data science/machine-learning-type models could help solve the increasingly complex problems associated with wind energy. This dissertation relies on three papers as examples of these concepts. The first paper shows how an engineering model (blade element momentum method) can be replaced with a free vortex lifting line theory method for floating platform simulations. Furthermore, the instantaneous aerodynamic damping of a rotor in fore-aft motion was derived. The second paper explores, through experimental work, the highly complex aerodynamics of wind turbines in standstill conditions. To address these complex aerodynamics, the third paper engages machine learning methods to highlight the cycle-to-cycle differences in dynamic stall. These three papers demonstrate where engineering models, medium-fidelity simulations and machine learning can be useful. This work makes recommendations for seven strategies to integrate the newest family of models - the machine learning models - into the wind energy system.

## Zusammenfassung

Die Entwicklung von Windkraftanlagen basiert auf zuverlässigen numerischen Modellen. Von der Spitze der Blätter bis zur Gestaltung von Windparks ist die Verwendung von numerische Modelle unersetzlich. So nützlich diese Modelle auch sind, sie haben ihre Grenzen. Die vorliegende Arbeit untersucht, wie eine Kombination aus vereinfachten Modellen und moderneren Modellen der Datenwissenschaft sowie des maschinellen Lernens dazu beitragen kann, die immer komplexeren Probleme der Windenergie zu lösen. Diese Arbeit basiert auf drei Papieren als Beispiele für dieses Konzept. Das erste Papier zeigt, wie die Blade Element Momentum Methode durch eine Free Vortex Lifting Line Theory Methode für schwimmende Plattformsimulationen ersetzt werden kann. Daraus wird die momentane aerodynamische Dämpfung eines Rotors in der Längsbewegung abgeleitet. Das zweite Papier untersucht, durch experimentelle Arbeiten, die hochkomplexe Aerodynamik von Windkraftanlagen im Stillstand. Um diese komplexe Aerodynamik zu verstehen, beschäftigt sich das dritte Papier mit maschinellen Lernmethoden um die Unterschiede zwischen den Zyklen im dynamischen Strömungsabriss aufzuzeigen. Alle drei Papiere zeigen, wo Ingenieurmodelle, Simulationen mittlerer Genauigkeit und maschinelles Lernen nützlich sein können. Abschließend gibt diese Arbeit Empfehlungen für sieben Strategien zur Integration der neuesten Modellfamilie - den Modellen des maschinellen Lernens - in die Windenergie.

# Contents

# List of Figures

# Chapter 1 | Wind Energy

# Chapter 1 | Wind Energy

## 1.1 Climate change in four pictures

The climate is changing [138]. The industrial age has brought wonders - running water, high quality housing, electricity, modern medicine, the Internet, and a great leap forward in the general well-being of humanity. However, these developments have come at a cost, and the planet has been extending its line of credit. As fossil fuels burn, the emissions blanket our planet thus trapping more heat [138]. Greenhouse gases, such as carbon dioxide, have always existed in the atmosphere for hundreds of millions of years, and their concentrations have varied. In more recent times, carbon dioxide concentrations have dramatically increased measurements since the industrial revolution [146].

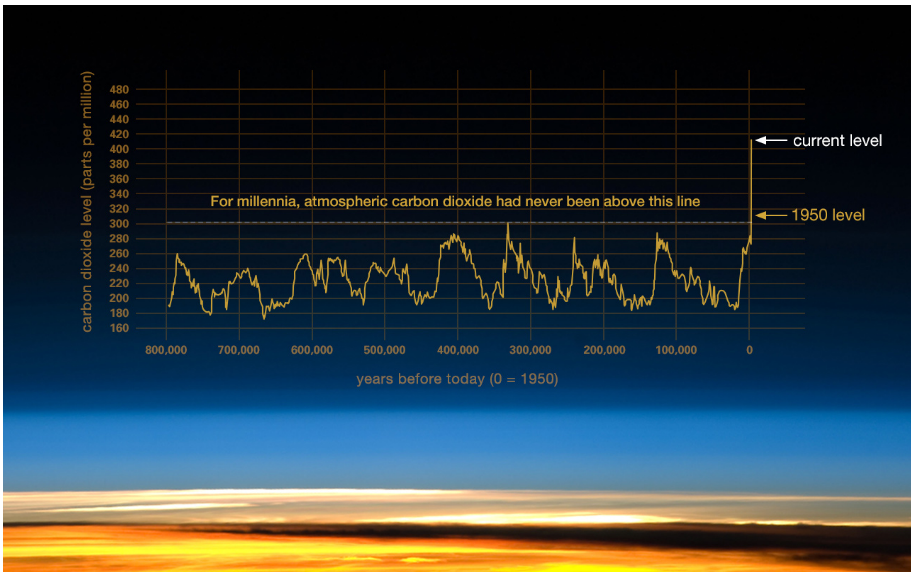
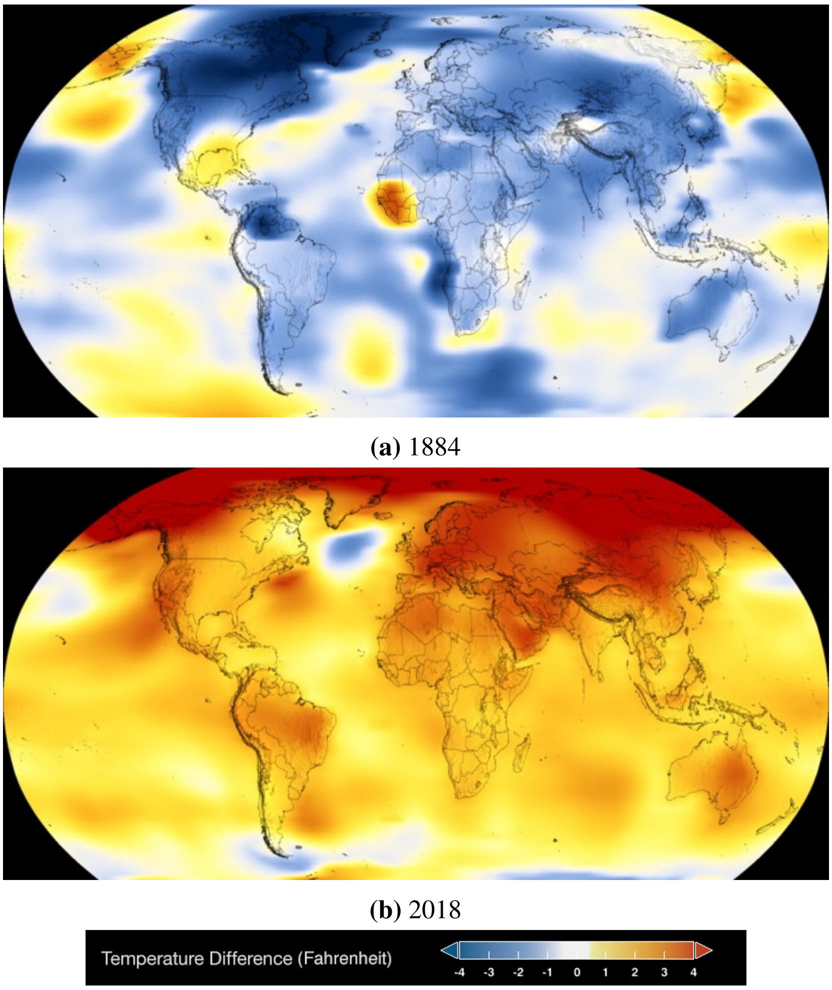


**Figure 1.1:** Carbon dioxide measurements from ice core samples (After NASA [146])

By trapping heat in the atmosphere, greenhouse gases cause the temperature of the planet to rise [146].

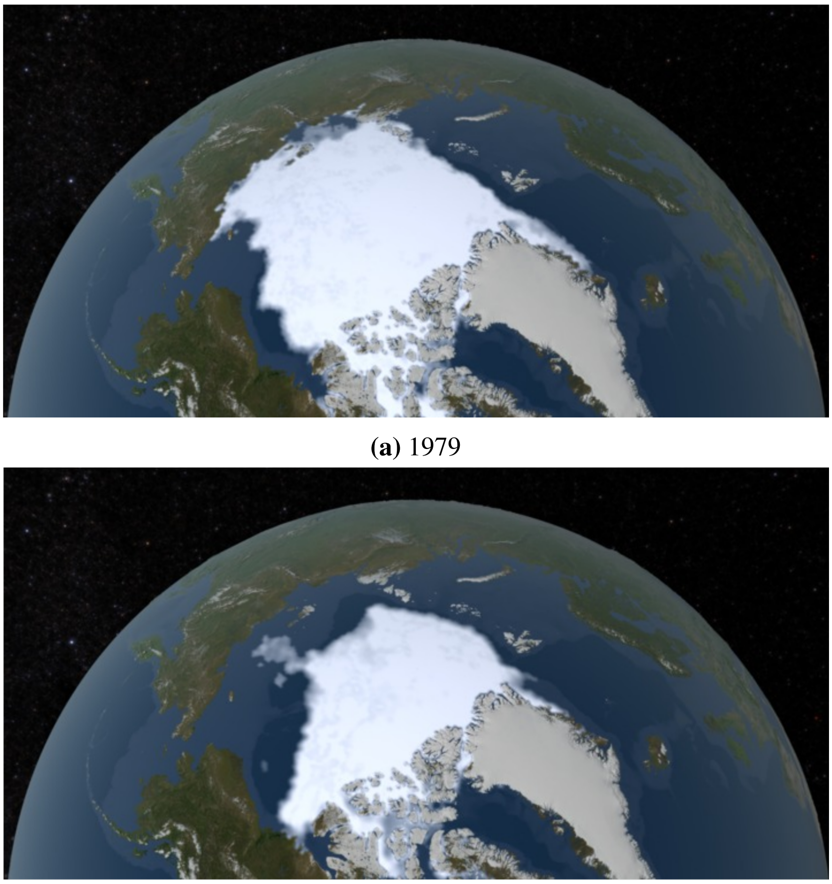


**(a)** 1884

**(b)** 2018

**Figure 1.2:** Global temperature change (After NASA [146])

This represents the largest-scale experiment ever run by humanity [120]. As the temperature of the planet rises, we can expect to see changes in global rainfall patterns, with some previously fertile regions experiencing droughts [146]. Hurricanes are expected to grow larger and more intense [146]. We can already see the effects of global climate change on the extent of Artic sea-ice levels [146].



**(a)** 1979

**(b)** 2018

**Figure 1.3:** Minimum seasonal Arctic sea ice (After NASA [146])

Electricity generation is one of the largest sources of carbon emissions, but has enormous potential for improvement [7]. It is difficult, on a global scale, to expect people to use less electricity, especially in developing countries that are making the slow climb into affluence. Increasing the price of electricity to cover the costs of renewable energy is politically unpalatable in many countries, including those that can well afford it [37]. It seems that given the current political climate, the only option is to make renewables so cheap that financial investors abandon all fossil-fuel electricity production. If we look at the current growth in renewable electricity generation, we may be encouraged, but we must bear in mind that fossil fuel generation is also on the increase.

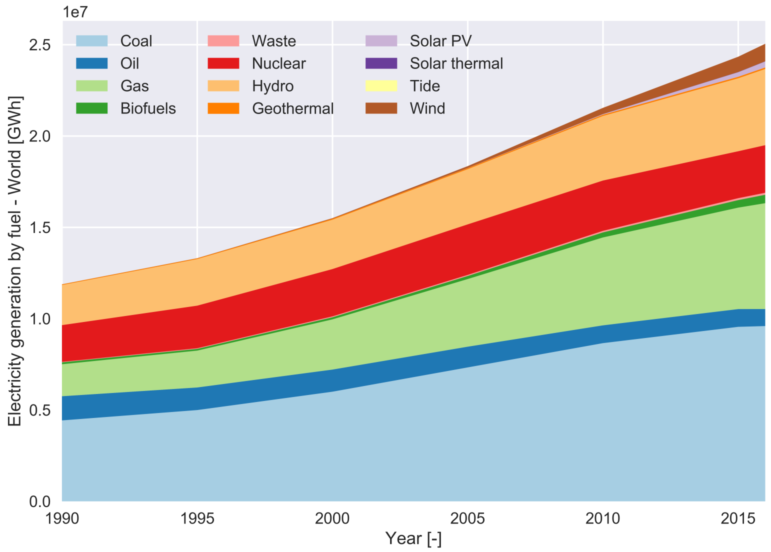


**Figure 1.4:** Electricity generation by fuel source [86]

## 1.2 Wind energy in three pictures

Although it only presents a small sliver of the total energy mix, wind energy production is rapidly increasing.

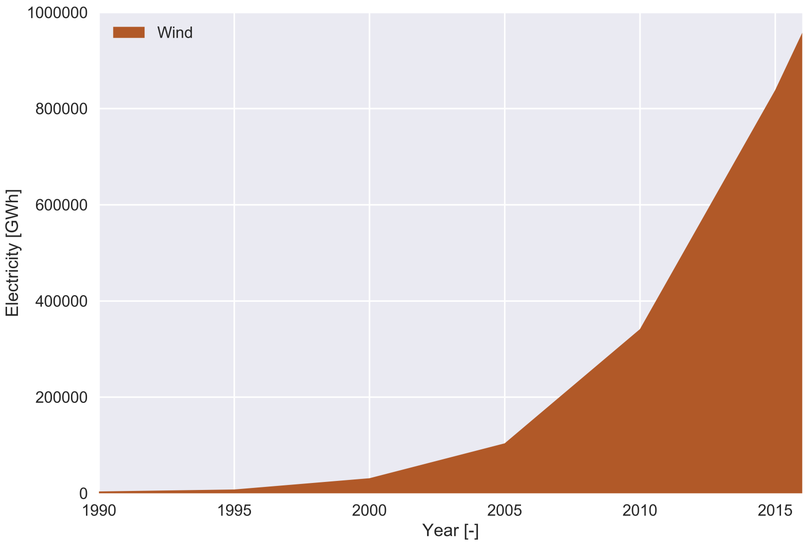


**Figure 1.5:** Electricity generation by wind [86]

If we look at the economic figures, we can understand the rapid uptake of wind energy. If we simplify our view down to the levelized cost of energy (LCOE), we can see that onshore wind is already particularly competitive, with offshore being slightly more expensive [96]. However, we must additionally account for deployability; an area in which gas or coal plants excel. In order for wind to be part of the renewable energy transformation, further cost reductions will be necessary. It seems that politicians share the view that wind energy needs to become more competitive, as they have begun to withdraw fossil fuel subsidies [169].

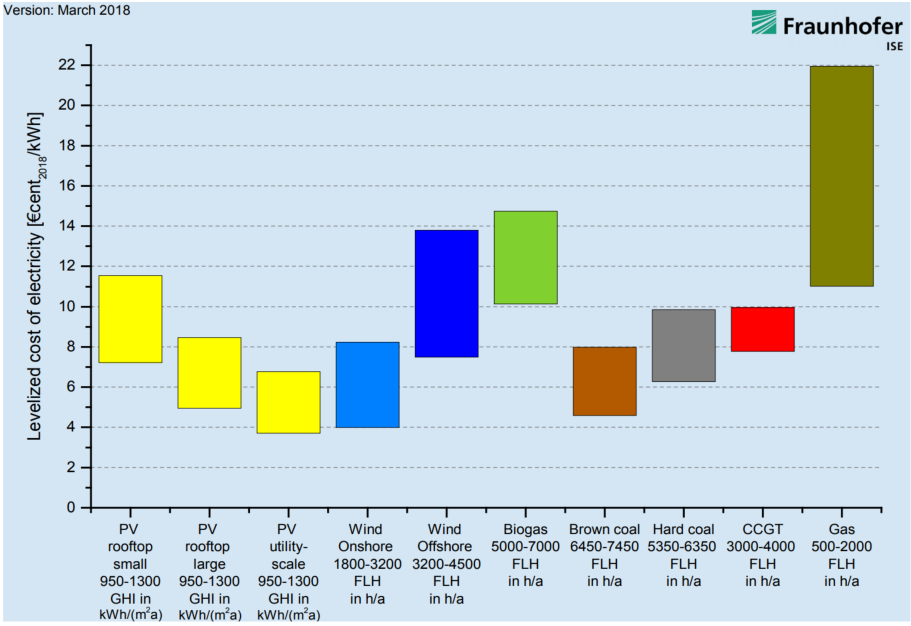


**Figure 1.6:** LCOE for electricity in Germany (after [96], reproduced with permission)

Recently, the charge for increasing rotor sizes (Figure 1.7) has come from the offshore sector. Unlike land based transport, transportation of the components by sea allows for larger rotor sizes. Offshore wind energy is more expensive and more complicated to operate. However, offshore wind is usually stronger and has lower levels of turbulence [6]. Larger rotors have better capacity factors and can amortise fixed project costs out across larger wind farm capacities for a given area [6].

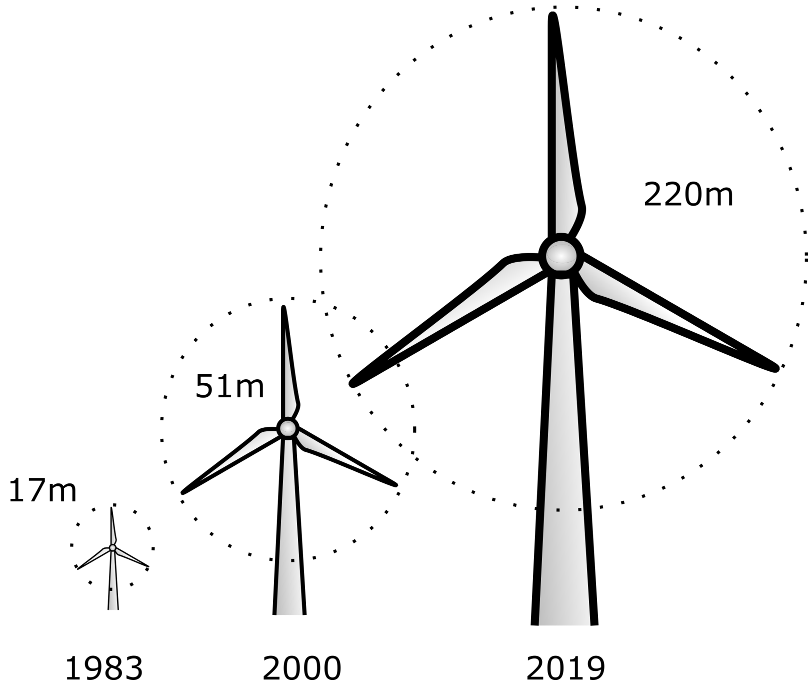


**Figure 1.7:** The dramatic growth of wind turbine rotors through time (rotor lengths after [6])

## 1.3 The current challenges of wind energy

As market pressure applies and the sizes of wind turbines grow, the challenges of wind energy shift. In 2016, the European Academy of Wind Energy (EAWE) compiled a document *"Long-term research challenges in wind energy"* [203]. The document provided a good overview of the kinds of problems that are relevant to wind energy. In the following section, I extract several examples from this document to point out another underlying pattern; namely that many of the challenges fit into one of three categories:

1. The assumptions baked into the current engineering models are too restrictive;

2. The higher physics model is too computationally expensive; and

3. Current data availability and data analysis is insufficient.

We can see that these three limitations pervade each of the technical areas shown in Figure 1.8.

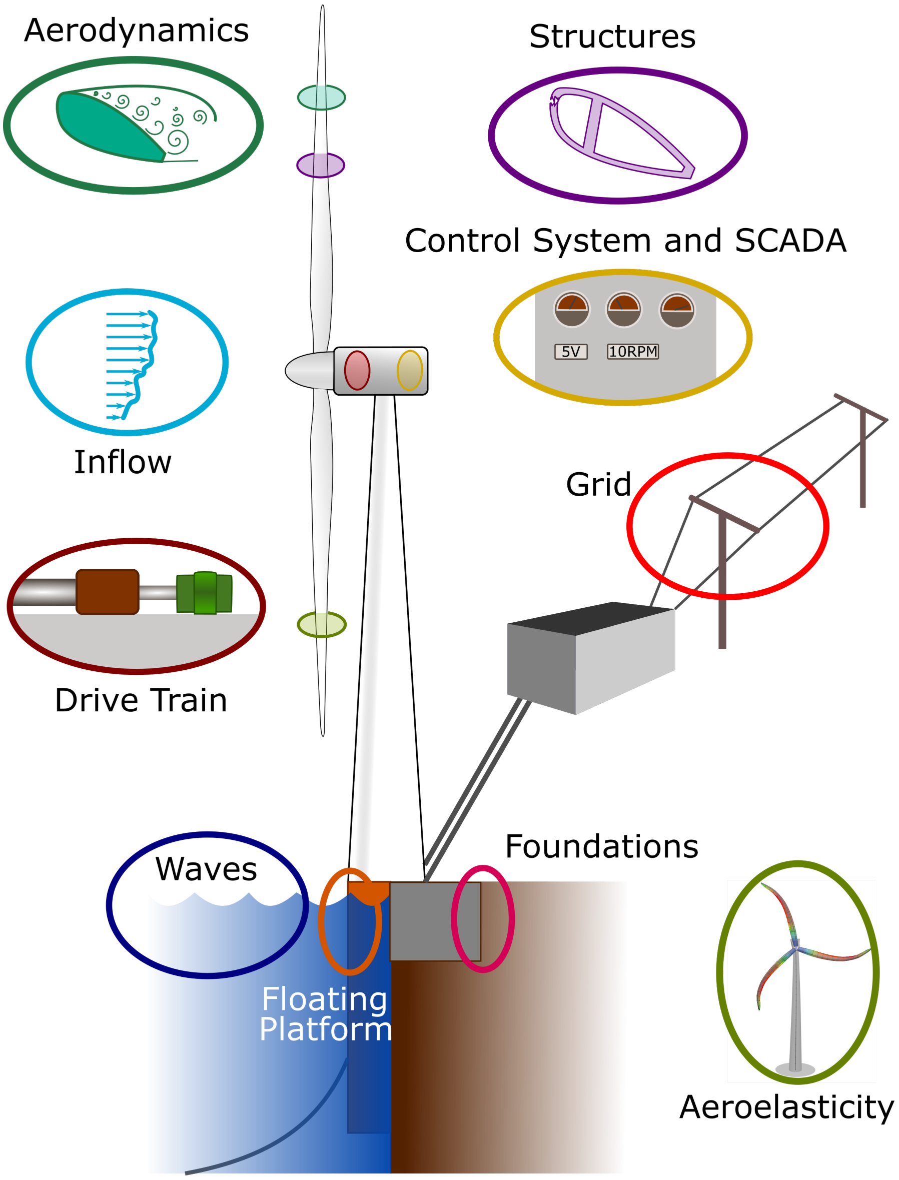


**Figure 1.8:** The different technical areas of wind energy

The following section briefly highlights some of the limitations faced in each of these areas. As aerodynamics was the most relevant focus of this study, it is featured more heavily.

### 1.3.1 Aerodynamics

#### Engineering models

The most obvious and powerful model used in wind energy aerodynamics is the blade element momentum method (BEM). In this method, a rotor blade is cut into span-wise sections and treated as two-dimensional [77]. Furthermore, the wake is treated as a simple momentum balance across the rotor plane which gives an angle of attack at each span-wise section [77]. The coefficients of lift, drag and moment are read from a table (aka polar diagrams) that derives from simulations or measurements [77]. This algorithm is implemented as an iterative solution bouncing between the wake and airfoil. It is a fast and robust algorithm and has been the workhorse simulation tool for wind turbine aeroelastic simulations.

BEM has some strong assumptions baked into its formulation. The input polar diagrams strongly influence the quality of the simulation [108][1]. The treatment of the sections as two-dimensional does not hold up to reality in areas such as the hub and tip [134][2]. Rotational flows change the stall behavior of airfoils, creating stall delay effects [55]. Wind turbines will often operate under yawed conditions [97, 89, 57, 210]. Furthermore, dynamic inflow and unsteady aerodynamics are also neglected by the model. All of these problems are treated as secondary corrections on top of the base model.

What happens when we strip away some of these assumptions? Free vortex lifting line theory (FVLLT) simulation approaches treat the wake explicitly using a Lagrangian wake formulation [134]. This means that markers that represent circulation are shed from the blade into the flow at each time step. Through the Biot-Savart equation, this vorticity translates into induced velocity, meaning that it is possible to cheaply simulate the wake of the wind turbine. This simulation approach explicitly treats tip effects (at the wake level), unsteady inflows, apparent angle of attack delay and yawed inflows. A recent work by Perez-becker et al. [159] highlighted that the BEM method has over predicted loads in quite a number of load cases. Leroy et al. [118] also showed that the vortex methods and BEM with corrections differ for floating platform cases.

The FVLLT approach still has its limitations. Although the wake elements can represent the attached flow delay effects, as you would find in Theordorsen [195] or Wagner functions [207], there is no treatment for the time delay effects of stall. This means that there is still a need for a stall model [209]. Furthermore, it is still necessary to account for the stall delay effects of the rotational flows [55].

Vorticity as a mathematical concept, has shortcomings, as vortex cores become badly conditioned at $Radius = 0$. In potential flows, the inner radius is treated with an approximation that does not diverge [134]. Furthermore, the FVLLT methods rely on relatively unsophisticated models for the viscous dissipation of the vorticity [134]. The wakes of FVLLT simulations diverge after some rotor diameters due to the Lagrangian formulation's poor ability to mimic the stretching of vortex filaments [134]. Even with the drastic improvements that of vortex methods bring to simulations compared to BEM, we are quite dependent on numerical short cuts.

BEM and the vortex models both rely on stall models to simulate the behaviours not captured by steady polars. The stall models come in a variety of styles; we will talk here about the

[1] Associated publication

[2] Associated publication

Beddoes-Leishmann type models [105, 106, 158, 78, 82, 209]. The Beddoes-Leishmann model is built in three parts - attached flow, light stall and dynamic stall. The model faces its first limitation here, as deep stall with periodic-type shedding is not included in the model - a problem that is discussed in much greater depth below. When applied to a FVLLT simulation, the attached flow part of the model is unnecessary [209]. The light stall appears in the form of a simple blending function between the attached and separated states [209]:

$$C_L^{st} = fC_L^{att} + (1 - f)C_L^{sep} \tag{1.1}$$

The blending parameter, $f$, is then determined using a simple differential equation:

$$T_f \dot{f}^{dyn} + f^{dyn} = f \tag{1.2}$$

The point at hand here is not so much the context of the derivation, but to see examples of the shortcuts used. While in some interpretations the blending parameter, $f$, may be described as the development of trailing edge separation, it is a pragmatic numerical trick; a convenient equation that seems to work.

Other numerical tricks appear in the aerodynamic models. There are two general extensions to Theodorsen's unsteady lift response functions for unsteady inflow [195] the Greenberg [88] and Isaacs [87] [3] formulations. The Greenberg derivation assumes that the unsteady component of the inflow would be small compared to the mean inflow [190]. Isaacs and Van Der Wall avoided this shortcut [190]. When tested against wind-tunnel data, it can be seen that the shortcut did eventually limit the predictive performance of the Greenberg approach [190]. In this case, a numerical short cut resulted in a lack of flexibility in properly representing the physics.

All the methods outlined above make a very similar assumption, in that we can treat the wing as a collection of two-dimensional airfoil sections. In a rotating system, with a non-infinite aspect ratio, and twisted, tapered and swept blades, such an assumption is brave. One of the key issues lies in the lack of availability of a three-dimensional integral boundary layer model in the same way as Drela's XFOIL [203].

We have seen that the engineering models can face a number of problems:

1. A mathematical simplification imposes an unrealistic boundary condition (i.e. Greenberg's assumption of low inflow variance);

2. The underlying model does not include certain physical effects or regimes (i.e. Beddoes-Leishmann's model does not account for periodic shedding);

3. The choice of formulation can be unstable (i.e. vortex core or vortex stretching); and

[3] Generalised by Van Der Wall [202]

4. There has been an over simplification of the problem, (i.e. two-dimensional airfoil sections).

These models remain very useful in spite of these shortcomings, but, in a lot of cases, the manual design of the equations has placed limitations on the predictive performance.

#### High fidelity models

Wind turbine aerodynamics simulations have a fundamental problem. The scales involved are massively different [188]. Let us consider a computational fluid dynamics simulation (CFD) that has no shortcuts - an absurd proposition, but a useful thought experiment. We would start by having a simulation of Earth's entire atmosphere. At one scale below this, we would have the terrain effects of mountain ranges and cities. One scale further down, and we would consider smaller scales of terrain effects, such as houses and forests. At this scale, we have to consider the effect of the wind turbine's wake on itself, and its effect on other wind turbines throughout the wind park. One scale further down, and we can consider the local blade inflow angles. Finally, at the smallest scale, we would consider the development of the boundary layer in millimetres. Such an approach is not possible, so at some stage during our simulations, we need to set boundary conditions to make the simulations feasible.

To investigate the effects of wind turbines on each other, it is necessary to model each wind turbine as an actuator disk [201, 63, 20, 8, 188] or actuator line [184, 214, 154, 90, 178]. An actuator disk is an idealised version of a rotor that has an infinite number of blades which induces a wake into the flow in a CFD simulation. It assumes that the forces spread evenly across the disk [185]. An actuator line model resolves the flow field using CFD but uses tabulated data to resolve the influence of the blade on the flow field [185].

It would be useful to be able to resolve the blade, nacelle and tower but this would require a grid architecture that would allow for movement, not to mention a huge number of cells to resolve the boundary layer while maintaining sensible limits on cell growth. Unsteady Reynolds averaged Navier-Stokes (URANS) simulations perform well in situations where the flow is attached [185]. However, once there is separation, URANS simulations fail to perform well [185, 173]. Large-Eddy simulations (LES) perform better for both airfoil aerodynamics and wake dynamics [173, 63]. LES results are generally higher quality but depend on the estimation of the inflow and the spatial resolution used. The National Renewable Energy Laboratory's (NREL, CO, USA) simulator for wind farm applications (SOWFA) model takes two days of supercomputer time to calculate a 10-minute simulation of two wind turbines, and that is for a model that simplifies away the blade [8]. LES simulations with the blade resolved are not currently common, and the examples that *do* exist usually cut down on computational time by using a far wake model, and only model a single blade with periodic boundary conditions [84].

Methods that mix fidelity across the scales can be quite a good way of approaching the wind energy problem. For example, resolving the full geometry of a vortex generator is a computationally expensive way of creating a simulation. Instead of fully resolving the vortex generator, the Bender-Anderson-Yagle (BAY) [15] model induces the effect of vortex generators into the boundary layer as a source term, without the mesh density required to fully resolve the geometry of the vortex generator [56]. This approach is effective, as along as the model is tuned correctly [131, 132, 13, 128, 129].

Another approach overcoming the scale problem is to use a mixed Euler-Lagrangian formulation to over come the far wake numerical stability problem of the vortex methods [11]. This model uses an Eulerian grid to resample the vortex markers in such a way that they remain numerically stable. Again, such methods rely on some sort of technique to relay the information between scales, which comes with an accuracy cost.

Within wind turbine aerodynamics, it is very difficult to create a simulation that is all-seeing, all-knowing at all of the scales we really need. For the foreseeable future, we will be setting the boundary conditions of these simulations with approximations. The high-fidelity methods tend to be relatively expensive, and are used more in the validation of lower-order models or in special cases, rather than as a work-horse tool for load case simulations.

#### Data

In wind turbine aerodynamics, accurate results are missing. While simulations struggle to set realistic boundary conditions at all scales, experiments also face significant challenges. The core of the problem lies in the impossibility of simultaneously performing tests under realistic boundary conditions while accurately measuring those boundary conditions. [203]

In the laboratory, it is difficult to create realistic levels of turbulence and a Reynold's number for airfoil testing. While improvements are being made in this direction, using active-grid wind tunnels [100], it is still challenging. Furthermore, once in stall, even a two-dimensional airfoil section has complicated three-dimensional stall patterns across the span [130]. Testing a three-dimensional blade in a rotating system in a wind tunnel is a step in the right direction, but the increase in complexity makes basic inflow conditions, or local blade inflow conditions, difficult to identify [156, 177].

The realistic operating environment of a wind turbine blade is complicated. Leading edge erosion and fouling are different in each case, and have a strong effect on the boundary layer [161, 162, 199, 168]. The blades will also have large manufacturing tolerances which can result in dents that cause small bubbles [199] not normally associated with wind turbine airfoils. Furthermore, it is hard to measure, to a fine scale, what inflow each section of the blade is experiencing at each time step. Normally, even instrumented wind turbines have a relatively sparse array of sensors [121], and commonly we only have measurements that are integral values such as the rotor power and root bending moments. Such integral values are dangerous error metrics, as errors can cancel out. Experimental campaigns, such as the DANAERO Project, represent a move in the right direction [121], but it is difficult to say that we have a solid ground truth from which we can design our models.

### 1.3.2 Aeroelasticity

#### Engineering models

Aeroelasticity presents an interesting array of engineering models. In static aeroelasticity problems, we need to incorporate predictions of the aerodynamics and structural stiffness [50]. In dynamic aeroelasticity problems, we need additional inertial terms in order to complete the collar triangle [50]. In a wind turbine context, then also have to include the effects of a pitch, torque and yaw controller.

We can test aeroelastic stability in the time domain, but often it is desirable to have a direct formulation [172]. In a standard derivation of coalescence flutter we would assume:

1. Harmonic motion;
2. A linear lift slope or, in more advanced versions, a principle component analysis reduced order model [194];
3. Linear structural properties;
4. Attached flow; and
5. A two-dimensional section.

If we extended this out to a wing shape, we would rely on the additional linear superposition of modes in formulations [50].

In time-marching aeroelastic simulations, we are also forced to make simplifications. It is standard practice in many codes to sparsify the description of the structural dynamics to the first two flap-wise modes - first edge-wise and first torsional models [91]. BEM methods are the most commonly-used representations of the aerodynamics, with extra models to account for the unsteady aerodynamic effects [76]. It is an issue that the unsteady aerodynamics models vary significantly, and fail to accurately capture the physics of dynamic stall or periodic shedding; these affect the aeroelastic stability [9, 81, 112][4]. Even the quality of the input polar data would have a strong effect on our load estimations [111, 2][5].

While these methods have generally performed well, it should be noted that we are relying on the manual tuning of semi-empirical models. The unsteady aerodynamics models are quite

[4] Associated publication

[5] Associated publication

commonly tuned against the NREL dataset measured at Ohio State University [27, 197, 74, 160, 209, 73, 196, 142, 82, 51, 168][6]. As airfoils geometries have changed over time, we can justifiably question whether our models are still correctly tuned. The structure of such models can also be a design choice; for example, in the Rayleigh-Ritz approach, we have to manually select the trial functions used to fit the mode shapes [50]. In the end, the choices need to be justified against data.

#### High fidelity models

The highest-fidelity approaches available for time marching aeroelastic involve a finite element representation of the turbine with CFD for the aerodynamics [49, 176]. This is not a trivial task, though, as the blades are then moving through the domain, either requiring computationally expensive remeshing or another interpolation scheme [49]. FVLLT with a finite element structural model moves in the direction of higher fidelity, while maintaining acceptable computational costs for engineering design [113, 175, 134, 135]. The downside of these methods is that they are time marching. This means that, to test that a wind turbine will not flutter, we need to run relatively exhaustive tests across the operational domain.

#### Data

Aeroelastic data is difficult to obtain for wind turbines. While some wind tunnel tests do use aeroelastic similarity laws to scale down wind turbines [21, 32], this is still represents an incomplete comparison to real machines. In the field, however, it is unlikely that we have a full enough understanding of the boundary conditions to enable us to make fine adjustments to the models. This has been a challenge in addressing vortex-induced vibrations [183, 69] However, some examples of field-testing *do* exist that have helped in blind-testing aeroelastic models [29, 12, 162, 121, 122, 119, 16, 74].

### 1.3.3 Inflow, wakes and turbulence

#### Engineering models

At its core, turbulence is an abstraction used to understand flow perturbations. If we look at the Van der Hoven spectrum of wind speeds, we can see that there is a spectral gap between long-term patterns, such as seasons, synoptic changes and the day/night cycle, and the short-term fluctuations [19]. In wind energy, we consider everything shorter than the spectral gap as turbulence [19]. Our description of the turbulence in terms of the zeroth- and second-order spectral moments, relies on a strict assumption that the unsteady wind component is normally

[6] This may change in the future with recent efforts to make data easily accessible [67]

distributed. Storm conditions, in particular, can be instationary and break these assumptions, including the assumption that we can split the steady and unsteady components into independent parts [19]. Furthermore, offshore conditions tend to create abnormal wind shear profiles which are not described by the typical low-order models, the simplest of which, rely on assumptions of constant vertical fluctuations of temperature, velocity and shear stress, sufficient time averaging and a uniform surface roughness [141].

Even under less extreme conditions, we tend to make strong assumptions. The IEC61400 standard calls for load cases, including the Mexican hat gust [123]. This gust arrangement is considered to be an extreme alignment of various scales of turbulence. Bos [19] makes the point that, just because an event is rare, it does not necessary follow that it produces the greatest amount of damage. There are more common events, that tend to cause higher damages even though the Mexican hat is a severe test case for the controller [19]. With modern rotor sizes, it is also open to question whether engulfing the entire rotor in a gust is a realistic test case [19].

Unfortunately due to the scale problems, it is not possible to directly imitate turbulence in wind-turbine design simulations. Instead, we rely on a statistical representation of the flow perturbations. One strong assumption is that inflow turbulence can be superimposed onto the mean wind speed. It is unclear whether this assumption holds in cases with complicated terrain or wake interactions [188]. It is also common practice to use Taylor's frozen wake hypothesis to simplify the calculation of turbulent wind fields in wind field simulations [188]. Turbulent wind fields are often generated using pragmatically designed models [125, 205].

### High fidelity models

When we increase the fidelity of the representation of turbulence, we can see that results diverge. Branlard [24] created a vortex-particle-resolved LES simulation. Strong differences arise in the wake development, particularly when the wind shear is resolved rather than frozen. Ramos-García et al. [165] also showed such the same effects for shear, but went further to demonstrate the dramatic effects of inflow turbulence on wake mixing. To be clear, this method simply allowed for turbulence to be freely transported through the model rather than be frozen. The inflow turbulence was still defined by the Mann model [125], although it is also possible to use precursor simulations to achieve the same result [192]. These models are too computationally intensive to include in day-to-day wind turbine analysis.

### Data

Atmospheric turbulence exerts a strong influence with on wind farms. Measurements show that different atmospheric stability conditions yield different performance outputs from wind farms [75]. More stable conditions cause a greater drop in performance due to decreased turbulent mixing. The most common measurement devices - anemometers - only sample the wind speed in exactly the stream-tube where they are located. In reality, gusts do not necessarily engulf the entire wind turbine, especially given the size of modern rotors. Fortunately, recent developments in LIDAR [14] technology mean that atmospheric data are becoming a normal part of the wind energy dataset.

### 1.3.4 Floating platforms, foundations, waves and offshore

#### Engineering models

As wind turbines move farther offshore, floating platform wind turbines are transitioning from being a research topic to commercial reality. The HyWind project from Equinor consists of five floating platform wind turbines, each with a 6-MW capacity [1]. Further project plans have been announced for a 200-MW floating platform wind farm in South Korea [62].

The simplest models rely on potential flow assumptions, augmented with drag and damping coefficients obtained from experiments or high-fidelity simulations [203]. The real difficulty arises because of the highly non-linear nature of waves. Consider that a model may have to accomodate; breaking waves, free surface stretching and, interaction with the structure, not to mention the effects of changing water depths [139]. It is difficult to compress these effects into a simple numerical formulation. Approaches like the Morison equation [143] discard effects such as wave diffraction based on Taylor's long-wavelength approximation [139]. The Morison [143] equation also assumes that viscous drag dominates, and ignores wave radiation damping [139], the implication being that the Morison equation assumes that the structure is invisible to the waves. All of these simplifications are required to reach a workable set of equations.

Even onshore foundations are not straightforward to model. The currently used p-y curves represent a simple model that relates the deflection of the pylon to a non-linear spring constant [203]. The problem is that, over time, soil properties change due to cyclic stresses. For both onshore and offshore the models seem to be over simplified representations of the system.

#### High fidelity models

For floating platforms, once the Morrison equations are no longer valid because of the structure being visible to the waves, we need to move to higher-fidelity options. A common approach is to treat the problem of incoming wave-boundary interactions using panel methods. The cheapest version of the simulation uses quadrilateral panels, whereas the expensive version uses B-splines to represent the potential [151, 92, 103]. Another approach is to use particle methods to represent the waves [102].

#### Data

Floating platform validations are difficult to make at the moment due to a shortage of available data at the appropriate scale [189]. For full-size testing, the only data available is from Equinor (previously Statoil) [54], this data set is not publicly available but does appear in some academic papers. In this case, a lack of data will be a barrier to improving the models used in floating platform research.

### 1.3.5 Structures

#### Engineering Models

Fatigue in composite wind turbine blades has always been a challenging topic. In available guidelines, such as the IEC61400 [123] or the DNV GL [47], the full stochastic loading scenario compresses down to a single set of damage-equivalent loads. This family of methods is called fatigue-life models. The biggest limitation is probably the use of Miner's rule, which sums up the damage from each cycle [124, 203]. Some test data seem to agree with the Miner's rule assumption, but researchers still openly question how well this formula generalises to all types of composites and loading states [35]. Constant life diagrams also can incorporate very strong assumptions such as a linear relationship between strain amplitude and strain mean for a given constant life [171]. Both Miner's rule and the constant life diagrams are good examples of engineering type models that do not actually consider the actual physical damage mechanisms [33, 34][7].

In reality, the geometry and material properties of a wind turbine can vary strongly depending on the manufacturing process. Sources of variations include cycle curing, bonding defects, the fibre volume fraction, waviness and voids [36, 171, 31, 147, 148, 150, 149]. Furthermore, there is evidence to suggest that the order of loading is actually important. Castro-Ardila [36] attempted to extend the fatigue life models by creating a statistical representation of the constant life diagram.

Phenomenological fatigue models predict material property degradation based on empirical relations something in the direction of a data-driven approach [36]. Unfortunately, such models need test data in order to tune them to each case analysed, making them difficult to generalise [36]. The topic of model generalisation is discussed in depth in Chapter 2.

#### High fidelity models

The third composite fatigue category includes damage mechanics models [36]. These models incorporate the most detailed physics of the damage mechanisms, such as off axis matrix cracks,

[7] Associated publication

delaminations and fibre breakages. Essentially, damage initiation and propagation is physically modelled, rather than implied. While two-dimensional finite-element methods do provide useful information, they struggle to analyse certain sections, such as the transition between the circular root and the airfoil. Furthermore, buckling (which can be brought on by fatigue damage) and delaminations lie outside the predictive ability of two-dimensional finite-element models [36]. For these reasons, a three-dimensional damage-mechanics based finite-element model would be necessary to entirely analyze a wind turbine blade in high fidelity.

#### Data

As of writing, the largest blade in development is 107 m for a 12-MW wind turbine [95]. Fatigue testing is usually conducted at the eigenfrequency of the lowest blade modes to make the test energy efficient [153, 104]. As the structures get larger, the eigenfrequencies drop, making each cycle slow, nonetheless the blades have to undergo enough cycles to represent a 20+ year service life. This means the tests are becoming very long.

Composite material data is usually collected at three different levels small coupons, subcomponents and full-scale tests [140]. Coupon tests are cheap, and so a large number of tests can be performed, ensuring statistical convergence. However, a coupon test does not necessarily represent real test conditions. As we move into subcomponent and full scale testing, the cost of testing increases dramatically. For full scale tests, usually only one or two blades can be subjected to a full test. This means that, even at the full scale, where we can test something close to real conditions, we will still never have enough data to provide statistical significance [140].

There is a significant motivation for assessing wind turbine composites from a statistical standpoint. As already highlighted, manufacturing problems cause a wide variation in properties. Defects such as waviness in the matte material will result in drastically reduced strength. Nelson et al. [148] suggested applying adjustments for such defects, but high degrees of uncertainty remain, which propagate into all fatigue-life estimates [171, 35]. Research on these problems is on-going in projects such as "BLATIGUE" at Denmark Technical University [46].

### 1.3.6 Control system, SCADA, grid

#### Engineering models

The wind turbine controllers implemented in the industry are based on classical design techniques. They often use simple PI or PID algorithms coupled with various filters, such as notch or band-pass filters. In the wind energy context, the control system can usually be reduced to a set of (almost) uncoupled, single-input/single-output controllers [28]. The downside of these classical-type models is that the tuning requires a high level of skill and experience [28]. Floating platform wind turbines [65, 18], wind farm wake effects and degrading aerodynamic performance are challenging for simple controllers [203]. For example, floating platforms require an extra control loop in order to use the pitching of the wind turbine blades in order to stabilize the platform motion [54].

As with aeroelasticity, testing in the time domain is a very expensive undertaking due to the broad operating envelope. Usually, the controllers are first tuned in the frequency domain, on linear controllers, and then fine-tuned in the time domain. While the linearisation is conducted locally [91], there is still an associated accuracy cost.

#### High fidelity models

Many more advanced controller concepts exist that could be used for wind energy. Burton et al. [28] argued that, while such methods have the advantage of being mathematically rigorous, tuning the cost function for these models ends up just as cumbersome as manually tuning a classical controller.

#### Data

Data is a real problem for wind turbine controllers. For many controllers[8], the only availiable inputs will be the rotor/generator speed, pitch angles and wind speed from the anemometer. The one environmental signal available, the anemometer will be offset and quite noisy due to it being mounting on the nacelle. Of course, these readings are often corrected through CFD simulation, but such a correction can not possibly recover a good-quality signal. Any improvement in wind turbine controllers will probably start with the inclusion of additional sensors, such as LIDAR or optic-fibre strain gauges.

## 1.4 The common themes

Throughout this brief, and by no means exhaustive sampling of wind-energy problems, the reader will likely have noticed the following three recurring themes:

1. The assumptions baked into the current engineering model are too strong;
2. The higher physics models are too computationally expensive; and
3. Current data availability and data analysis methods are insufficient.

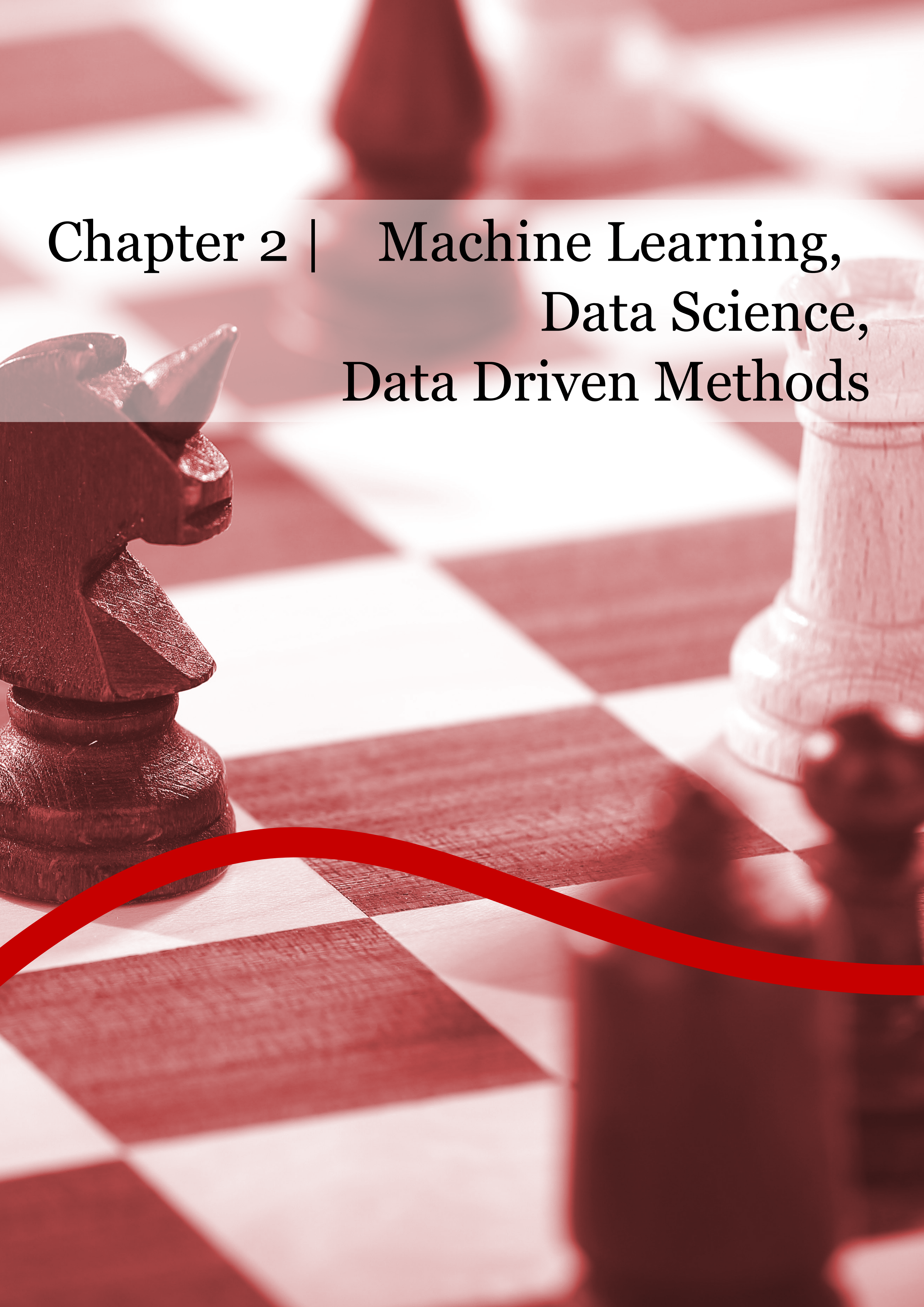
These three themes are distributed differently across the various disciplines, but roughly the same issues are there in each.

[8] To be clear, the supervisory controller will have more monitoring inputs

These themes are important because, in all three areas, there is work to be done improving the availability of data, improvement of engineering models, and the improvement of higher fidelity tools, but, data science and machine learning may also help make improvements. For instance, part of an expensive numerical process may be replaceable with a cheaper, machine-learned equivalent. An engineering model may be extended by replacing an assumption with a machine-learned model. The manual choice of empirical equations could also be replaced with machine-learned equations [67, 94, 25]. Data may be reinterpreted in a way that makes it useful for model building. While this study focused on aerodynamic and aeroelastic problems, the principles should transport into other fields as well.

Here I would like to make a philosophical claim about black-box models. Many of the models I discussed above are incredibly complex and their derivations even more so. Here is the core of my claim: once models reach a certain complexity, it does not matter whether a scientist nearly 100 years ago created the formulation or whether a computer decides - we probably do not fully understand the implications of the mathematical formulas used in the model. For most of us, these models are black-box, or at least very murky. Why do we trust them? Because they work, and because we understand the boundary conditions under which they work. Machine-learned models are no different, we just need to think differently about how we test the models. In the end, all models are wrong, but some are useful [23]. In the next section, we will explore this concept further by discussing machine learning approaches in greater detail.

# Chapter 2 | Machine Learning, Data Science, Data Driven Methods



# Chapter 2 Machine Learning, Data Science, Data Driven Methods

On the 3rd of May 1997, the world gasped as Gary Kasparov resigned during the second game of the chess tournament against IBM's Deep Blue. It did not matter that a bug in the programming was responsible for the move that had stupefied the Grandmaster [182]. It did not matter that Kasparov could have held out for a draw. What mattered is that, for the first time, there were enough transistors available in Deep Blue's architecture to encompass enough of the informational entropy of chess to enable play at the highest level.

Fast forward through 21 years of computer hardware and artificial intelligence (AI) innovation and we arrive at DeepMinds AlphaStar [206][9]. AlphaStar has been trained to play the computer game StarCraft. The increase in complexity since Deep Blue is startling. The game plays out in real time, and the computer has to make decisions based on imperfect information; that is, the computer does not know what the opponent is doing. The game involves incredibly long chains of cause and effect throughout each game which run for up to 1 hour, resulting in tens of thousands of time steps. Each time step can involve, between 10 and 26 actions, meaning that the overall entropy of the game is orders of magnitude greater than chess. Against one of the worlds best human players, AlphaStar won 5-0. It won by creating strategies never previously seen by humans, the training data did not lie and the computer found a way to beat human player.

The computing power behind this strong agent comes from an architecture called a tensor processing unit [206]. TPUs and their cousins the graphics processing unit (GPU), are well suited for the tensor operations that form the backbone of neural networks [93]. The cost of using such a computational resource has been dropping at a rate of roughly one order of magnitude per 12 years [3]. This has resulted in vast amount of computational power becoming more readily available. Ultimately, this gives developers the chance to use highly complicated and flexible AI models. The result is that incredible achievements like the AlphaStar AI, are possible.

Computational power is only one part of the AI equation. The second part is the vast amounts of research resources being invested by private sector companies like Amazon, Baidu, Netflix, Uber, Tesla, Facebook, Alphabet/Google, Microsoft and IBM. The consulting firm McKinsey estimated a research and development investment from the top-tier tech companies of

[9] It is worth reading the original article to obtain an understanding of the training process [206]

$20-$30billion in 2016 alone [26]. This investment and other, publicly-funded, research efforts have yielded well-developed open-source machine-learning libraries, such as sci-kit learn[157], TensorFlow (Google) [133] and PyTorch (Facebook) [155]. These abstract away the daunting task of programming on a GPU or TPU from the user, and provide front ends in high-level languages, such as Python. Furthermore, projects such as Dask [43] and Numba [101] make it possible to compile Python and parallelise code across large-scale computational infrastructures. Projects like fast.ai [83] and Keras [39] have provided even higher-level interfaces, exposing some of the most powerful machine-learning techniques to anyone with a modest amount of programming skill. In short, the most advanced machine-learning techniques are within the reach of non-specialists.

The third part of the machine-learning recipe is data - huge volumes of data [191]. Each of the technology companies mentioned above has billions of users and thus billions of users' data, transactions, driving behaviours and images. By the way of example, the ImageNet database contains over 14 million labeled images [45], the AlphaStar AI was trained on 200 years' worth of game-play data [206] and Tesla has been tracking driver behavior since 2016 [42]. Such huge datasets contain a wide diversity of data meaning that the AI is unlikely to be 'surprised' by new examples. The more flexible a model is, the more complex a task it can achieve; however, highly-flexible models need to be constrained with large datasets or they will tend to explode in regions not populated with training data.

The modern environment of huge investment, large-scale computationale power and big data has resulted in some other amazing examples of machine learning. The WaveNet neural network architecture can generate convincing new music samples [61]. Style-transfer neural networks can transform a picture into any given art style, such as a Picasso or Monet [68]. 'Deep fake' images and videos of famous actors and politicians have been convincing enough provoke controversy. Such advancements show that the deep learning side of machine learning has a tremendous future.

WaveNet, style transfer and deep fakes represent the most exotic examples of machine learning. Underpinning these, there is a vast range of algorithms and methods that are more useful in the day-to-day context of wind energy. Machine-learning models can undertake a range of tasks, such as classification, regression, clustering, creating association rules, forecasting, dimensionality reduction and density estimation [157]. Each of these tasks can be achieved by a large numbers of algorithms, ranging from simple linear regression through deep learning. The important constant is that the techniques behind machine learning can turn data into models. Models that even chess Grandmasters cannot out think.

## 2.1 The Language of Machine Learning

Machine learning has it's own language and definitions. Let us rewind and first establish what is meant by machine learning.

**Definition 1.** *"Machine learning is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as 'training data', in order to make predictions or decisions without being explicitly programmed to perform the task" [40]*

The most basic example of machine learning is the humble polynomial regression. Let us generate some random data points, and then make a fit with different polynomial orders.

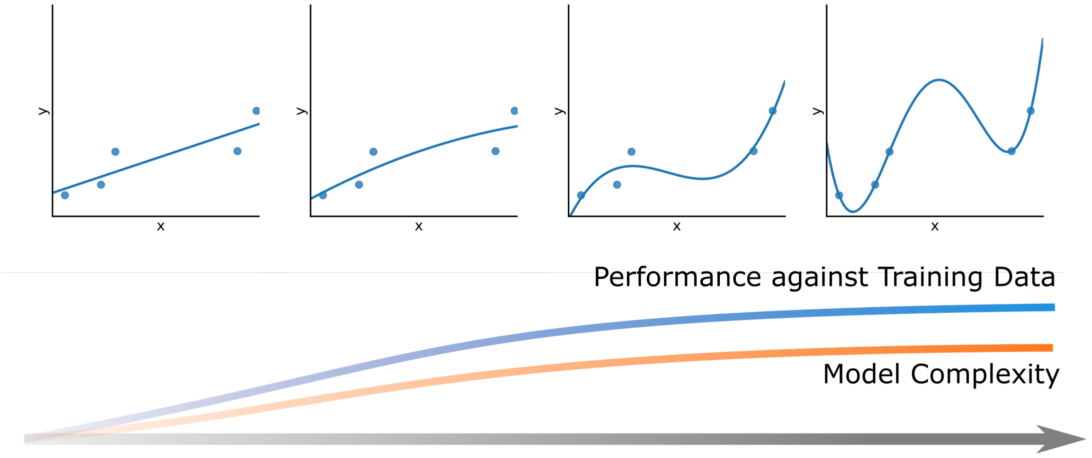


**Figure 2.1:** The development of training error with model complexity (in this case, polynomial order)

We see that, for the linear fit, the line roughly captures the trend, although the data doesn't sit on the line perfectly. This introduces the second definition:

**Definition 2.** *Bias occurs when an algorithm has limited flexibility to learn the true signal from a dataset, also known as underfitting [58].*

As the polynomial order increases, the fit becomes perfect. We may be tempted to take the highest-order polynomial and consider the job done; however, there is a danger sign. While the highest-order polynomial passes through all the data points perfectly, the curve is 'wiggly', more formally said, the curve contains some large gradients [17]. Let us now take some more data from the data set and add it to the plot (in red).

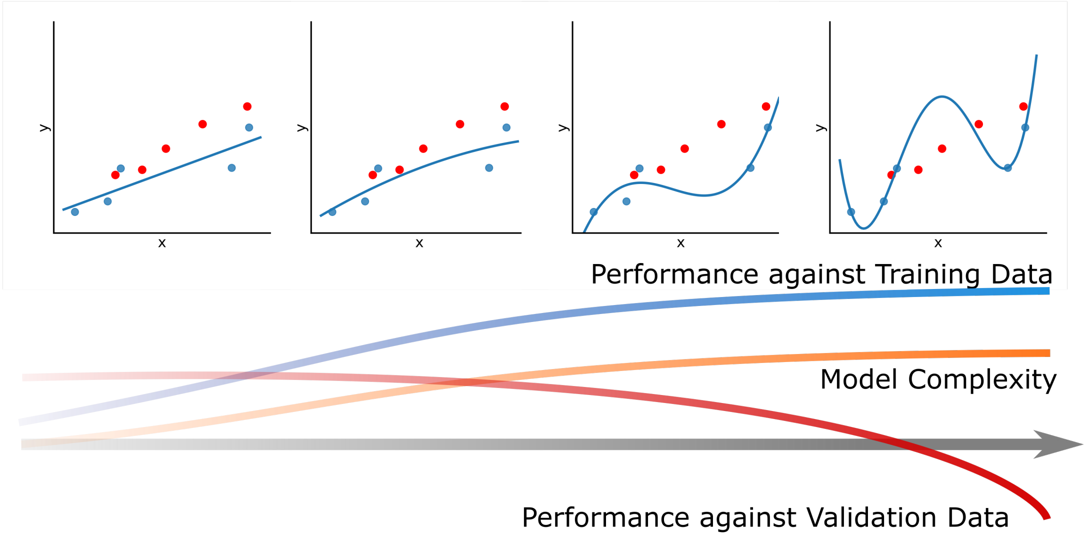

y
x
y
x
y
x
y
x
Performance against Training Data
Model Complexity
Performance against Validation Data

**Figure 2.2:** The development of validation error with model complexity (in this case, polynomial order)

While the high-order polynomial could fit the data it trained with, when we apply it to new data, the model is almost useless; the model does not generalize well. This introduces our third definition.

**Definition 3.** *Variance refers to an algorithm's sensitivity to specific sets of training data, also known as over-fitting [17].*

When designing a model, we have to consider both the bias and variance. If the model is too simple, it will never represent the underlying data [17]; if the model is too complicated, it will do everything possible to match the data it trained on [17]. However, it will not represent new data well. This is called the bias - variance trade-off and the aim is to find the sweet spot that minimises the total error that the model produces [17]. The bias/variance trade-off is one of the ground truths of machine learning.

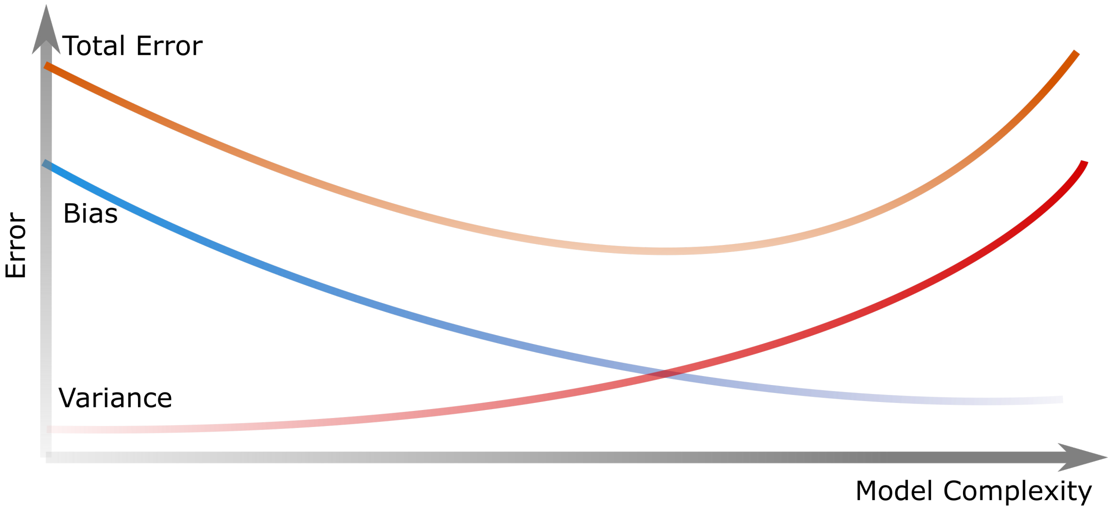


**Figure 2.3:** Bias/variance trade off

Now we know that we have to account for both bias and variance, how do we go about selecting the best polynomial fit from the example given above? We split the data into three parts. The first part is called the training data, and this is usually the largest share. The training data will be used to regress the selected model. The error from the training process indicates the model's bias - commonly referred to as 'training error'. The second part of the data - the validation data - is used to check how well the model generalises, so that a model selection can be made. The validation error is the indication of the model's variance[10]. We can use the model error from the training and validation data together to find which of the polynomials fits the data best. The third part of the data - the test set - gives a final indication of the model's performance.

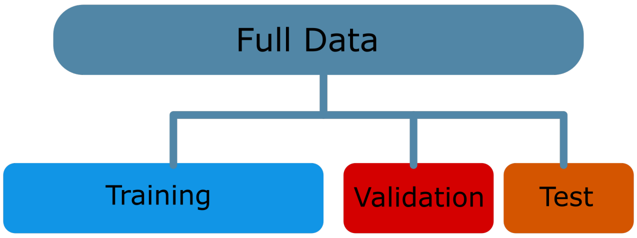


**Figure 2.4:** Splitting data into - training, validation and test datasets

In the polynomial regression, we used the validation data to test which order of the polynomial suits the data. The polynomial order is an example of a hyper-parameter:

**Definition 4.** *A hyper-parameter is a parameter that is set before the training process commences [204]*

[10] Here, I avoid diving into cross-validation techniques for the sake of simplicity. For a more detailed treatment see Bishop [17]

In contrast, the coefficients of the polynomial regression are parameters.

**Definition 5.** *A parameter is a variable that is set during training [204]*

We can interpret hyper-parameters as determining the architecture of the model, whilst the parameters are the contents of the model itself.

## 2.2 The tasks of machine learning

Machine learning encompasses a wide range of tasks, each of which has its own challenges and algorithms. In the following section, the main tasks of machine learning are discussed in depth, and either a known example of an application or a plausible application in wind energy is provided. I avoid discussing specific algorithms wherever possible, as this would take an entire textbook to cover, even for the most common algorithms (e.g. Bishop [17] or Raschka [166]).

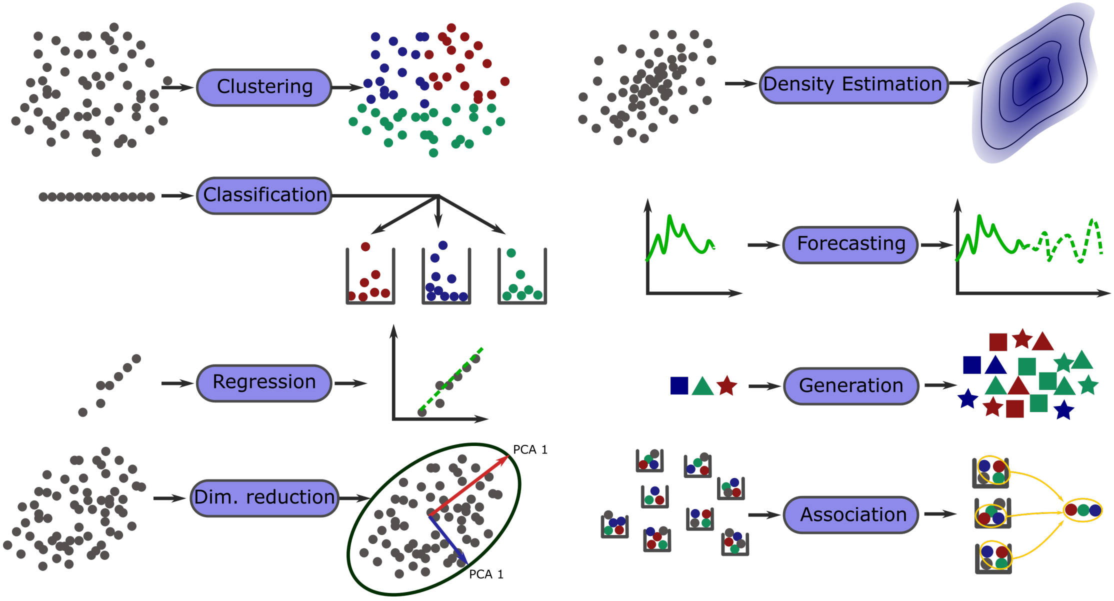


**Figure 2.5:** The tasks of machine learning

### 2.2.1 Regression

Regression is the most obvious form of machine learning, where input is transformed into a continuous output [166]. Indeed, the polynomial example given above is an example of regression. In wind energy, there are many examples:

1. Fitting lift and drag polars from simulation data or experiments;

2. Estimating the power curve of a wind turbine from operational data;

3. Determining the required blade pitch setting for each wind speed; and

4. Finding the convection speed of a dynamic stall vortex traversing an airfoil from pressure information

The input data may vary, but regression always results in the output of one or several, continuous variable(s).

### 2.2.2 Classification

Classification takes input data and puts it into a category or categories [166]. An example of a single-label classification is determining whether a picture is of a dog or a cat. A multi-label classification would list all the features in a satellite image (i.e. land, river, trees). Training a classifier requires a set of input/output pairs, making it a supervised method [166].

In the context of wind energy, there are a number of uses for clustering. In the computer-vision-type models, a good example of classification would be using images taken from a drone making blade inspections [181]; each image is processed, and labels are added for common elements, such as vortex generators, or defects like visible cracks or lightning-strike burns. Classification can also determine the operating state of the wind turbine from the Supervisory Control and Data Acquisition (SCADA) data. This means that we can identify unusual operating states, which can arise from faults; this is called anomaly detection [213]. Classifiers can also be built for time-series classification, an example of this is determining the boundary layer state on an airfoil using microphone data [109].

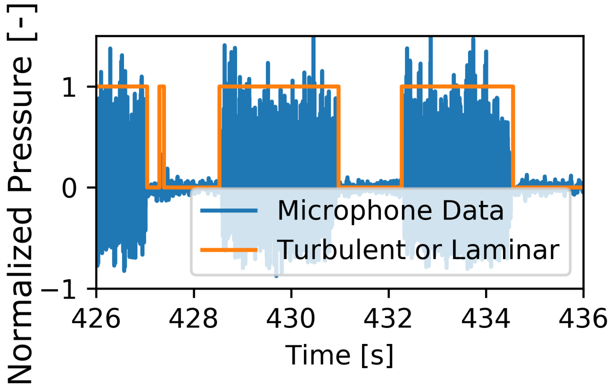


**Figure 2.6:** Labelling parts of the pressure signal as laminar or turbulent (after Lennie et al. [109])

### 2.2.3 Clustering

Clustering takes a set of data and splits it into groups of local neighborhoods [166]. A well-functioning clustering algorithm will have groups where each of the items in the group are very similar to each other and very different from the items in other groups. Clustering is an unsupervised method, meaning that the algorithm is not based on pairs of input and output data [166]. Rather, the methods have to have their own success metrics. The beauty of neighborhood-type methods is that you can break down the dataset before trying to make inferences. This is particularly useful in datasets where there are multiple modes or highly skewed data [166].

In the context of wind energy, there are two notable examples. The first is the use of clustering to split data into groups to help in understanding the variations within the dataset. This is covered at length in this thesis. The second use is breaking up the operating regime of a wind turbine into several clusters.

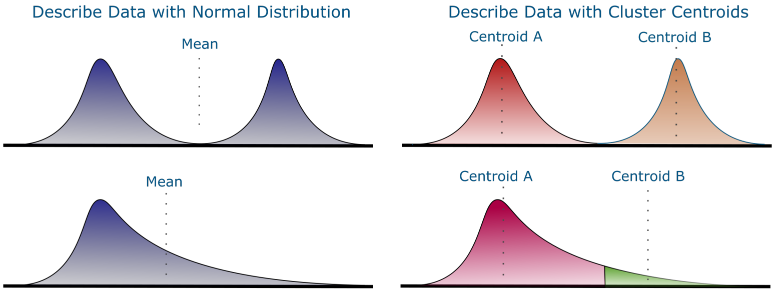

Describe Data with Normal Distribution
Describe Data with Cluster Centroids
Mean
Centroid A
Centroid B
Mean
Centroid A
Centroid B

**Figure 2.7:** Breaking up a dataset into clusters

Each cluster can then have a customised controller. The advantage of this is that each controller can be perfectly suited to its operating region. This is preferable to a case where a single controller has to be highly flexible in order to fit significantly different regimes. Examples of this can be found in the work of Nair et al. [145].

### 2.2.4 Dimensionality Reduction

Consider a relatively simple geometry - a spiral. The spiral can be plotted in terms of its x and y coordinates. In this case, Euclidean geometry is an inefficient way of describing the spiral. Euclidean geometry has to be flexible enough to describe all possible shapes. In this case, this spiral represents the location of our data, meaning that we can discard the geometry that does not describe anything. In other words, the geometry of the spiral is sparse in the two dimensions used to display it, and our data is confined to a small volume fraction [17]. We could, instead, define the spiral as a unidimensional manifold, simply mark the center as zero, the outermost point as one. By doing this, we are unrolling the spiral into a flat line and reducing the two-dimension system to one dimension [180].

Why is it useful to do this? Let us consider a standard-sized photograph from a modern digital camera. Each picture will have 10 million pixels, each of them containing three colour channels, resulting in 30million features/dimensions. In each colour channel, there are 255 possible values[11]. The information entropy of this system is astronomically high, being able represent almost everything. If we were to randomly set the pixel values, we would have the same chance of finding a sheet of paper floating in space as of finding a picture of something real, such as a dog. However, if we found the sheet of paper, we would find that all natural pictures and scientific pictures actually exist on this one piece paper, very near to the picture of the dog. This small fraction of all possibilities contains all natural shapes, gradients and patterns. To strip away the metaphor, photographic images have a relatively tiny set of real dimensions compared to the feature space we display them on-screen [17].

The problem with huge dimensional spaces is that our model first has to find the sheet of paper floating in space - the tiny volume with in the high dimensional space - where reasonable outputs can be found. The model exists in a space where nearly all outputs are not just wrong, but are not even viable. The huge search space and the tiny space of reasonable outputs (the volume fraction) is known as the 'curse of dimensionality' [17]. In practical terms, increasingly large dimensional spaces usually require more data and more computational time to converge.

Let us revisit the geometry of the spiral [180]. Imagine we have placed a prize - the correct output - somewhere along the spiral. If we were to instruct a child to follow the spiral, we could describe the instructions as 'at coordinate x go to coordinate y'. By describing the space in the original dimensionality, we first have to get the child to the spiral, and then need to expend a lot of effort in keeping the child constrained to the spiral. If the child takes one step off the spiral, they have no chance of finding the prize. If we instead give the child instructions in terms of the embedded coordinate, we can simply, tell the child to walk along the spiral until they find the prize. The instructions are much easier, and the child can instantly discard all areas other than the spiral as possibilities. Training models in a reduced dimensionality is far easier for the same reason [17].

The most common dimensionality reduction used in the context of wind energy is the simplification of structures by casting the structure as an eigenvalue problem. Usually, the long, slender blades of a wind turbine blade would be represented by beam elements [91]. We can take such a finite element representation, and push it it through an eigen-decomposition [91]. We would then use the mode shapes to describe the deformations of the blade rather than the individual dimensions [110, 108, 137, 136] [12]. It turns out the top three to five structural mode shapes contain nearly all of the information, and that the rest of the modes can be discarded without losing much accuracy [91]. This approach is used in aeroelastic simulation software, such as NREL's FAST [5]. The eigenvalue decomposition is a particular example of a singular value decomposition that forms the basis of many of the linear algebra dimensionality reduction

[11] There is a good interview where this concept is used to explain high dimensionality and entropy, in reference to transfer learning [200]

[12] Associated publications

methods.

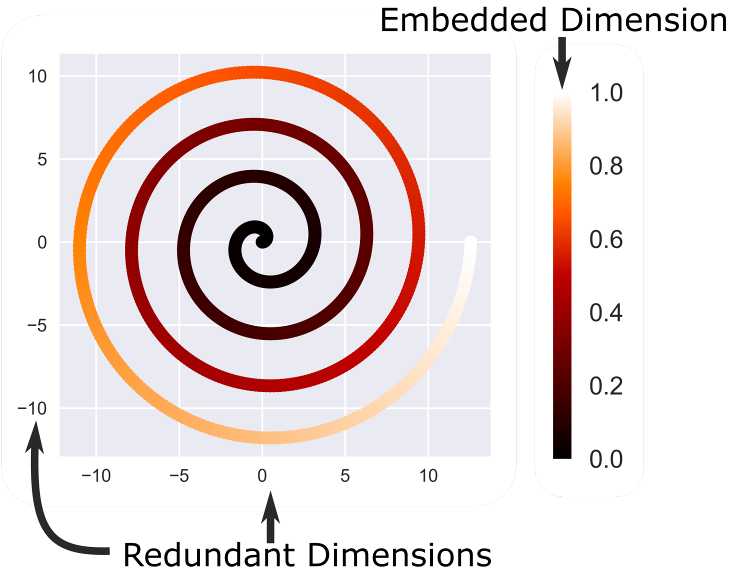


**Figure 2.8:** Removing the redundant dimensionality of a spiral using local linear embedding (an adapted from the lecture notes of [180]

The reduced-order representations can be transferred back into their full representations, which is the point. A good example of this uses a sparse set of sensors to predict an entire wind field. We first put simulated wind data into reduced-order space, and then use a sparse number of sensors to predict the reduced coordinates - a much easier task than predicting every grid point of the wind field. The first implementation of this approach from Annoni et al. [8] used the Manohar et al. [127] method, which uses a dynamic mode decomposition to get the mode shapes. The mode shapes are pushed through a second proper orthogonal decomposition using the QR pivoting method. The nice thing about the QR pivoting algorithm is that it ranks which of the sensors contribute the most predictive power within the coordinates in the reduced-order system. We then have an automatic greedy sensor selection, which does not result in a combinatorial search. Reduced-order representations are very powerful when used judiciously.

### 2.2.5 Density Estimation

The simplest forms of density estimation are probably familiar to most researchers in this field. We can take a dataset and fit a function, such as a normal distribution to present the density of the data in a concise way. We can compare the mean and variance of two different noisy datasets to measure changes in the population. The normal distribution is an example of a parametric distribution, which is defined by an equation that has just a few coefficients that can be learned from the data. However, in many real-world datasets, such simple functions fit the data poorly.

For complicated data, we can use non-parametric density estimations, such as the kernel density approximation [17]. Ice-fall from wind turbines presents a good use-case for non-parametric distributions. When building wind turbines in areas with ice, it is possible for ice chunks to be thrown or to fall from the blades and nacelle. It is necessary to ensure that a threshold probability of injury or death is not exceeded. Probability distributions have been measured for the ballistic properties of possible ice chunks and wind conditions. A physics-based simulation in QBlade samples ice chunks and then launches them from the wind turbine, producing a scatter pattern of landing points. This two-dimensional scatter pattern is complicated, but a non-parametric distribution is able to represent the complicated geometry[13].

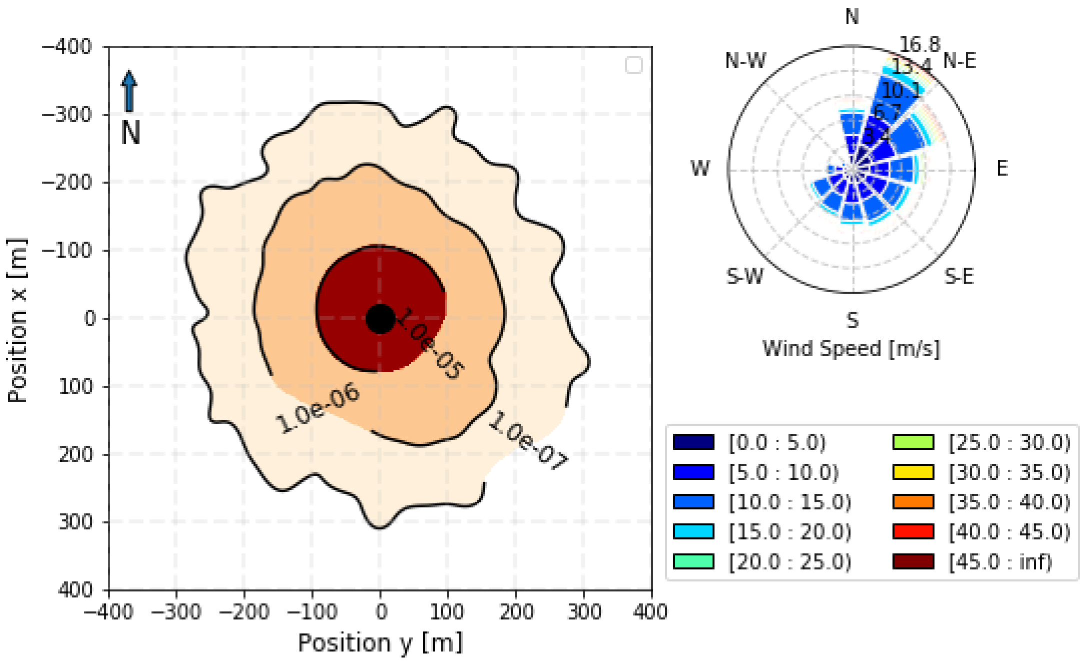


**Figure 2.9:** A kernel density estimate representation of the risk distribution from an icing simulation conducted using QBlade

This kind of risk assessment approach asks the general question: from the possible outcomes, how likely is each outcome? For instance, if we were to estimate the energy yield from a wind farm, we could sample from distributions of wind speed, wind direction, temperature, breakdown probabilities and power curves. We would then get a distribution of viable energy yields. The expected value might show a good energy yield, with the expectation of a good profit. However, the distribution of possible energy yields might show that, for 20% of the time, the wind farm would run at a loss, with break-downs appearing in moderate numbers - good grounds to question the project's viability. We use the knowledge of such distributions to make decisions [17].

[13] These references are associated publications [115, 116]

Distributions can also help us to simply understand the underlying structure of the data. When airfoils experience stall, the boundary layer becomes very, unsteady and cycle-to-cycle variations explode. Figure 2.10, illustrates violin plots of the pressure over an airfoil experiencing dynamic stall. A violin plot is a kernel density estimation [17] of the probability [14]. We can see that, as the vortex traverses the airfoil, the probability distributions become bimodal and skewed. In this case, a density estimation shows clearly that, before trying to fit a model, we should understand the process itself has a massive underlying level of variability.

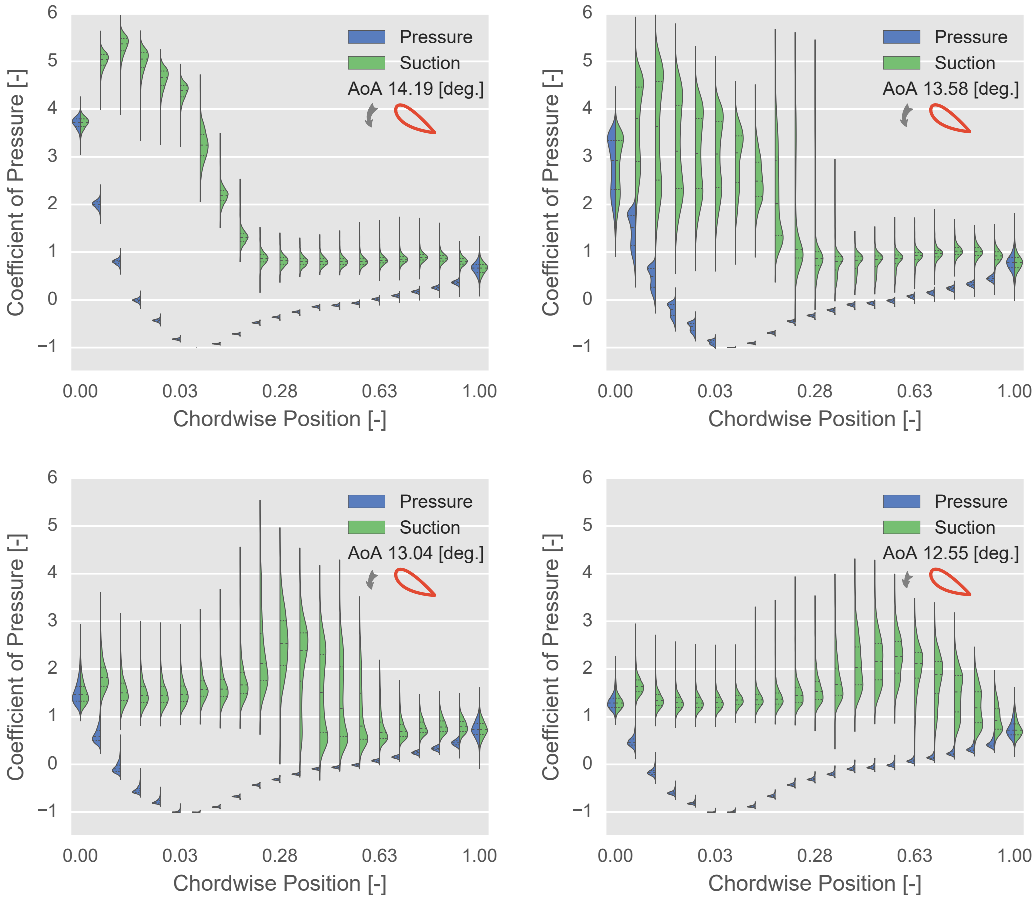


**Figure 2.10:** Cycle-to-cycle variations in dynamic stall in the pressure data

### 2.2.6 Forecasting

When people think of forecasting, they most often think of weather. In this case, this is entirely appropriate because successfully selling the energy from a wind farm requires a high-quality weather forecast. Forecasting can yield a continuous output (regression) or a label (classification).

[14] These figures and analysis are from the associated publication of Lennie et al. [114]

The difference is that forecasting tries to predict a future state based on the current and past states. Forecasting is a very well developed field due to econometrics, where there are significant incentives for predicting the price movements of stocks and bonds before one's competitors [167]. One peculiarity of forecasting is how careful one has to be in ensuring that the validation and test data are not accidentally included in the training set [167].

In the context wind energy, there are several challenging applications for forecasting. One challenging example is forecasting ice growth on wind turbine blades [30]. This task is particularly difficult because of the sensitivity of the ice formation to conditions such as the surface properties of the blade and the tower, as well as the humidity and temperature, which can change over the altitude difference that occurs in a single revolution. In this case, a more accurate ice forecast would help to predict the power output, and would help operators to optimise the use of de-icing and anti-icing systems. A second challenge is that obtaining a labelled set of training data is not trivial. Instruments can develop ice earlier, and that ice can melt earlier than it does on the blades [30], and other methods based on deviations from the power curve [44], or in shifts of the rotor blade's natural frequencies from the added weight, can react slowly [198]. Anti-icing/de-icing systems are expensive to run, and a forecast would have to be accurate within a time window of roughly 10 minutes; however, most ice forecasts are probabilistic, usually over the course of a day [30]. In summary, we have a system sensitive to input conditions, with imperfectly-labeled and unbalanced training data, unbalanced training data and the system requires a relatively high-quality output. This is the types of problem that is very challenging for machine learning.

### 2.2.7 Association Rules

Association rules attempt to find patterns in data [38]. A common use is analysing people's shopping baskets; with enough data, it is possible to infer information from the contents. Let us say someone has in their basket mince-meat, bread rolls, lettuce and cheese. From basic human knowledge, we can infer that the person is not planning to cook tomato soup, but more likely a hamburger. Association rules might tell us that the person is also likely to buy ketchup. If you have ever shopped on a website and see the words 'People who bought this item also bought...', then you have seen an association rule [38].

A possible application in the context of wind energy would be in the analysis of faults and repairs. An obvious example would be wind turbines where the lubrication system has a faulty seal, will often need some cleaning. Less obvious examples may lurk in the data. We may find something more exotic, such as a pitch misalignment leading to cracks in a specific area of a blade where a composite layer drop-off occurs. Such a connection would be difficult for a single technician to make, especially if they were not working on both the pitch alignment and rotor blade inspection. However, such associations could be mined from fleet-wide datasets, with something like 'Wind turbines that had this fault also had...'.

### 2.2.8 Generation

Generation is the concept of teaching a system to imitate a set of rules with a certain amount of randomness, so that it can create new, believable examples [41]. A recent demonstration of this concept took a set of celebrity faces and pushed them into a generative adversarial network (GANS). The network then created new, believable celebrity faces. Such a network is trained in such a way that there is a second neural network attempting to label pictures as real or fake [41]. This means that the generated pictures end up as photo-realistic [41].

A simpler form of generation, in the context of wind energy is responsible for creating turbulent wind fields. Fake turbulent wind fields are required for load-case simulations under realistic turbulent conditions. The Mann [125] and Veers [205] models are good examples of this. These methods are based on manually selected equations rather than purely data-driven, but they do the same job.

## 2.3 Why wind energy is challenging for machine learning

Wind energy is a challenging area to apply machine learning to. The following anecdote provides an insight into the experience of working with wind-energy data. When the wind industry began to use vortex generators to improve aerodynamic performance, a reasonable question was asked: how much annual energy production will I get from this product?[15] The answer is roughly 2%, but it is not trivial to arrive at this answer. Why is this? This is a hypothesis test where we have to reject the null hypothesis that the performance of the wind turbine is the same before and after the installation. If we use frequentist statistics, we would usually accept or reject a hypothesis based on a P-value, but here is the problem, as the variance of the data increases[16], the data with and without vortex generators will increasingly overlap. Now consider that we are searching for an effect that is only 2%, a very small confidence interval. This means we will need a lot of data, or data with a very low variance.

So what kind of data is available to check the performance of a wind turbine? SCADA data, measured by the wind turbine. Here we encounter our first problem. The wind speed experienced by each wind turbine is usually measured by an anemometer on the nacelle, in the middle of the separated wakes of the blade root regions [17]. The anemometers are calibrated using CFD and against meteorological mast measurements, where possible. Even under ideal conditions, an

[15] The example described in this paragraph comes from Hwangbo et al. [85], extra details come from discussions with the authors of that paper

[16] For the sake of this discussion, normality is assumed

[17] Vortex generators affect the separation of the root region, complicating the analysis further

anemometer is only a point measurement, representing an entire wind field. Usually, we assess a wind turbine's performance in terms of its power curve, the key input variable being wind speed. The most important feature for this analysis is very noisy.

The second problem is that most SCADA systems have classically recorded data averaged over 10-minute intervals, partly due to the wind spectral gap [19]. This is unfortunately a long time interval for time-resolved data, and such measurements result in large uncertainties on all axes of the power curve. Furthermore, such a low sampling rate decreases the samples available. The first method presented by Hwangbo et al. [85], using the ten-minute interval data eventually arrived at the correct answer, but with large error bars or a large confidence interval [85]. Let us recall the answer - 2%; very good machine learning resulted in error bars of roughly 4% [85]. This means that there is still a relatively high chance that the change we are seeing is from random sampling effects rather than being a real effect.

Finally, a second method used shorter interval data with simpler methods. The key was getting a good set of baseline data enable a comparison between two turbines. Vortex generators were installed on one set of blades, with the other remaining as a reference - in essence a giant anemometer. In data science, this is known as an A/B test [17]. With higher-frequency data and reference to another turbine, it was possible to get an answer with high enough confidence to satisfy the customers. The lesson is that basic methods using good-quality data and features will beat the best machine-learning methods on uncertain data.

Unfortunately, many forms of wind-energy data have large deviations. Rotor blades are massive structures, and the resulting tolerances are orders of magnitude larger than what is expected in gas turbines. The constituent glass fibre mat are hand-laid into the molds, which results in waves along with other defects; which are highly detrimental to the strength properties of the blades [171]. Once the two halves of the molds are filled with resin and cured, the massive blade sections are glued together. Bonding lines are one of their most critical failure points [35]. Thus, the production process results in small and large defects. This means that, across a family of blades, a large amount of variation is inherent [35]. Also, variations in the parameters of the input materials can introduce uncertainty into a blade's expected fatigue life [35, 10].

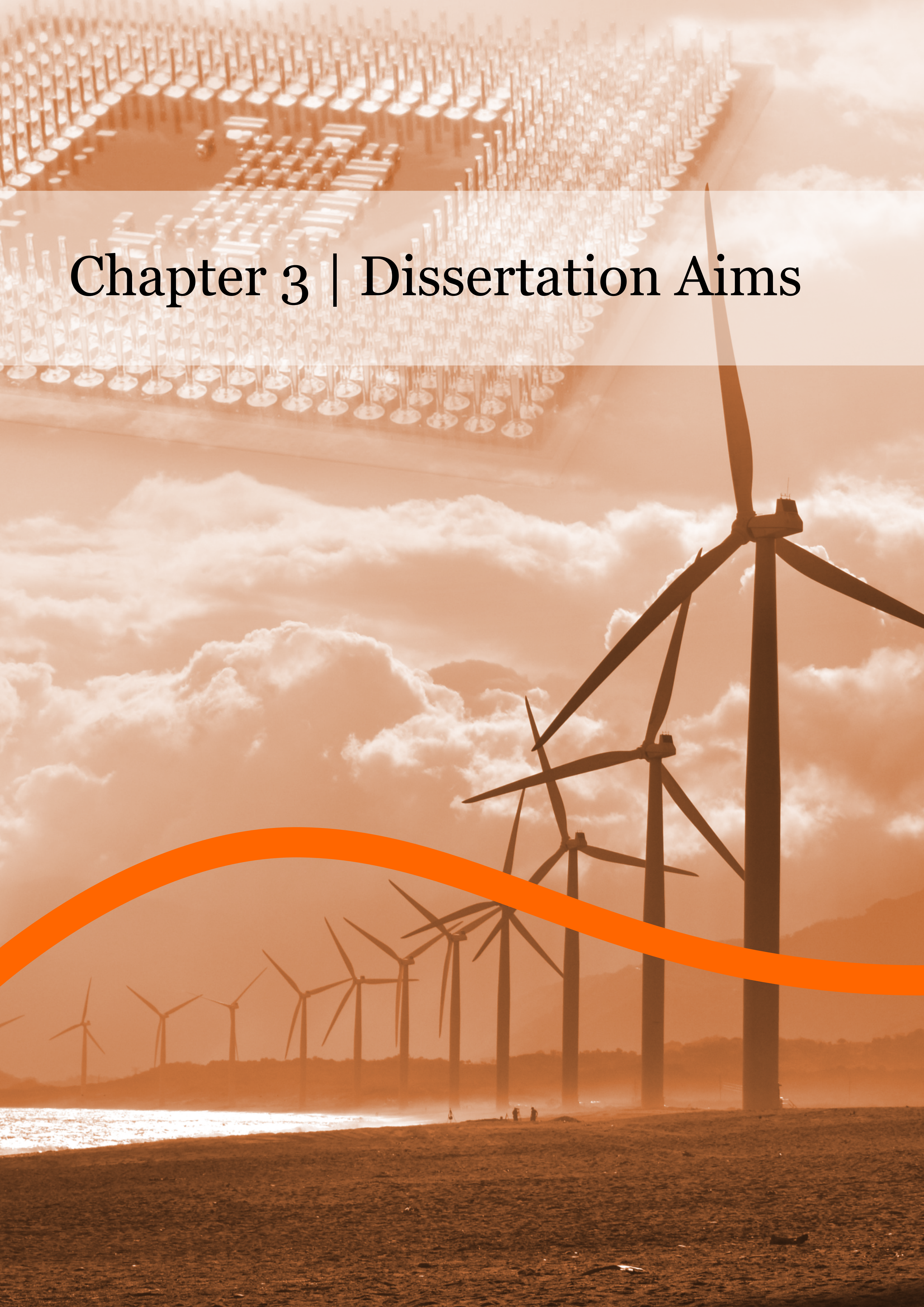
Now, let us imagine that we would like to predict delamination occurrences in the rotor blade's trailing edge. In order for the results to be useful, we would like to predict this early in the process, to initiate repairs before damage propagates. After the blades have been in production for some time, the maintenance personnel may begin to note faults during their inspections, these potentially being years apart. Thus, the blades are highly variable, and the onset of failure is only known to be sometime since the last inspection. Again, converging such a model is possible, but only with a huge amount of data. Unlike for the Silicon Valley giants, billions of data points are not available because there are not billions of rotor blades in service.

Engineers who work with wind turbine data often complain about the quality, not only because of the uncertainty of the input features, such as the wind speed from the nacelle-mounted anemometer, but also because of the uncertain baseline. A common problem is that repairs can fix leading edge tapes, yaw errors or pitch misalignments, or perhaps vortex generators are installed. A huge dump of data arrives from the operator, with no mention of the change [98]. At some stage during the time-series, the wind turbine can seem like a different machine. Eventually the engineers will find the problem and 'clean' the data using additional information from the maintenance logs. These steps take time, and there can be other small details like non-standard data formats, meaning that even parsing the data becomes a manual chore. Further technical issues that arise from the nature of the data itself not just the measurement techniques [80]. However, if wind turbine analytics are to be efficiently scaled, adopting a different culture around data quality seems necessary. Quaeghebeur and Zaaijer [164] provided a useful review of the quality of SCADA data and the common problems. Conversations with data engineers have indicated that some of the major companies are taking the data-quality issue seriously, but recommendations from Quaeghebeur and Zaaijer [164] indicate that in many cases, there is an almost primitive approach to data collection.

These examples do not represent an overarching call for pessimism regarding machine learning in the context of wind energy. However, the fact is that the largest data source in wind energy - operational SCADA data - is not of a high quality. Dealing with SCADA data requires big data infrastructure and expertise. Machine-learning algorithms can deal with noise, but poor-quality data makes the job much more difficult, and sets an upper limit on the eventual quality of the model. If this was the only approach available, that *would* be cause for despair.

There are better problems to take on. One example is the use of text analysis to extract key information from maintenance reports without having to manually read them. This task has its challenges, including abbreviations and spelling mistakes in the reports; nonetheless the project was successful [174], resulting in a drastic reduction in time spent reviewing maintenance reports. The examples given here have been largely to do with predictive maintenance - a highly valuable, but difficult, problem to take on. However, these are not the only open problems in wind energy that can be solved by machine learning. One of the themes of this study was to demonstrate other uses of machine learning that could harvest the low-hanging fruit.

# Chapter 3 | Dissertation Aims



# Chapter 3 Dissertation Aims

So far, I have reviewed the state of affairs with respect to wind energy and data science. Now it is possible to put this study into context. In the discussion about the challenges of wind energy, there were recurring themes, with engineering models needing improvement, high-fidelity models often being expensive to run and data sometimes not being well described by basic approaches. Solving these problems requires a number of lines of attack:

1. Improved simulation tools to match the expanding needs of wind energy;
2. Better engineering models and analysis methods that meet modern challenges; and
3. Better utilisation of machine learning to help mine and model data.

The publications in the following sections give examples of these approaches.

In this thesis, I demonstrate how a combination of approaches can be used to increase the quality of our simulations and data analyses. The first paper demonstrates the use of the FVLLT aerodynamics simulation approach for floating-platform wind turbines. This involved the simulation of a case that fell outside the standard BEM engineering model. In the paper, I developed an engineering method for estimating the instantaneous aerodynamic damping as a demonstration of how engineering tools, even with their limitations, can still be useful.

The second paper is a demonstration of the kinds of problems that are very difficult to solve with traditional engineering models. This experimental study investigated the shedding patterns behind a wind turbine blade at very high angles of attack. The results were so complicated, that it motivated the third paper, in which I used machine learning to dissect dynamic stall. From the machine-learning results, it became apparent that cycle-to-cycle variations of stalled airfoils are significant. Any approach that simulates airfoils has to first address, what exactly should be reproduced? The mean?!

In these three papers, I provide cases of how we can use engineering models and data science in wind energy. I highlight where engineering models and high-fidelity simulations can be effective, and where they are not. Based on the conclusions from these papers, and my associated research, I develop seven new strategies for developing numerical models for wind energy. The new methods, models and measurements will have a large impact on load case simulations of wind turbines especially for previously difficult to simulate cases, i.e. floating platform wind turbines or parked wind turbines. The rotor plane instantaneous aerodynamic damping model may also find use in helicopter aerodynamic applications. The grand concept of this study is presented in Figure 3.1.

In the next immediate section, I will briefly describe the contents of each of the attached publications before each of the papers is presented in full.

**Lennie, M**., Marten, D., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2017). Modern methods for investigating the stability of a pitching floating platform wind turbine, Wind Energ. Sci., 2, 671–683, https://doi.org/10.5194/wes-2-671-2017, 2017.

**Lennie, M**., Selahi-moghaddam, A., Holst, D., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2018). GTP-17-1616 Vortex shedding and frequency lock in on stand still wind turbines, a baseline experiment. Journal of Engineering for Gas Turbines and Power, 140(11), 112603–112603. https://doi.org/10.1115/1.4039818 Republished with permission of ASME; permission conveyed through Copyright Clearance Center, Inc.

**Lennie, M**., Steenbuck, J., Noack, B. R., & Paschereit, C. O. (2019). Cartographing dynamic stall with machine learning, Wind Energ. Sci., 5, 819–838, https://doi.org/10.5194/wes-5-819-2020, 2020.

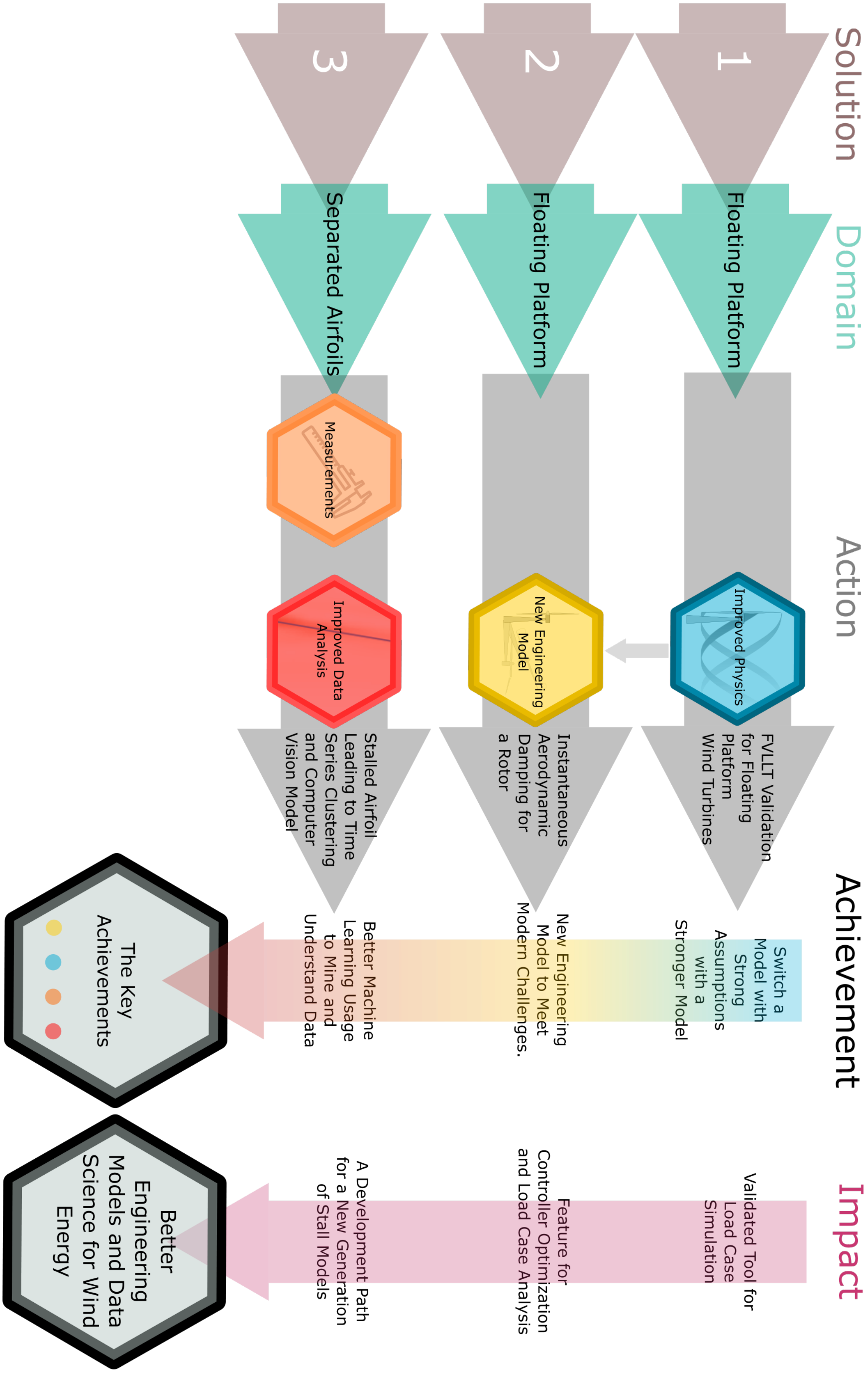

Solution
Domain
Action
Achievement
Impact
1
Floating Platform
Improved Physics
FVLLT Validation for Floating Platform Wind Turbines
Switch a Model with Strong Assumptions with a Stronger Model
Validated Tool for Load Case Simulation
2
Floating Platform
New Engineering Model
Instantaneous Aerodynamic Damping for a Rotor
New Engineering Model to Meet Modern Challenges.
Feature for Controller Optimization and Load Case Analysis
3
Separated Airfoils
Measurements
Improved Data Analysis
Stalled Airfoil Leading to Time Series Clustering and Computer Vision Model
Better Machine Learning Usage to Mine and Understand Data
A Development Path for a New Generation of Stall Models
The Key Achievements
Better Engineering Models and Data Science for Wind Energy

**Figure 3.1:** The grand concept

## 3.1 Modern methods for investigating the stability of a pitching floating platform wind turbine

Floating-platform wind turbines are a challenging problem in wind turbine modelling. The rotors are large and flexible, and the rotor plane can move, leading to much larger deflections than would be expected from an onshore machine. For floating platforms, we also need to take into account a new dimension of stability because of the platform. Furthermore, it is not clear that the simple BEM approaches normally used are suitable for such cases.

For this paper, I undertook two major tasks:

1. Validated the FVLLT code in QBlade against higher fidelity simulations of an airfoil rotor undergoing pitch; and

2. Deriving a new formulation for the instantaneous damping of a rotor under going fore-aft motion.

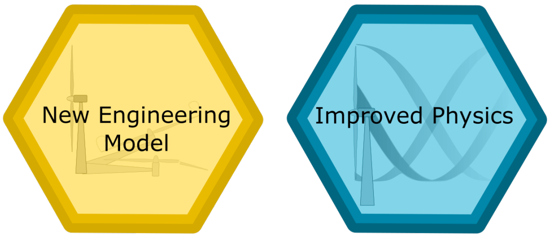


**Figure 3.2:** Outcomes

The FVLLT code performed well against the higher- and lower-fidelity simulations, although there were some small discrepancies between the different simulation tools' predictions of thrust. This probably originates from the strong assumptions made by some models, it serves a reminder to carefully tune the wake simulation with FVLLT codes.

The new formulation uses the Hilbert transform to create a phase-difference estimate of the rotor motion and it's thrust. After a lengthy derivation and normalisation, it is possible to use this phase difference as a representation of the aerodynamic damping of the system. By analogy, we can think of it as a single degree of freedom flutter. This can only occur when we account for the delay effects between the pitching of the airfoil and the moment. Usually, we do this through Theordorsen type functions [195]. The Pitt-Peters formulation of the thrust forces incorporates the same wake memory effect as the Theordorsen function [163]. This may be an unsatisfying explanation, but it can be alternatively formulated as: rotor planes experience added mass and circulatory terms in the same way as an airfoil, though for slightly different reasons. This analogy means that we can take the Bowles et al. [22] formulation as a model for looking at the rotor plane. The instantaneous damping term helps us understand how control actions and non-linear wake effects create an unstable or stable platform.

This paper demonstrates the importance of a new generation of simulation tools that are needed to take on challenging cases like floating platform wind turbines. Furthermore, the derivation of the instantaneous aerodynamic rotor damping is a perfect example of how an engineering approach can give us a useful tool, even in the age of data-driven models.

## 3.2 Vortex shedding and frequency lock in on stand still wind turbines, a baseline experiment

When wind turbines are under construction or maintenance, the rotor is usually locked. It might also be that there is no power, leading to the turbine being in a locked yaw position. Under these conditions, the blades of a wind turbine can experience very high-angle-of-attack inflows that can induce vortex induced vibrations. This means that the blades are collecting fatigue cycles before they are even generating electricity. Vortex-induced vibrations is the phenomenon of a structural frequency and a shedding frequency synchronizing over a certain bandwidth; this leads to large amplitude vibrations.

In this study, I performed extensive wind-tunnel experiments on a blade at large angles of attack. The measurements are, to my knowledge, the only measurements of a twisted-tapered blade at high angles of attack. The aim was to provide baseline results for the wake behaviour that could be translated into an engineering-type reduced-order model. These results are the key outcome. The most obvious candidate model would be simply identifying the Strouhal number of a blade and working out the frequencies from there. I even tentatively made that recommendation in this paper.

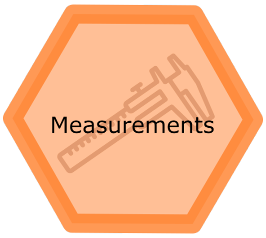


**Figure 3.3:** Outcome

The problem was that this kind of model would be over-fit from the outset, meaning that it was very difficult to generalise the patterns observed in these experiments. This is because there are *patterns*, not a single pattern. This paper explores the complexity presented by these high-angle-of-attack flows. The shedding behaviour changes with angle of attack, and, unfortunately other research indicates the same with azimuth [183]. This data set lead me to realize that certain physical processes can't be squashed into a reduced-order system, because there is no underling low order process! This data inspired me to look at stall behaviour from a data-driven, machine-learning perspective. The data serve as baseline for CFD and model calibration.

## 3.3 Cartographing dynamic stall with machine learning

We saw in the previous section that stalled airfoils are incredibly difficult to interpret. In this paper, I take a deep dive into how stalled airfoils are highly variable. To do this, I used several advanced machine-learning techniques.

The first of two methods developed employs dynamic time-warping as a distance metric to create clusters via a hierarchical clustering algorithm. The many repetitions of a dynamic stall experiment are split, and then clustered together into groups of similar behaviours. By doing this analysis, we found that the shedding mechanisms that create dynamic stall are not reliable for a given set of input conditions. In some repetitions, secondary vorticity disappears, whereas in others, the secondary vorticity is stronger than the primary. The method developed in this paper represents a new approach to inspecting cycle-to-cycle variations in airfoils.

It is useful to look broadly at groups of similar behaviours, but it is also useful to extract features from data repetitions to allow a direct comparison. As it convects over an airfoil, a dynamic stall vortex shows up on a pressure-versus-time plot as a strong diagonal stripe . Extracting this feature from data looks trivial - it's a straight line - but creating a rule that generalises well is not possible. In this paper, we implemented a highly sophisticated deep learning computer vision model using transfer learning to automatically extract such features from the data.

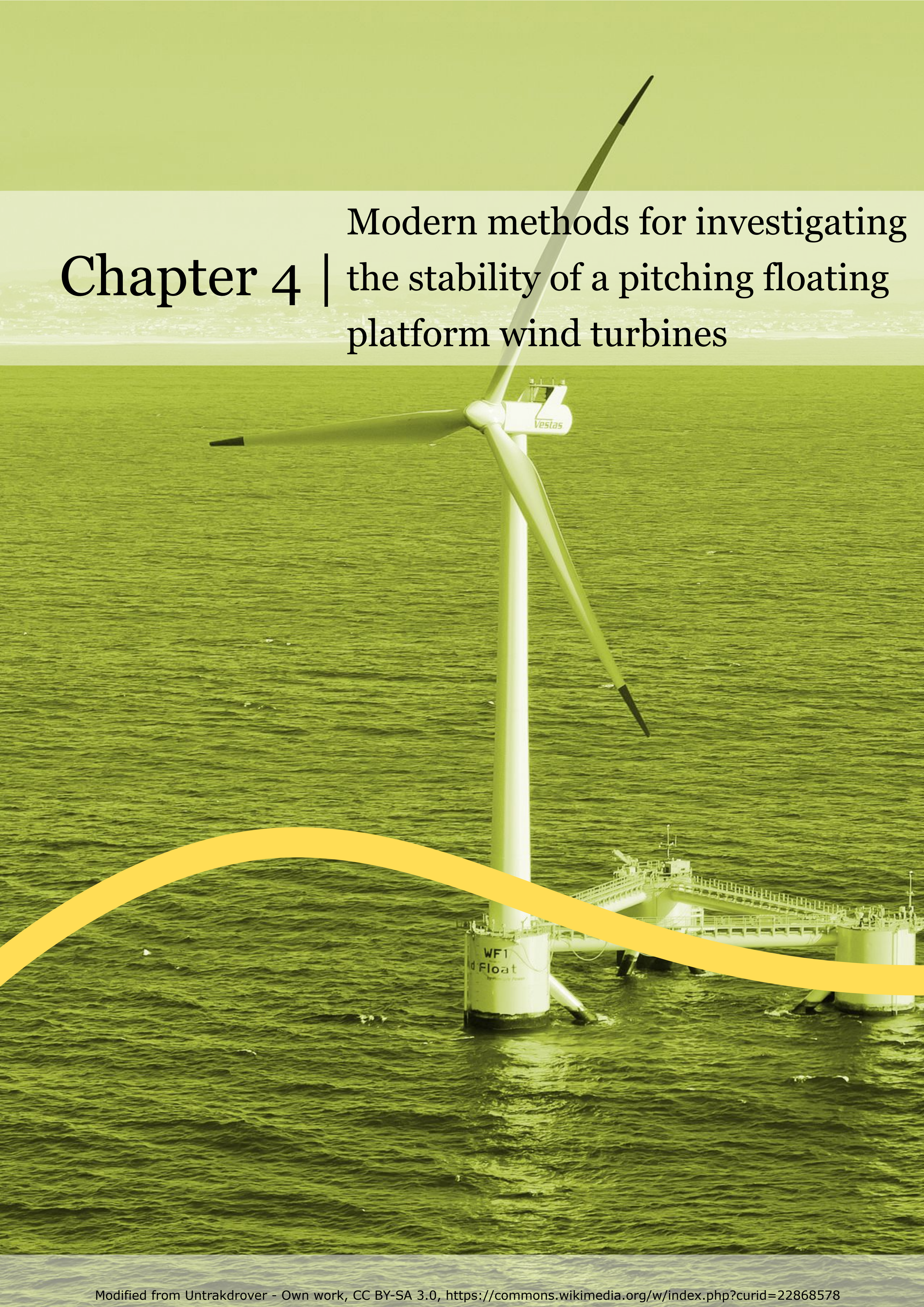
The key achievements of this paper are:

1. time series clustering algorithm for airfoils

2. a computer vision machine learning model for dynamic stall convection detection.

The achievements should be considered as an important step towards unleashing machine learning and data science on the field of wind energy.

**Figure 3.4:** Outcome



# Chapter 4 | Modern methods for investigating the stability of a pitching floating platform wind turbines

Modified from Untrakdrover - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=22868578

Wind Energ. Sci., 2, 671–683, 2017
https://doi.org/10.5194/wes-2-671-2017

eawe european academy of wind energy
Open Access
WIND ENERGY SCIENCE

# Modern methods for investigating the stability of a pitching floating platform wind turbine

**Matthew Lennie, David Marten, George Pechlivanoglou, Christian Navid Nayeri, and Christian Oliver Paschereit**

Chair of Fluid Dynamics, Hermann-Föttinger-Institut, Technische Universität Berlin,
Müller-Breslau-Str. 8, 10623 Berlin, Germany

**Correspondence:** Matthew Lennie (matthew.lennie@tu-berlin.de)

Received: 15 December 2016 – Discussion started: 21 December 2016
Revised: 18 September 2017 – Accepted: 1 November 2017 – Published: 22 December 2017

**Abstract.** The QBlade implementation of the lifting-line free vortex wake (LLFVW) method was tested in conditions analogous to floating platform motion. Comparisons against two independent test cases using a variety of simulation methods show good agreement in thrust forces, rotor power, blade forces and rotor plane induction. Along with the many verifications already undertaken in the literature, it seems that the code performs solidly even in these challenging cases. Further to this, the key steps are presented from a new formulation of the instantaneous aerodynamic thrust damping of a wind turbine rotor. A test case with harmonic platform motion and collective blade pitch is used to demonstrate how combining such tools can lead to a better understanding of aeroelastic stability. A second case demonstrates a non-harmonic blade pitch manoeuvre showing the versatility of the instantaneous damping method.

## 1 Introduction

The proliferation of large wind turbine rotors has been accompanied by the need for accurate and computationally inexpensive aeroelastic simulation tools. For aeroelastic simulations, the aerodynamics of the wind turbine are most typically calculated using methods based on blade element momentum (BEM). In the scenario of offshore wind, particularly when designing for floating platforms, the significant motion of the rotor leads to complicated aerodynamics. Sebastian and Lackner (2013) have made a convincing case that, even with secondary correction factors, floating platform wind turbine aerodynamics exceed the capabilities of BEM-based simulation methods. The main reason is that a floating platform wind turbine will dynamically pitch and yaw. As BEM does not explicitly solve the flow pattern of the wake, it is simply not possible to accurately represent such behaviour.

The lifting-line free vortex wake (LLFVW) method uses non-linear polar data[1] to calculate the blade forces coupled with a free vortex wake formulation and serves as a good method for simulating cases in which large rotor displacements and yaw misalignments occur (see Fig. 2). Recently, the implementation of an LLFVW code was completed and included in the QBlade wind turbine simulation code (Marten et al., 2015). Simultaneously to this study, the LLFVW solver was extended to include an unsteady aerodynamics model[2] and coupled with the structural formulations of the FAST framework (Wendler et al., 2016; Saverin et al., 2016). In this paper, a comparison is made between the LLFVW code and existing literature comparisons in which higher-order aerodynamic simulation techniques were used, i.e. URANS CFD (Tran et al., 2014; Sebastian and Lackner, 2013). The comparisons and further test cases are made using the NREL 5MW reference turbine undergoing prescribed harmonic motion (see Fig. 1; Jonkman, 2013).

[1] Including viscous effects such as separation.

[2] LLFVW formulations inherently account for attached flow unsteadiness; the unsteady aerodynamic model mentioned here only includes terms for detached flow and leading edge vorticity. The details are given by Wendler et al. (2016).

Published by Copernicus Publications on behalf of the European Academy of Wind Energy e.V.

After the validation of the LLFVW code for simulations involving a moving rotor plane, the aerodynamic damping of the rotor is investigated. For this analysis, a new formulation is presented for the instantaneous aerodynamic damping of the fore–aft motion degree of freedom. The formulation is a modification of an existing formulation that was first presented by Bowles et al. (2014) and Corke and Thomas (2015) and later applied by Lennie et al. (2016) to an airfoil with microtabs. For the first time, this new formulation makes it possible to look at the aerodynamic damping throughout the pitch cycle of the wind turbine as opposed to the traditional approach in which only cycle-averaged values are inspected. Such a formulation is particularly useful for analysing aeroelastic instabilities for which limit cycle oscillations are present. Limit cycle oscillations will have cycle-averaged values that are neutral but could have occurrences of highly negative damping. Using this method on LLFVW data makes it possible to understand the aeroelastic thrust stability of the rotor without the heavy linearization of the aerodynamics applied in most stability analysis techniques. It is also a useful way of understanding the full effects of controller wind turbine interactions. An example will be presented showing the effect of collective blade pitch cycles during fore–aft motion of the rotor.

## 2 Rotor motion

For the scope of this paper, two varieties of prescribed motion are considered. The first variety, pitching, is the more realistic representation in which the rotor plane undergoes both pitch and linear translation (see Fig. 1). The second variety, fore–aft motion, assumes that the rotor plane pitching component is insignificant compared to the influence of the linear translation. Within the scope of this study investigating the total rotor thrust, the difference between the assumptions is assumed to be small. There may be applications for which this assumption is unsuitable. In the comparisons, the same magnitude and type of motion is used as in the literature so that no additional assumptions were introduced.

## 3 Comparison cases

The QBlade LLFVW implementation has been previously tested for a range of standard HAWT and VAWT cases as can be found in the existing literature (Marten et al., 2015; Saverin et al., 2016). When the wind turbine starts moving relative to the steady inflow, the wake will become distorted. In the case of a harmonic movement, the wake will display harmonic contractions and expansions (see Fig. 2) which induce velocity onto the rotor plane. The publications mentioned above focused on verifying the performance of the QBlade LLFVW under stationary conditions and cases with yaw. This means that the battery of verifications undertaken should be extended to include cases in which platform motion is present, thus ensuring that LLFVW techniques are a suitable approach for floating platform wind turbine aerodynamics. A number of comparison papers have been sought from the literature that test a horizontal axis wind turbine rotor in prescribed floating platform motion. For the scope of this paper, only rigid body motion will be considered. The comparison will be undertaken by replicating the simulations from the literature which used higher-order methods.

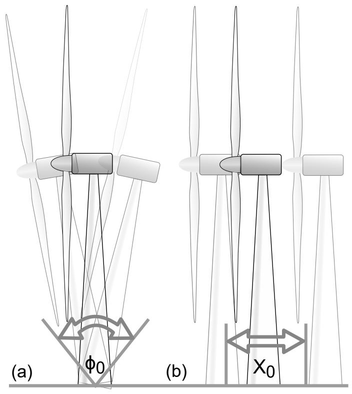


**Figure 1.** The two different assumed motions for the wind turbine: **(a)** wind turbine pitching and **(b)** wind turbine fore–aft motion.

Two different papers were used as a basis for comparison, both investigating the NREL 5MW reference turbine (Jonkman, 2013). Tran et al. (2014) compare a number of techniques with virtual blade motion using multiple reference frames (CFD-MRF) and real rigid body blade motion (CFD-RBM). The highest-order simulation is a 3-D unsteady Reynolds-averaged Navier–Stokes (URANS) CFD simulation with a $k-\omega$ shear-stress transport turbulence model. The blade rigid body motion was achieved using an overset grid, which is described at length including a discussion of the mesh convergence. The actual CFD simulations were conducted using the commercial code Fluent™ with Star-CCM+™ for meshing. It appears from the presented information that the simulations should be high quality and within the limitations of URANS.

Tran et al. (2014) also compared their results against lower-order simulations using the unsteady blade element momentum method. The Tran et al. (2014) implementation of the unsteady blade element moment (UBEM) method was taken from Hansen (2008) with corrections for tip losses, wake unsteadiness and unsteady aerodynamics. This particular implementation took the platform motion into account by changing the relative inflow velocity. Further comparisons were made using modified versions of FAST (Jonkman, 2015) from the National Renewable Energy Laboratory; one comparison used a momentum balance for the wake solution (FAST-BEM) and the other used generalized dynamic wake (FAST-GDW). In both cases, the structural modes were

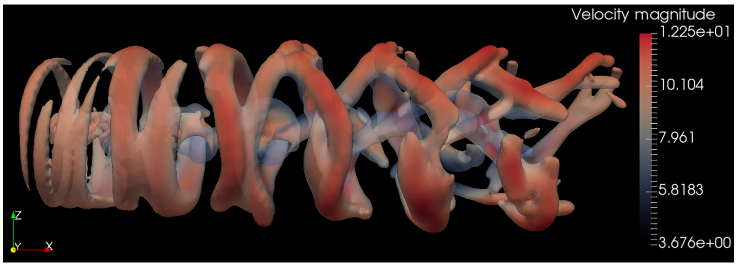


**Figure 2.** Snapshot of LLFVW simulation during pitching platform motion; vorticity isosurface of the wake coloured with velocity magnitude.

locked and no controller was used; for more complete details, see the original paper from Tran et al. (2014).

Tran et al. (2014) simulate the pitching of the wind turbine as shown in Fig. 1 (see the left-hand side). Two cases were simulated with platform pitching amplitudes of 1 and 4° and a constant harmonic pitching frequency of 0.1 Hz. The calculations were performed at a steady inflow speed of 11 $\mathrm{m\,s^{-1}}$ with a constant rotational speed of 12 rpm and a constant blade pitch angle of 0°. From this paper it was possible to compare thrust, power and integrated blade forces.

The QBlade simulations were run with the same conditions as described above. The unsteady aerodynamic model from Wendler was enabled without vortex lift as the wind turbine is operating at near rated speed without yaw (Wendler et al., 2016). The standard NREL 5MW model (Jonkman, 2013) was set up according to the definition[3].

The simulation settings can be found in Table 1. The FVLLW requires a number of input parameters which control the behaviour of the wake. The QBlade implementation of FVLLW lumps together wake elements in two stages in order to simulate a full-length wake without prohibitive computational costs. The wake age, full wake and fine wake parameters determine the positions at which the two stages of wake thinning occur and at which the wake is finally truncated. The parameters are described in terms of rotor revolutions to remove the dependency of the parameters on the tip speed ratio. The wake thin factor describes the extent to which the wake is thin. The initial vortex core radius is an important parameter for the de-singularization of the Biot–Savart equation. The turbulent vortex viscosity introduces diffusion to the wake. These parameters should be noted in attempts to reproduce the simulations performed in this paper. The full explanation of each of these parameters can be found in the paper from Marten et al. (2015).

**Table 1.** QBlade FVLLW wake simulation settings.

| | |
|---|---|
| Wake age in revolutions | 8 |
| Full wake in revolutions | 0.5 |
| Fine wake in revolutions | 4 |
| Wake thin factor | 2 |
| Initial vortex core radius | 0.20 |
| Turbulent vortex viscosity | 40 |
| Time step | 0.1 s |

The LLFVW (Fig. 2) shows moderately good agreement for all cases, which can be seen in Figs. 3, 4 and 5. It is interesting to note that steady and even unsteady BEM simulations, when compared to CFD or LLFVW results, underpredict the magnitude of the load cycle in most cases. The three classes of simulations compared all face limitations. The LLFVW and BEM cases both rely on 2-D polar data which are sensitive to measurement or simulation settings like wind tunnel turbulence or turbulence model. Lennie et al. (2015) and Eisele et al. (2013) have shown that the quality of the 2-D polar data is critical for power and load prediction. For the BEM-based methods, it is expected that the empirical corrections will struggle to represent the complicated fore–aft motion. The CFD solutions are based on unsteady Reynolds-averaged Navier–Stokes (URANS) equations. After conducting thorough verification and validations of multiple URANS solvers with multiple turbulence closure models, Rumsey showed that URANS-based models have a very limited ability to model cases with separation (Stangfeld et al., 2015; Rumsey, 2016). Furthermore, the tendency of URANS codes to smear vorticity will cause errors in the wake induction for cases in which the blades are modelled. With each of the simulation methods facing some sort of limitation, it is difficult to choose one method as the baseline or “most” accurate. Nonetheless, integrated blade forces agree well for the CFD from Tran et al. (2014) and the LLFVW (see Fig. 5). Discrepancies in thrust and power can be seen in Figs. 3 and 4, but they are of a reasonable magnitude.

In the publication chosen for a second comparison, Vaal et al. (2014) use a moving actuator disc CFD hybrid method, which allows for a good comparison of the unsteady wake induction between CFD and the LLFVW. The moving actuator disc model essentially places a moveable actuator disc into a CFD simulation (implemented in Fluent™). In practice this means that the actuator disc acts as a volume force onto

[3] The standard 5MW project file is available for download with the standard QBlade package.

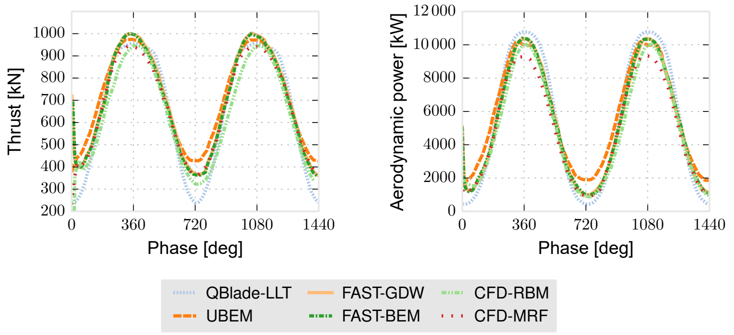


**Figure 3.** Thrust and power over phase angle for pitching platform motion (4° pitch amplitude). Comparison case: Tran et al. (2014).

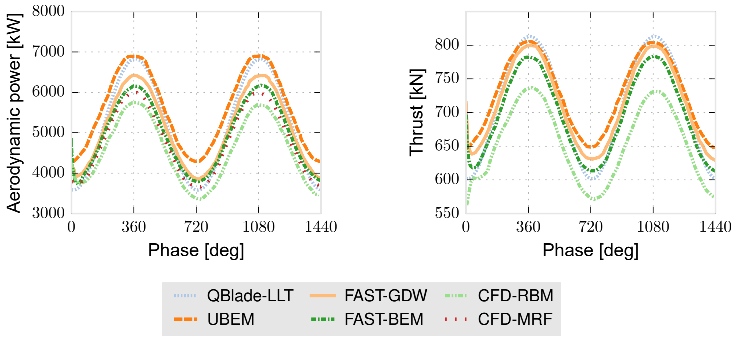


**Figure 4.** Thrust and power over phase angle for pitching platform motion (1° pitch amplitude). Comparison case: Tran et al. (2014).

the surrounding cells. It is argued by Vaal et al. (2014) that, because this method explicitly solves the wake rather than relying on simplified relations, the method should be more robust than commonly used methods such as the Pitt–Peters model (Pitt and Peters, 1983) or the Stig Øye model (Hansen, 2008). Vaal et al. (2014) present a number of investigations into the relative performance of the models; for this paper, the rotor plane induction is the most interesting to compare.

Vaal et al. (2014) undertook a sensitivity study showing the wake velocity before and after the rotor at different phases for different operating conditions. From this study, the authors choose the largest amplitude (16 m) of fore–aft movement (right-hand side of Fig. 1). The fore–aft motion was harmonic with a frequency of 0.08 Hz; the inflow speed was 11.2 m s$^{-1}$, the blade pitch was 0° and the rotor speed was a constant 0.2 Hz. Vaal et al. (2014) allowed several oscillations to pass in order to let the wake effects develop. The grid extended 10 rotor diameters upstream and downstream. It appears that the approach and settings used by Vaal et al. (2014) provide good-quality results for comparison.

The QBlade simulations were conducted again using the settings stated above, with prescribed linear rotor plane movement. Like Vaal et al. (2014), a number of oscillations were simulated before finally extracting the data. The comparison was made at the exact rotor plane for which the axial velocity could be sampled over an area determined by the rotor radius. There was not enough information provided by Vaal et al. (2014) to ensure the consistency of the sampling area for the induced velocity. In the context of a wake with expansion and contraction occurring, the assumptions have a distinct effect on the induction results. Therefore, no upstream or downstream comparisons were attempted. A rotor plane axial velocity field snapshot was taken at equally spaced points within the cycle. The results in Fig. 6 show that the rotor plane induction for the two methods matches well over the four snapshots. It therefore seems that there is a good agreement between the two methods for the most challenging test case presented by Vaal et al. (2014).

From the two verifications performed here it seems that the QBlade LLFVW simulation model produces results that are comparable to other higher-order or hybrid methods. These results and the results already published give a high degree of confidence in the simulation tool's ability to model wind turbines undergoing platform motion.

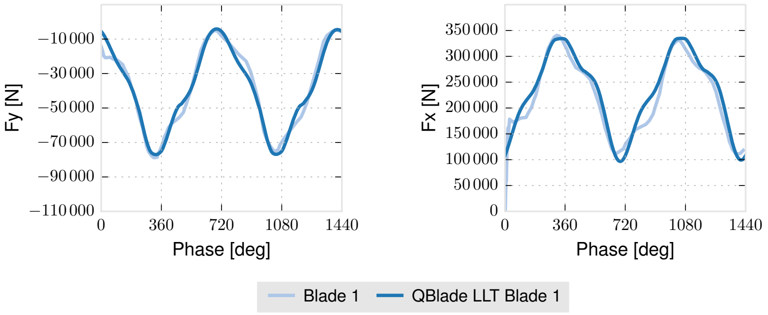


**Figure 5.** Integrated blade forces at blade root for pitching platform motion (4° pitch amplitude). Comparison case: Tran et al. (2014).

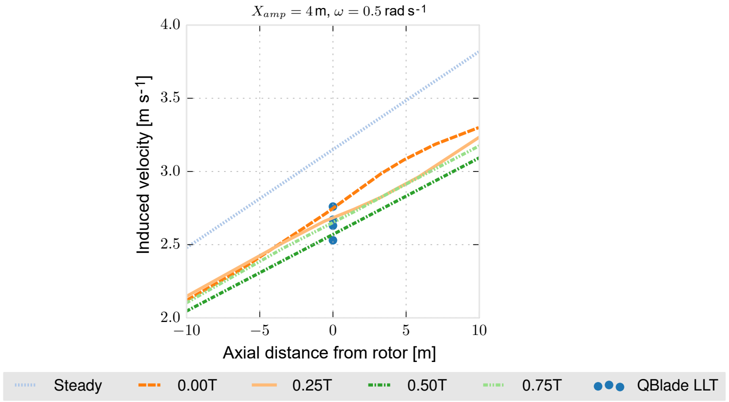


**Figure 6.** Induction plot compared to moving actuator disc CFD hybrid from Vaal et al. (2014).

## 4 Using the Hilbert transform method to obtain instantaneous aerodynamic damping of a translating rotor

The following section briefly outlines a reformulation of the instantaneous damping calculation method outlined by Bowles et al. (2014) and Corke and Thomas (2015). The original reference by Bowles et al. (2014) describes the original method in complete detail, and Lennie et al. (2016) provide an application of the method to an airfoil with microtabs and gurney flaps. In the reformulation described here, the instantaneous damping calculation is applied to the fore–aft motion of the whole rotor, a situation particularly interesting for floating platform wind turbines. It is assumed for this paper that a small pitch angle means that the linear motion will have a greater effect on the wake than the pitching of the rotor plane.

### 4.1 Cycle harmonic damping

Before setting out the derivation of the instantaneous damping coefficient it is first essential to set out the cycle total aerodynamic damping. The approach taken follows the derivation provided by Carta and Niebanck (1969) but for a rotor undergoing fore–aft motion rather than a pitching airfoil section. The authors would like to clearly acknowledge that the following derivation is a modification of existing concepts rather than a completely new derivation.

#### 4.1.1 Linear harmonic system in a vacuum

To begin the derivation, let us start with the homogeneous equation of the wind turbine oscillating in linear fore–aft motion (denoted as $x$):

$$m\ddot{x}^* + c\dot{x}^*(t) + kx^*(t) = 0, \tag{1}$$

where we make an assumption of harmonic motion thus taking

$$x^*(t) = x_0 e^{i\omega t}, \tag{2}$$

which then gives

$$\dot{x}^*(t) = i\omega x_0 e^{i\omega t}, \tag{3}$$

$$\ddot{x}^*(t) = -\omega^2 x_0 e^{i\omega t}. \tag{4}$$

Substituting these results into the equation of motion then gives

$$(-\omega^2 m - \overbrace{i\omega c}^{\mathbb{C}} + k)x^*(t) = 0. \tag{5}$$

As highlighted by Carta and Niebanck (1969), the damping terms are contained within the imaginary term of the equation of motion. For similar linear systems, the damping should be contained within the imaginary terms of the differential equation.

### 4.1.2 Linear harmonic system in air

Now by introducing the aerodynamic terms into the equation we arrive at the particular equation of

$$(-\omega^2 m - \overbrace{i\omega c}^{\mathbb{C}} + k)x^*(t) = T_\mathrm{U}^*. \tag{6}$$

Still following the logical steps set down by Carta and Niebanck (1969), the unsteady thrust force can be written as

$$T_\mathrm{U}^* = T_1\ddot{x}^* + T_2^*\dot{x}^* + T_3^* x^*, \tag{7}$$

where

$$T_2, T_3 \in \mathbb{C}, \tag{8}$$

but

$$T_1 \in \mathbb{R} \tag{9}$$

due to the rationale that $T_1$ represents the apparent mass terms of the system which are identified in terms of the instantaneous reaction forces of an impermeable disc in still air. Instantaneous reaction forces are in phase with the acceleration and therefore real. A similar rationale was used in the SDOF torsional airfoil oscillator formulation by Carta and Niebanck (1969), Bisplinghoff et al. (2013) and Scanlan and Rosenbaum (1951), this time by directly comparing to the Theodorsen theory (Theodorsen, 1935) in which a well-known distinction is made between the real and imaginary parts of the Bessel function. By substituting in the Cartesian forms of the thrust force

$$T_2 = T_{2\mathrm{R}} + iT_{2\mathrm{I}}, \tag{10}$$

$$T_3 = T_{3\mathrm{R}} + iT_{3\mathrm{I}}, \tag{11}$$

we arrive at the following equation:

$$\begin{aligned}(m - T_1)\ddot{x}^* &+ (c - T_{2\mathrm{R}} - iT_{2\mathrm{I}})\dot{x}^* \\ &+ (k - T_{3\mathrm{R}} - iT_{3\mathrm{I}})x^* = 0.\end{aligned} \tag{12}$$

By assuming harmonic motion and collecting real and imaginary terms, the equation is reduced to

$$\begin{aligned}&[(-\omega^2(m - T_1) + \omega T_{2\mathrm{I}} + k - T_{3\mathrm{R}}) \\ &+ i(\omega(c - T_{2\mathrm{R}}) - T_{3\mathrm{I}})]x^* = 0.\end{aligned} \tag{13}$$

By eliminating the mechanical damping terms, which are the terms present in a vacuum, the aerodynamic damping can be shown as

$$\xi = -\omega T_{2\mathrm{R}} - T_{3\mathrm{I}}. \tag{14}$$

This result will form a key step in the next decomposition.

### 4.1.3 Work done by a rotor in fore–aft motion

Let us now take a different decomposition of the thrust force into its constituent steady and unsteady parts[4].

$$T_\mathrm{TOTAL} = T_\mathrm{MEAN} + T_\mathrm{UR}\cos\omega t + T_\mathrm{UI}\sin\omega t \tag{15}$$

Two options exist for the normalization of the thrust force: the freestream velocity or with the inflow velocity thus accounting for the rotor movement. The former assumption simply implies that the unsteady coefficient will contain the freestream effects for the expected velocity ratios expected for wind turbine pitching movement and freestream velocities. This may cause some peculiarities in the appearance of the data similar to the lift coefficient overshoots seen by Müller-Vahl (2015) and Strangfeld (2015) in unsteady airfoil wind tunnel measurements. Nonetheless, the freestream velocity is taken as convention, meaning that the unsteady features will be wrapped up into the unsteady thrust coefficient.

This gives the coefficient form

$$C_\mathrm{TTOTAL} = C_\mathrm{TMEAN} + C_\mathrm{TUR}\cos\omega t + C_\mathrm{TUI}\sin\omega t. \tag{16}$$

The work performed over one cycle of rotor fore and aft motion can be given as

$$W_\mathrm{T} = \oint T_\mathrm{TOTAL}\,\mathrm{d}x \tag{17}$$

or in coefficient form

$$C_\mathrm{WT} = \oint C_\mathrm{TTOTAL}\,\mathrm{d}x. \tag{18}$$

Here the differential operator can be switched

$$\mathrm{d}x = x_0\sin\omega t\,\mathrm{d}\omega t, \tag{19}$$

and the integral range can be set from $0 < \omega t < -2\pi$ to capture a single cycle, finally giving

$$\begin{aligned}C_\mathrm{WT} = -\int_0^{2\pi} &[C_\mathrm{MEAN} + C_\mathrm{TUR}\cos\omega t \\ &+ C_\mathrm{TUI}\sin\omega t]x_0\sin\omega t\,\mathrm{d}\omega t.\end{aligned} \tag{20}$$

By assuming that the thrust force will be simple harmonic (or deviate minimally), evaluating the integral shows that the

[4] $T_\mathrm{Unsteady}$ is hereafter abbreviated $T_\mathrm{U}$.

real unsteady term and the mean terms are both eliminated during the integration, leaving

$$C_{\mathrm{WT}} = \pi x_0 C_{\mathrm{TUI}}. \tag{21}$$

This result can also be represented in the form

$$C_{\mathrm{WT}} = \pi x_0 C_{\mathrm{TUI}} = x_0 C_{T_0} \sin\phi. \tag{22}$$

This result is the second of the building blocks required to extract the aerodynamic damping from the measurement or simulation data.

### 4.1.4 The third decomposition of the thrust force

Let us now inspect the unsteady thrust force terms further. If we assume the thrust to be a sinusoidal time-dependent function, the unsteady thrust force can be given in the Cartesian form

$$T_{\mathrm{U}} = T_{\mathrm{UR}} + i T_{\mathrm{UI}}. \tag{23}$$

For a prescribed motion system, the earlier homogeneous equation (Eq. 13) can be written as a particular equation, thus giving

$$\begin{aligned} T_{\mathrm{U}} = T_{\mathrm{UR}} + i T_{\mathrm{UI}} = [(-\omega^2(m - T_1) + \omega T_{2\mathrm{I}} + K - T_{3\mathrm{R}}) \\ + i(\omega(C - T_{2\mathrm{R}}) - T_{3\mathrm{I}})]x. \end{aligned} \tag{24}$$

By equating real and complex terms we get

$$T_{\mathrm{UI}} = [(\omega(c - T_{2\mathrm{R}}) - T_{3\mathrm{I}})]x. \tag{25}$$

This result provides the key to extracting the aerodynamic damping coefficient from the thrust data. From earlier we know that

$$\xi = (-\omega T_{2\mathrm{R}} - T_{3\mathrm{I}}). \tag{26}$$

These two equations can be related through

$$\xi = -\frac{\mathrm{d}T_{\mathrm{UI}}}{\mathrm{d}x} \tag{27}$$

or written in the coefficient form

$$\Xi = -\frac{\mathrm{d}C_{\mathrm{TUI}}}{\mathrm{d}x}. \tag{28}$$

Using the earlier result of

$$C_{\mathrm{WT}} = \pi x_0 C_{\mathrm{TUI}} = x_0 C_{\mathrm{T0}} \sin\phi, \tag{29}$$

we can finally arrive at the conclusion that

$$\Xi_{\mathrm{CYCLE}} = \frac{\mathrm{d}\frac{C_{\mathrm{WT}}}{\pi x_0}}{\mathrm{d}x} = -\frac{1}{\pi x_0^2} \oint C_{\mathrm{TTOTAL}} \mathrm{d}x. \tag{30}$$

This equation will form the basis of checking whether the instantaneous equation formulation is correct. In practice it also provides a useful debugging tool for the code implementation.

## 4.2 Instantaneous damping derivation

In most formulations, certainly as shown above, only a cycle-averaged value of aerodynamic damping is found. Bowles et al. (2014) and Corke and Thomas (2015) provided a breakthrough on this front by using the Hilbert transform to get an estimate of the magnitude and phase of a signal. If we inspect the following equation from the earlier whole-cycle derivation, we may already see the general direction that such a method would follow.

$$C_{\mathrm{WT}} = x_0 T_0 \sin\phi \tag{31}$$

The instantaneous damping derivation begins with yet another form of the basic equation of motion for the prescribed fore–aft motion of a wind turbine rotor. The prescribed motion leaves only the aerodynamic forces,

$$m_t \ddot{x} + h^*(t)\dot{x}(t) + \kappa^* x(t) = T(t), \tag{32}$$

into which we can insert the apparent mass or inertia of air (Pitt and Peters, 1983),[5]

$$m_{t\mathrm{air}} = \frac{8}{3}\rho R^3, \tag{33}$$

and then the complex damping and stiffness terms can be described in polar form as

$$h^* = h_r + i h_{\mathrm{I}} = \overline{h} e^{i\gamma_1}, \tag{34}$$

$$\kappa^* = \kappa_r + i\kappa_{\mathrm{I}} = \overline{\kappa} e^{i\gamma_2}. \tag{35}$$

Thus we arrive at

$$\frac{8}{3}\rho R^3 \ddot{x} + \overline{h} e^{i\gamma_1}\dot{x} + \overline{\kappa} e^{i\gamma_2} x = T. \tag{36}$$

By introducing the natural frequency parameter $\omega_0$ as

$$\omega_0 = \sqrt{\frac{3\overline{\kappa}}{8\rho R^3}}, \tag{37}$$

the equation reduces to

$$\ddot{x} + 2h_0\omega_0 e^{i\gamma_1}\dot{x} + \omega_0^2 e^{i\gamma_2} x = \frac{3}{8\rho R^3} T. \tag{38}$$

Now taking the Hilbert transform of both sides of the equation results in

$$\ddot{\mathcal{X}} + 2h_0\omega_0 e^{i\gamma_1}\dot{\mathcal{X}} + \omega_0^2 e^{i\gamma_2}\mathcal{X} = \frac{3\mathcal{T}}{8\rho R^3}, \tag{39}$$

[5] The inertia is not used in this derivation and the apparent mass terms are actually cancelled out later. However, it is important to note that the apparent mass analogy can be made for a rotor. If that were not true, then the first term would also be complex and this derivation would be invalidated.

where we replace the thrust and movement with their analytical signal counterparts

$$\mathcal{X} = X + i\widetilde{X} = x_{\text{amp}} e^{i\omega t}, \tag{40}$$

$$\mathcal{T} = T_u + i\widetilde{T}_u = A_t(t) e^{\phi(t)}, \tag{41}$$

and equate the imaginary and real components

$$-\omega^2 - 2h_0\omega\sin\gamma_1 + \omega_0^2\cos\gamma_2 + i(2h_0\omega\cos\gamma_1 + \omega_0^2\sin\gamma_2) \tag{42}$$
$$= \frac{3}{8\rho R^3 x_{\text{Amp}}}(T_u + i\widetilde{T}_u)e^{-i\omega t}.$$

Again, the imaginary components correspond to the damping of the system.

$$2h_0\omega\cos\gamma_1 + \omega_0^2\sin\gamma_2 \tag{43}$$

$$= \frac{3}{8\rho R^3 x_{\text{Amp}}}(\widetilde{T}_u\cos\omega t - T_u\sin\omega t) \tag{44}$$

$$= \frac{3A_T(t)}{8\rho R^3 x_{\text{Amp}}}\sin\overbrace{(\phi(t) - \omega t)}^{\psi} \tag{45}$$

The left-hand term of this equation correlates with the damping of the system normalized by the apparent mass of the air using a combination of the equations set down by Carta and Niebanck (1969) and the normalization highlighted by Bowles et al. (2014) and Corke and Thomas (2015).

$$\xi = \frac{A_T(t)}{x_{\text{Amp}}}\sin\overbrace{(\phi(t) - \omega t)}^{\psi} \tag{46}$$

Finally, we can normalize

$$\Xi(t) = \frac{\xi}{P_{\text{dyn}}A} = -\frac{A_{C_t}(t)}{x_{\text{amp}}}\sin\psi(t), \tag{47}$$

where $\widetilde{C}_t(t)$ is given by the Hilbert-transformed thrust coefficient time series $C_t(t)$

$$\widetilde{C}_t(t) = \mathcal{H}[C_t] = -\frac{1}{\pi}\mathcal{P}\int_{-\text{inf}}^{\text{inf}} \frac{C_t(\tau)}{\tau - t}\mathrm{d}t, \tag{48}$$

thus giving the analytical signal magnitude

$$A_{C_t} = \sqrt{C_t^2 + \widetilde{C}_t^2} \tag{49}$$

and phase

$$\phi(t) = \arg(Y(t)) = \arg(C_t + \widetilde{C}_t), \tag{50}$$

which gives us the phase difference between the lift and the fore–aft motion,

$$\psi(t) = \phi(t) - \omega t, \tag{51}$$

from the assumed motion

$$X(t) = X_0 e^{i\omega t}. \tag{52}$$

The time-averaged damping then gives us the cycle damping:

$$\Xi_{\text{avg}} = -\frac{1}{T}\int_0^T \Xi(t)\mathrm{d}t. \tag{53}$$

As previously undertaken by Bowles et al. (2014), Corke and Thomas (2015) and Lennie et al. (2016), the cycle-averaged damping formulation provided by Carta and Niebanck (1969) can be used as a comparison. For this derivation, the comparison will be against the formula derived earlier for the fore–aft motion of a rotor.

$$\Xi = \frac{\mathrm{d}\frac{C_{\text{WT}}}{\pi X_0}}{\mathrm{d}x} = -\frac{1}{\pi X_0^2}\oint C_{\text{TTOTAL}}\mathrm{d}x \tag{54}$$

Agreement between the two calculations provides a useful (although not completely "leak-proof") verification that the analytical signal is well conditioned and that no implementation errors are present. Verifications undertaken in Lennie et al. (2016) for the original formulation showed less than < 1 % variation between the methods; Bowles et al. (2014) also remarked on the good agreement.

On the practical side, Hilbert transforms are intended to analyse narrowband signals. It was previously established in Lennie et al. (2016) that numerical or experimental noise does not cause problems for this formulation; therefore no signal filtering will be applied. Both original and phase-averaged data were analysed, although only graphs of the phase-averaged data are presented in the paper. In the original time series, stochastic variations due to turbulent inflow are present in cases with turbulence. Otherwise after a few cycles the results converge to the phase mean.

## 5 Demonstration case 1: collective blade pitch

Having presented the analysis methods, it is possible to use these methods to investigate an example case of floating platform wind turbine aeroelasticity. A case was selected that should demonstrate more complicated thrust damping behaviour. The case chosen is harmonic collective pitch in the presence of platform translation. Further potential test cases for future work would include harmonic platform movement in combination with the following:

- yawed inflow;
- inflow turbulence;
- gusts or sudden changes in direction;

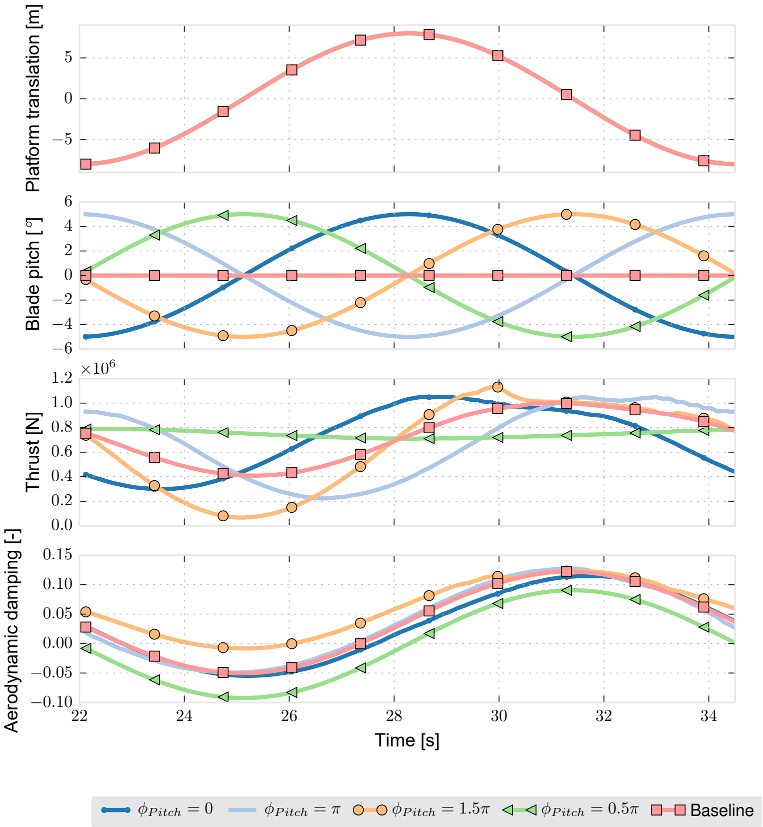


**Figure 7.** Collective pitch damping cycles.

- changes in airfoil performance through simulated active flow control;
- and/or non-synchronous pitch and platform movements.

Harmonic collective pitch in conjunction with platform movement is a complicated test case, but it is still simple enough to give a good demonstration of this particular tool chain. The collective pitch motion was prescribed using the formula

$$\alpha = \alpha_0 \sin(\omega t + \phi_{\text{pitch}}). \tag{55}$$

The test case settings are given in Table 2.

The collective pitch cycle chosen is not a realistic control regime, but it was chosen to give a clear demonstration of the method. The LLFVW simulation was run for 60 s with a single cycle chosen for analysis after the initial wake effects had died out. The instantaneous damping was calculated from the thrust data using the method already discussed. As a verification the two cycle-averaged values were compared and had good agreement; the values are presented in Table 1.

The cycle-averaged aerodynamic damping values do in fact show that collective pitch has an effect. While thrust forces tend to be positively damped (with this sign convention, that means good damping), we can see that the magnitude of the damping is altered. In Fig. 7, it is possible to follow the chain of logic that leads to these changes. In the thrust force sequences, it is possible to see that while there are some magnitude shifts, the more important feature is that the phase of the thrust force is shifted. This then manifests as changes to the aerodynamic damping.

**Table 2.** Demonstration case 1 simulation settings.

| | |
|---|---|
| Rotor speed (rpm) | 12.1 |
| $\alpha_0$ (°) | 0.5 |
| $\phi_{\text{pitch}}$ (–) | 0, 0.5$\pi$, $\pi$, 1.5$\pi$ |
| Inflow velocity (m s$^{-1}$) | 11.4 |
| $\omega_{\text{pitch}}$, $\omega_{\text{platform}}$ (rad s$^{-1}$) | 0.5 |

A closer inspection reveals an interesting feature: a 0.5$\pi$ (green) phase shift of the pitching sequence leads to an almost constant thrust force. This may appear to be favourable to reduce the fatigue loads of the wind turbine. However, what has effectively happened is that there is no force in phase with the velocity of the movement and therefore no

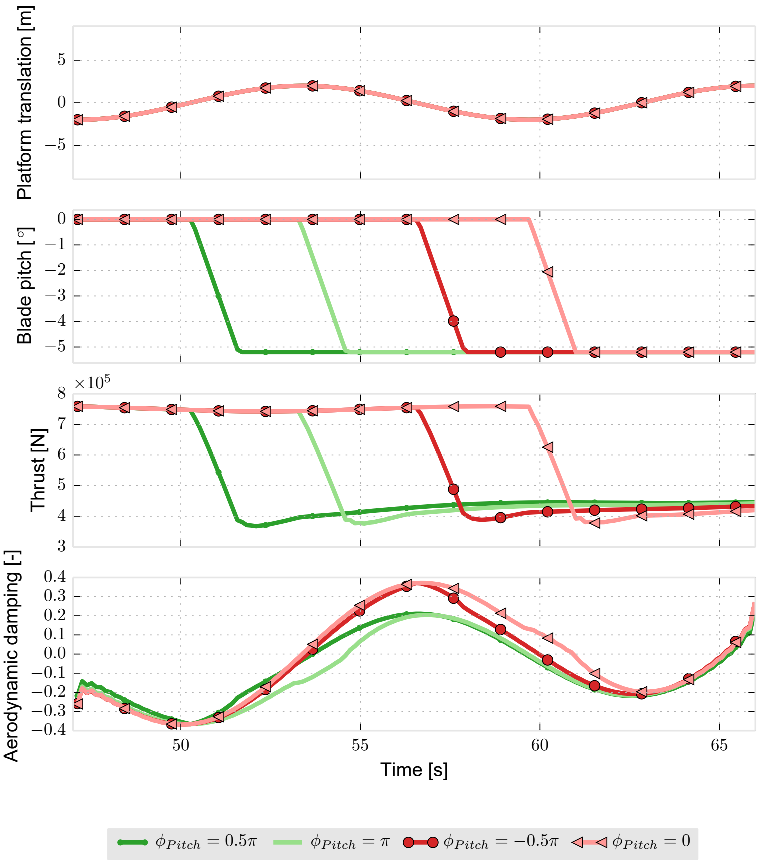


**Figure 8.** Negative blade pitch manoeuvre.

**Table 3.** Comparison of cycle damping values.

| | Averaged instantaneous damping | Cycle-averaged damping | Error |
|---|---|---|---|
| $\phi_{\text{Pitch}} = 0$ | 0.030 | 0.031 | 3.5 % |
| $\phi_{\text{Pitch}} = \pi$ | 0.036 | 0.038 | 3.5 % |
| $\phi_{\text{Pitch}} = 1.5\pi$ | 0.060 | 0.062 | 3.5 % |
| $\phi_{\text{Pitch}} = 0.5\pi$ | −0.001 | −0.001 | 3.5 % |
| Baseline | 0.036 | 0.037 | 3.5 % |

complex term, thus resulting in slightly negative aerodynamic damping. In this case, the system would rely on the other sources of damping[6] to reduce the amplitude of oscillation.

In the opposite case with a pitch phase shift of $1.5\pi$ (yellow), the thrust force is more in phase with the velocity, and thus the opposing movement of the rotor is enhanced. The cycle-averaged damping reflects this with a stronger damping value. The instantaneous damping value starts to show some departure from a pure harmonic signal. This can be traced to the matching non-linearity in the thrust force, which could arise from rotor wake interactions; it is these effects that are difficult to account for in a cycle-averaged value. In the literature examples in which a pitching airfoil was examined (Bowles et al., 2014; Corke and Thomas, 2015; Lennie et al., 2016), the non-linearities were very strong due to dynamic stall and caused strong spikes in aerodynamic damping. In simulations in which sudden changes of operating conditions are present, the instantaneous damping method will highlight sudden drops in aerodynamic damping when they occur, even if they do not show up in the cycle-averaged values.

## 6 Demonstration case 2: collective blade pitch manoeuvre

The second case attempts to demonstrate one of the most useful aspects of the instantaneous damping approach. Again in this case, we will look at a collective blade pitch manoeuvre, but this time the pitching will not be periodic. This highlights

[6] i.e. structural or that provided by the floating platform.

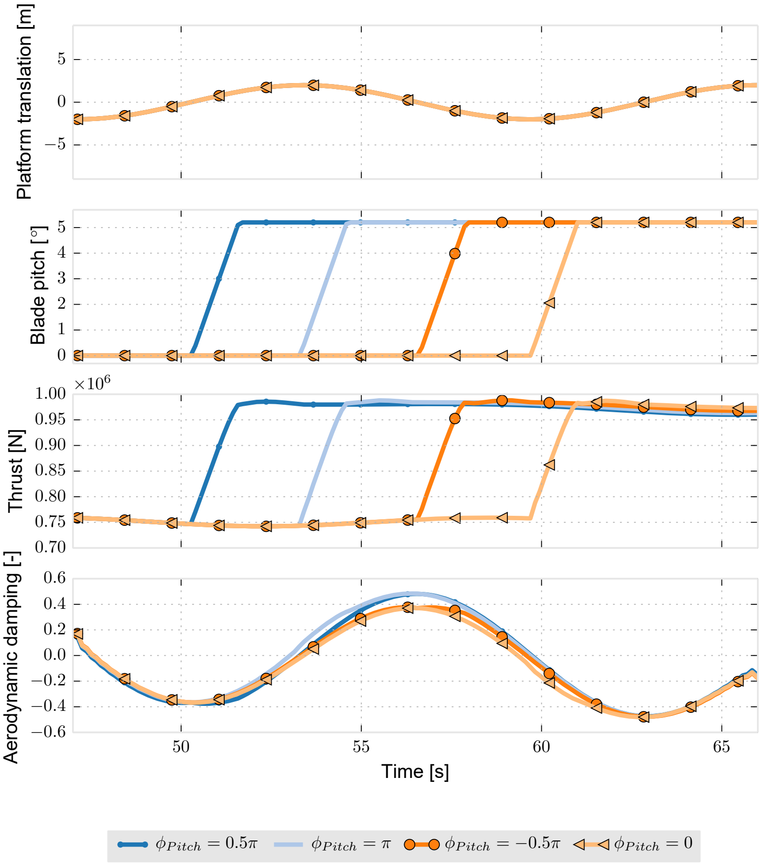


**Figure 9.** Positive blade pitch manoeuvre.

one of the advantages of the assumptions made in the derivation. The prescribed motion of the platform must be periodic, but the thrust force response has no restriction. This in turn means that we are free to try out any control manoeuvres as long as the platform motion remains periodic.

The second demonstration case will consist of a simple 5.2° blade pitch manoeuvre at a rate of $4° \mathrm{s}^{-1}$. In total, eight cases were simulated with the pitching beginning at different phase angles with respect to the harmonic translation of the platform. The platform translation had an amplitude of 2 m with a frequency of $0.5 \,\mathrm{rad\,s}^{-1}$. These simulation values are summarized in Table 4. The FVLLT simulation was run using the same inputs as listed earlier.

The results are presented in two sets with the negative blade pitch manoeuvre in Fig. 8 and the positive pitch manoeuvre in Fig. 9. For the negative blade pitch manoeuvres, we can see a clear increase in aerodynamic damping when the blades are pitched as the rotor retreats. Pitching the blades as the platform is advancing causes a clear drop in the aerodynamic damping. For the positive blade pitch manoeuvres, pitching as the rotor advances creates only a small increase in the aerodynamic damping compared to the retreating case. These kinds of results could help design a controller which restricts negative blade pitch rates as the rotor plane advances in order to maximize tower-top stability. The same approach is also useful for cases with other variations like changes in rotor speed for start up or shut down.

**Table 4.** Demonstration for case 2 simulation settings.

| | |
|---|---|
| Rotor speed (rpm) | 12.1 |
| $\alpha_0$ (°) | 0 |
| $\alpha_{\mathrm{Manoeuvre}}$ (°) | 5.2, −5.2 |
| $\omega_{\mathrm{pitch}}$ ($° \,\mathrm{s}^{-1}$) | 4 |
| $\phi_{\mathrm{pitch}}$ (–) | $0, 0.5\pi, \pi, 1.5\pi$ |
| Inflow velocity ($\mathrm{m\,s}^{-1}$) | 11.4 |
| $\omega_{\mathrm{platform}}$ ($\mathrm{rad\,s}^{-1}$) | 0.5 |
| $X_0$ (m) | 4 |

## 7 Conclusions

The QBlade implementation of the lifting-line free vortex wake (LLFVW) method proved to be a useful tool for analysing floating platform wind turbines. Comparisons against two independent test cases using a variety of methods showed relatively good agreement in thrust forces, rotor power, blade forces and rotor plane induction. Along with the many verifications already undertaken in the literature, it seems that the code will perform solidly even in these challenging cases. Further work is required to extend the same analysis with flexible blades, tower and eventually platform rather than prescribed motion; research on some of these topics is already under way.

A new formulation of the instantaneous aerodynamic thrust damping of a wind turbine rotor was described. The first demonstration case was used to verify that the cycle-averaged damping values line up with well-established methods. The case also showed how the system alternated between being stable and unstable within a single cycle. The second demonstration case showed a more complicated pitch manoeuvre; the instantaneous damping method was useful in understanding the system but provided helpful information for designing control strategies. It would be useful in future work to generalize the method so that any mode shape could be analysed without having to undertake the extensive derivation described in this paper.

**Code availability.** The full implementation of QBlade is available for download at https://sourceforge.net/projects/qblade/.

**Data availability.** No additional data were generated that have not been included in this article.

**Competing interests.** The authors declare that they have no conflict of interest.

**Special issue statement.** This article is part of the special issue "The Science of Making Torque from Wind (TORQUE) 2016". It is a result of the The Science of Making Torque from Wind (TORQUE 2016), Munich, Germany, 5–7 October 2016.

**Acknowledgements.** We acknowledge support by the German Research Foundation and the open-access publication funds of the Technische Universität Berlin.

Edited by: Carlo L. Bottasso
Reviewed by: Vasilis A. Riziotis and two anonymous referees

## References

Bisplinghoff, R. L., Ashley, H., and Halfman, R. L.: Aeroelasticity, Courier Corporation, Dover Publications, Minoela New York, 2013.

Bowles, P. O., Corke, T. C., Coleman, D. G., Thomas, F. O., and Wasikowski, M.: Improved Understanding of Aerodynamic Damping Through the Hilbert Transform, AIAA J., 52, 2384–2394, https://doi.org/10.2514/1.J052630, 2014.

Carta, F. O. and Niebanck, C. F.: Prediction of rotor instability at high forward speeds, Tech. rep., USAAVLABS TR, Clearinghouse for Federal Scientific and Technical Information, Springfield, Va, 1969.

Corke, T. and Thomas, F.: Dynamic Stall in Pitching Airfoils: Aerodynamic Damping and Compressibility Effects, Annu. Rev. Fluid Mech., 47, 479–505, https://doi.org/10.1146/annurev-fluid-010814-013632, 2015.

Eisele, O., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O.: GT2013-94689 The performance of wind turbine airfoils, in: Proceedings of ASME Turbo Expo 2013, 1–10, San Antonio, 2013.

Hansen, M. O. L.: Aerodynamics of Wind Turbines, 2nd Edn., Earthscan LLC, London Sterling, VA, 2008.

Jonkman, J. M.: NREL Offshore Baseline 5MW_Onshore with Compiled Controller, available at: http://wind.nrel.gov/public/jjonkman/NRELOffshrBsline5MW/ (last access: 1 April 2015), 2013.

Jonkman, J. M.: Fast Users Guide, available at: https://nwtc.nrel.gov/FAST7, last access: 1 April 2015.

Lennie, M., Pechlivanoglou, G., Marten, D., Nayeri, C. N., and Paschereit, C. O.: GT2015-43249: A review of wind turbine polar data and it's effect on fatigue loads simulation accuracy, Proceedings of ASME Turbo Expo 2015: Turbine Technical Conference and Exposition GT2015, https://doi.org/10.1115/GT2015-43249, 2015.

Lennie, M., Bach, A., Pechlivanoglou, G., Nayeri, C., and Paschereit, C. O.: The Unsteady Aerodynamic Response of an Airfoil with Microtabs and it's Implications for Aerodynamic Damping, 34th Wind Energy Symposium, 1–12, https://doi.org/10.2514/6.2016-1006, 2016.

Marten, D., Lennie, M., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O.: Implementation, Optimization, and Validation of a Nonlinear Lifting Line-Free Vortex Wake Module Within the Wind Turbine Simulation Code QBlade, Journal of Engineering for Gas Turbines and Power, 138, 072601, https://doi.org/10.1115/1.4031872, 2015.

Müller-Vahl, H. F.: Wind Turbine Blade Dynamic Stall and its Control, Doctoral thesis, Technische Universität Berlin, 2015.

Pitt, D. M. and Peters, D. A.: Rotor Dynamic Inflow Derivatives and Time Constants from Various Inflow Models, in: 9th European Rotorcraft Forum, 1–24, US Army Troop Support and Aviation Materiel Readiness Command, St. Louis Missouri, 1983.

Rumsey, C. L.: Turbulence Model Verification and Validation, in: ASME Verification and Validation Symposium, NASA Langley Research Center, Las Vegas, 2016.

Saverin, J., Marten, D., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O.: GT2016-56290 Coupling of an unsteady lifting line free vortex wake code to the aeroelastic HAWT simulation suite FAST, in: Proceedings of ASME Turbo Expo 2016, Seoul, South Korea, 2016.

Scanlan, R. H. and Rosenbaum, R.: Introduction to the study of aircraft vibration and flutter, Dover Publications, New York, 1951.

Sebastian, T. and Lackner, M. A.: Characterization of the unsteady aerodynamics of offshore floating wind turbines, 16, 339–352, https://doi.org/10.1002/we.545, 2013.

Strangfeld, C.: Active control of trailing vortices by means of long- and short-wavelength actuation, Doctor thesis, Technical University of Berlin, 2015.

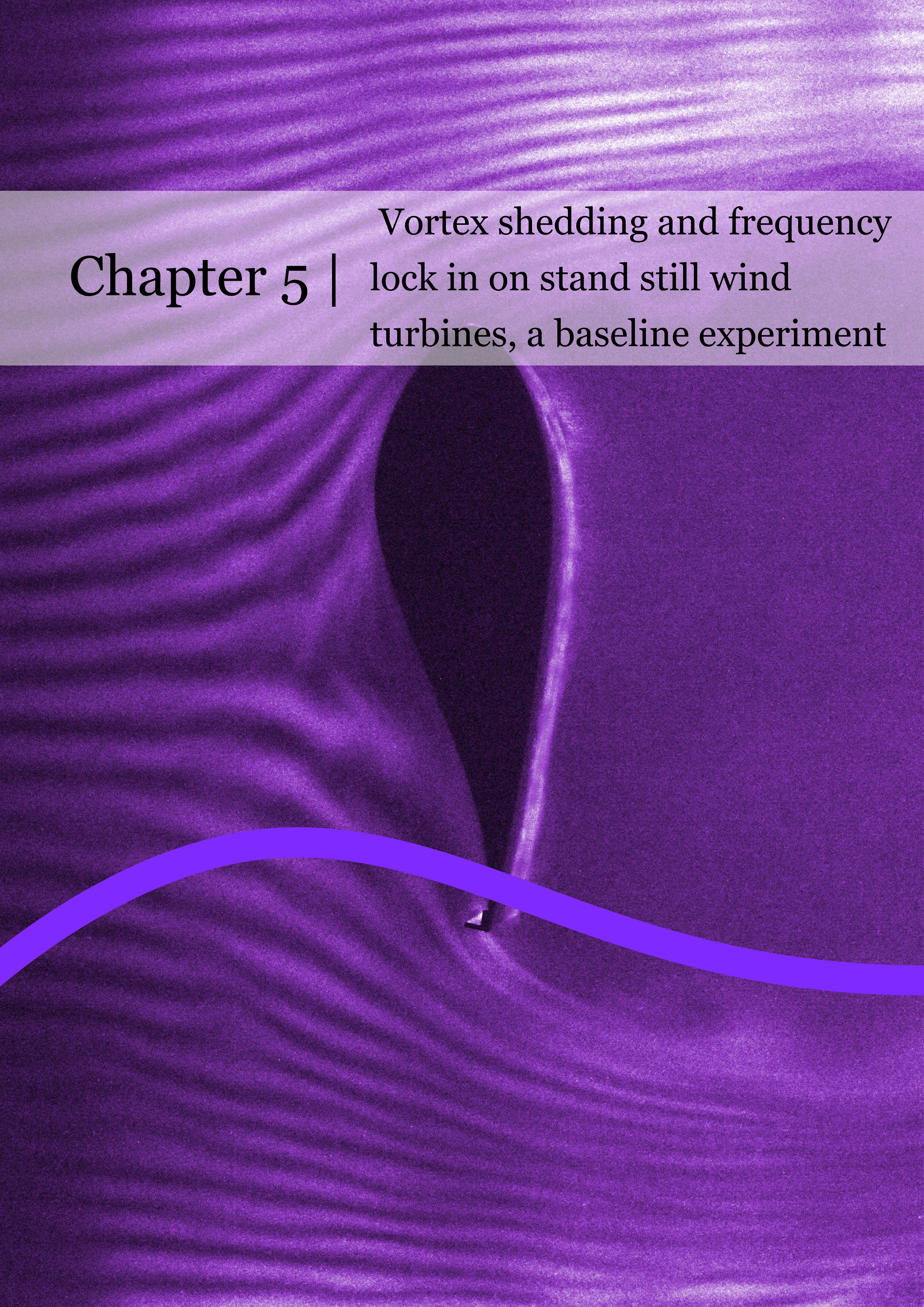
Stangfeld, C., Rumsey, C. L., Mueller-Vahl, H., Greenblatt, D., Nayeri, C., and Paschereit, C. O.: Unsteady Thick Airfoil Aerodynamics: Experiments, Computation, and Theory, 45th AIAA Fluid Dynamics Conference, 1–19, https://doi.org/10.2514/6.2015-3071, 2015.

Theodorsen, T.: General theory of aerodynamic instability and the mechanism of flutter, available at: https://ntrs.nasa.gov/search.jsp?R=19930090935 (last access: 10 December 2017), 1935.

Tran, T., Kim, D., and Song, J.: Computational Fluid Dynamic Analysis of a Floating Offshore Wind Turbine Experiencing Platform Pitching Motion, Energies, 7, 5011–5026, https://doi.org/10.3390/en7085011, 2014.

Vaal, J. B. D., Hansen, M. O. L., and Moan, T.: Effect of wind turbine surge motion on rotor thrust, 17, 105–121, https://doi.org/10.1002/we.1562, 2014.

Wendler, J., Marten, D., Nayeri, C. N., Pechlivanoglou, G., and Paschereit, C. O.: Gt2016-57184 Implementation and Validation of an Unsteady Aerodynamics Model, in: Proceedings of ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition, 1–10, 2016.



# Chapter 5 | Vortex shedding and frequency lock in on stand still wind turbines, a baseline experiment

[18]

[18] Republished with permission of Journal of Engineering for Gas Turbines and Power, from **Lennie, M.**, Selahimoghaddam, A., Holst, D., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2018). GTP-17-1616 Vortex shedding and frequency lock in on stand still wind turbines, a baseline experiment, 140(11), 112603–112603. https://doi.org/10.1115/1.4039818; permission conveyed through Copyright Clearance Center, Inc.

**Matthew Lennie**[1]
Chair of Fluid Dynamics,
Hermann-Föttinger-Institut,
Technische Universität Berlin,
Müller-Breslau-Str. 8,
Berlin 10623, Germany
e-mail: matthew.lennie@tu-berlin.de

**Alireza Selahi-Moghaddam**
Chair of Fluid Dynamics,
Hermann-Föttinger-Institut,
Technische Universität Berlin,
Müller-Breslau-Str. 8,
Berlin 10623, Germany

**David Holst**
Chair of Fluid Dynamics,
Hermann-Föttinger-Institut,
Technische Universität Berlin,
Müller-Breslau-Str. 8,
Berlin 10623, Germany

**George Pechlivanoglou**
Chair of Fluid Dynamics,
Hermann-Föttinger-Institut,
Technische Universität Berlin,
Müller-Breslau-Str. 8,
Berlin 10623, Germany

**Christian Navid Nayeri**
Chair of Fluid Dynamics,
Hermann-Föttinger-Institut,
Technische Universität Berlin,
Müller-Breslau-Str. 8,
Berlin 10623, Germany

**Christian Oliver Paschereit**
Chair of Fluid Dynamics,
Hermann-Föttinger-Institut,
Technische Universität Berlin,
Müller-Breslau-Str. 8,
Berlin 10623, Germany

# Vortex Shedding and Frequency Lock in on Stand Still Wind Turbines—A Baseline Experiment

*During the commissioning and stand-still cycles of wind turbines, the rotor is often stopped or even locked leaving the rotor blades at a standstill. When the blades are at a standstill, angles of attack on the blades can be very high, and it is therefore possible that they experience vortex-induced vibrations. This experiment and analysis helps to explain the different regimes of flow at very high angles of attack, particularly on moderately twisted and tapered blades. A single blade was tested at two different flow velocities at a range of angles of attack with flow tuft visualization and hotwire measurements of the wake. Hotwire wake measurements were able to show the gradual inception and ending of certain flow regimes. The power spectral densities of these measurements were normalized in terms of Strouhal number based on the projected chord to show that certain wake features have a relatively constant Strouhal number. The shedding frequency appears then to be relatively independent of chord taper and twist. Vortex generators (VGs) were tested but were found to have little influence in this case. Gurney flaps were found to modify the wake geometry, stall onset angles, and in some cases the shedding frequency.* [DOI: 10.1115/1.4039818]

## Introduction

During the commissioning and stand-still cycles of wind turbines, the rotor is often stopped or even locked leaving the rotor blades at a standstill. Rotors can also be set to idle outside of the normal wind operation bounds or during grid failure. Furthermore, before blades are usually attached to the hub during assembly by crane. In these configurations, the inflow angle on the rotor blades can reach very high angles of attack. During informal discussions, a number of manufacturers have revealed that large vibrations have been observed and literature also points towards such problems. Such vibrations are a problem as they contribute to the fatigue life of the turbine, even when there is no power being produced. This paper will explore some of the aerodynamic drivers of problems that can appear during such load cases.

At high angles of attack, we leave the realm of attached flow and enter the region of stall. Stall is an inherently unstable process in both space and time. In light stall, we can observe unsteady pressure oscillations as the separation point moves along the chord and even span. If the angle of attack continues to dynamically increase into deep stall, the shear layer wraps up into a leading edge vortex [1], which eventually washes downstream to create lift spikes and moment dumps [2]. The wake structure here is coherent and this form of shedding is called the "wake mode" in literature [3,4].[2] If airfoil is exposed to high angles of attack statically, the shear layer rolls up into large-scale structures as part of a continuous process; these structures grow spatially as they travel downstream [5]. This mode is referred to the "shear layer" mode. At very high angles of attack, bluff body type shedding can be observed with vorticity shedding from both the leading and trailing edges. In terms of spatial variations, Manolesos et al. observed that "stall" cells would develop and then jump to otherwise spanwise locations without any observable cause [6]. Water tunnel experimental results are particularly effective in demonstrating just how unstable the lift can be due this kind of behavior [7]. Add

[1] Corresponding author.

Contributed by the Turbomachinery Committee of ASME for publication in the Journal of Engineering for Gas Turbines and Power. Manuscript received November 14, 2017; final manuscript received March 1, 2018; published online July 30, 2018. Editor: David Wisler.

[2] See "The Dynamics of Static Stall" [5] for a detailed discussion on the differences between static and dynamic stall wake behavior.

these effects to a flexible wind turbine blade and large vibrations can occur.

Besam et al. [8–11] have been exploring the concept of vortex induced vibrations and particularly the effect of frequency lock-in. The frequency lock-in effect sees the physical translation or pitch of an airfoil (or cylinder) drive the vortex shedding at high angles of attack with in a bandwidth of frequencies centered about the natural shedding frequency. A critical part of Besem's work was to establish through experiments and simulations the relationship between the magnitude of the excitation and the bandwidth of frequency control. With modern wind turbine blades being flexible, it is entirely possible that the rotor blades are not only the victim of pure aerodynamics but also part of an aeroelastic closed feedback loop.

A key question is: is it practical to manipulate this behavior in order to avoid these kinds of frequency lock-in effects? For the structure, it may be possible to change the modal frequencies or shapes to make the driving aerodynamic force and response incoherent. Unfortunately, making significant changes to mode shapes and frequencies of blades requires massive amounts of material and may cause the blades to be sensitive to other driving frequencies in its normal operation. Thus, changing the structure is probably undesirable but it may be possible to change the aerodynamic driving frequencies. This could be achieved through the use of passive flow control devices. The upside of this approach would be that many of these devices are cheap to manufacture and install.

The first device investigated in this work is the vortex generator (VG's). Vortex generators are small devices installed usually to the suction side of an airfoil to delay stall. VGs delay stall by creating longitudinal vortices, which transfer high momentum fluid into the boundary layer; this extra momentum helps overcome the adverse pressure gradient found on the suction side of airfoils. This effect tends to delay stall until higher angles of attack resorting in a high maximum lift at a higher angle of attack [12,13]. These improvements are accompanied with an increase in parasitic drag, which is most noticeable at low angles of attack. Of course, the VGs have to be exposed to high momentum fluid in order to function. This means that once the VGs are either swamped by the boundary layer or separated flow, they are no longer effective; the lift behavior will then merge again with the behavior of the original airfoil. Wind turbine airfoils are usually relatively thick with well-rounded leading edge geometry; we can expect stall to develop gradually from the trailing edge as the angle of attack increases [7,14]. An experimental parameter study from Mueller-Vahl et al. [12] demonstrates that this kind of trailing edge separation can be delayed. In practice, vortex generators have resulted in an increase in annual energy production with either neutral or slightly increased loading [15,16]. The results are particularly positive when leading edge roughness is present [17], which is commonly the case on wind turbines [18].

The question that remains, however, is: will the VGs have any effect on the kinds of lock on effects which are the main subject of this study? There is ample evidence that the leading edge type of shedding (wake mode) can be attenuated or removed completely through the use of periodic excitations. A zero-net mass flux actuator, for instance, forces shear layer roll up at a higher frequency than the natural frequency (thus have a smaller scale), which results in smaller vortices, which encourage mixing much like the vortex generators [19,20]. Various sources have investigated the use of VGs (and dynamic VGs) for dynamic stall control [21–24]. These studies invariably indicate that dynamic stall vortices can be attenuated using active or passive VGs. Experimental PIV studies seem to indicate that like the zero-net mass flux actuators, by breaking up the large structures into smaller structures, the flow seems to have a better resistance to the adverse pressure gradients [23]. Greenblatt and Wygnanski [19,20] indicated that the addition of control did introduce a small perturbation in the lift, drag, and moment coefficients but seemed to suppress the large inter-cycle oscillations usually present post stall [25]. Mulleners and coworkers [23], however, seemed to indicate that although the VGs did smooth the stall behavior, in post stall, high amplitude pressure perturbations where present, which were attributed to the shifting position of the separation point.[3] Such unsteadiness would be detrimental rather than beneficial. It seems that VGs can exert control on wake mode types of stall in a mostly beneficial manner. This study will try to establish whether VGs show an effect in shear layer mode shedding as well, that is, before the VGs at completely engulfed by the separated flow.

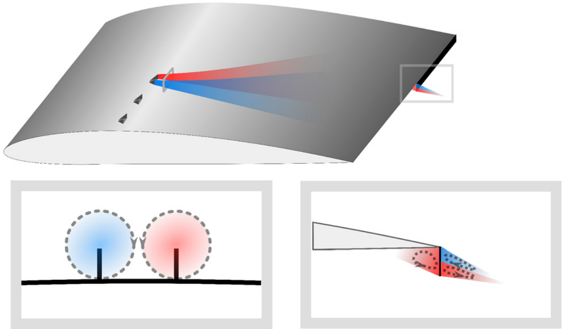


**Fig. 1 Vortex patterns from VGs and a gurney flap on the pressure side of an airfoil**

Vortex generators function by introducing a stream wise vortex into the flow, which acts to transfer momentum into the boundary layer and chops large flow structures into smaller structures. Another way to influence shear layer mode shedding to introduce vorticity with a spanwise core into the flow. We can achieve this passively by adding gurney flaps to an airfoil. Liebeck [26] was one of the first authors to consider the vortex system that would form around a gurney flap (see Fig. 1). This is a static representation of the vorticity; in real circumstances, we can expect a Von Karman street shedding system to form as the shear layer roll up (thus vortex formation) alternates between the upper and lower shear layers [27,28]. The shedding frequencies are usually much higher than the airfoil shedding frequency and may have a particularly strong effect on the trailing edge shear layer. In a steady sense, gurney flaps operate by disrupting the Kutta condition. In place of the normal Kutta condition, vorticity forms upstream and downstream of the gurney flap on both the suction side and pressure side as shown in Fig. 1. The shedding downstream of a Gurney flap results (on average) in a shift toward the direction of gurney flap. Nevertheless, the gurney flap inherits a periodic wake consisting of a pronounced downwash and a lighter upwash, as shown by Holst et al. [29] for finite width gurney flap.

Previous two-dimensional (2D) testing has shown that gurney flaps on the suction side of an airfoil will become ineffective at modifying the time-averaged coefficients at high angles of attack [30]. This is true for static and dynamic pitch tests. While the Kutta condition does not hold in separated flows, gurney flaps should nonetheless modify the wake geometry. Instead of steering the entire wake, we can expect the trailing edge shear layer to be steered instead. Through computational fluid dynamics (CFD), Bach [28] was able to demonstrate qualitatively that suction-side microtabs will modify the size and strength of trailing edge and leading edge vorticity even though the airfoil was already in deep stall at (18 deg). A simple flow visualization on an S809 airfoil was enough to demonstrate that even at very high angles of attack, gurney flaps will change the geometry of the wake which should have an effect on the unsteady lift (see Fig. 2). The flow visualizations show that the gurney flap does steer the shear layer downward thus increasing the separated zone, analogous to an increase in chord length. It was not possible to observe unsteady effects hypothesized above with the smoke visualization but they may

[3] These tests were conducted on an OA209 a thinner airfoil with harder stall characteristics than most modern wind turbine airfoils.

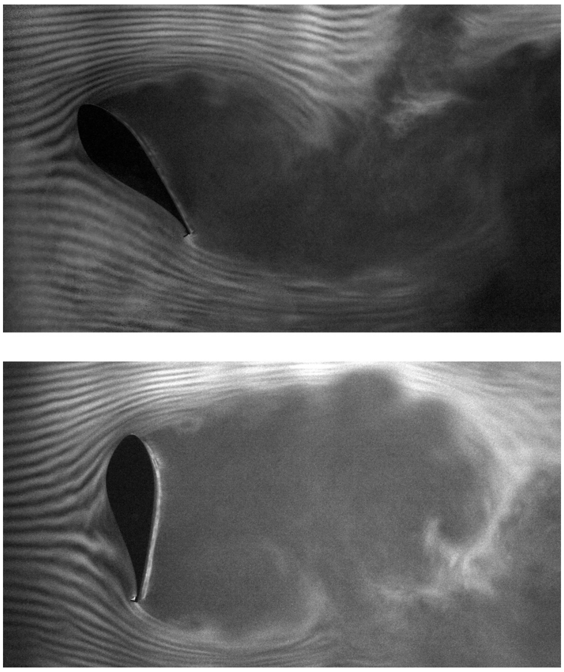


Fig. 2 Smoke visualization of an S809 with 5 mm gurney flap at high angles of attack (low Reynolds) in the visualization wind tunnel. Notice the steering of the trailing edge shear layer and also the interaction between the shear layers downstream.

well still be present. Between the mean effects of modified wake geometry and the dynamic effects of introducing high frequency vorticity at the shear layer, some effect on the shedding behavior of the airfoil can be expected. This study will try to establish whether this hypothesis is valid by also testing gurney flaps installed on the pressure side of the airfoil.

The simulations and experiments provided by literature so far cover prismatic airfoil sections with constant mechanical properties. Essentially, the test configurations were distilled to only consider 2D effects. Wind turbine blades have none of these features; they are twisted, have variable chords, and the airfoils are gradually changed over the span. Manolesos et al. [6] highlighted that even for a prismatic section, stall should be considered as a three-dimensional (3D) effect. But consider now that natural vortex shedding frequencies are partly driven by chord length and pitch. Over the blade, we now have a range of natural shedding frequencies and mode shapes (with associated eigenfrequency). So, there are still some steps required before understanding vortex-induced vibrations on a wind turbine.

Literature provides different ways of approaching instabilities at high angles of attack. Wang et al. [31] used nonlinear time domain and linear eigenvalue analysis to identify possible modes while the turbine is idling. Pirrung et al. [32] have been attempting to address the problem through reduced order methods by creating a trailed vorticity model for blade element momentum. If we look at a standard implementation of a Beddoes Leishman type [33] unsteady aerodynamics model like that provided by Wendler et al. [34], we can see that the treatment of very deep or massive stall is not specifically treated. So, if the wind turbines are being subjected to vortex-induced vibrations, we could not expect the current models to capture it accurately. Various CFD efforts using dynamic eddy simulation (DES) and Reynolds averaged Navier–Stokes have shown vortex-induced vibrations and lock-in effects for 2D profiles [35,36]. While Besam [9] used a harmonic balance method based on Reynolds averaged Navier–Stokes to resolve the shedding frequency, there is a lack of literature at this point in time extending the principle the 3D case,[4] likewise for DES or large eddy simulation solution methods. Deep dynamic stall comparisons between 2D and 3D indicate that pressure distributions, lift curves, moment curves, and vortical dynamics are better resolved by 3D simulations (of a prismatic airfoil section) [37,36]. In a recent study, Gaunaa et al. [38] used 3D DES simulations on blades during installation to build a correction factor for the static forces, thereby highlighting the influence of cross flows and other 3D effects on blades at unusually high angles of attack. It seems that there is still a lot to learn about the full 3D case and certainly a need for greater understanding.

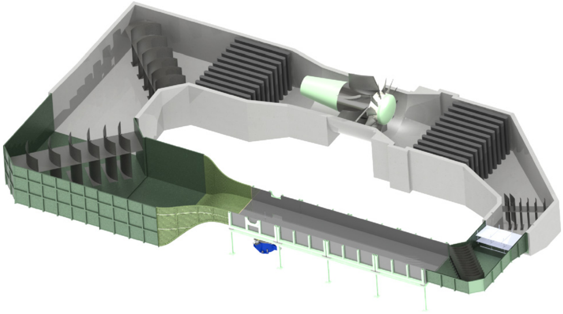


Fig. 3 GroWiKa wind tunnel

Table 1 Overview of the chord and twist of the blade for each measurement height

| Span (mm) | Chord (mm) | Twist (deg) | Span (mm) | Chord (mm) | Twist (deg) |
|---|---|---|---|---|---|
| 0 | 267 | 10 | 360 | 180 | 5.4 |
| 90 | 242 | 8.7 | 450 | 164 | 4.5 |
| 180 | 221 | 7.5 | 540 | 144 | 3.4 |
| 270 | 201 | 6.4 | 630 | 115 | 2.2 |

This paper describes an experimental campaign that will establish a baseline of a 3D blade at very high angles of attack. The experiment consists of a 3D wind turbine blade taken from the Berlin research wind turbine (BeRT) [39] being tested in the GroWiKa wind tunnel at the Technische Universität Berlin. A pair of X configuration hot wires was used to measure the flow velocity downstream of the blade. This setup is not actuated so these results will serve as a baseline study. With this setup, we can particularly demonstrate for the 3D case whether the frequency spectrum behavior found in the work of Tang and Dowell [11], where a distinct shift between broadband noise and narrowband shedding (with increasing angle of attack), can be discerned as clearly. The results will also provide an assessment of whether the 2D assumption made in the aforementioned literature can be treated as accurate enough. It is hoped that these results will provide a better understanding of vortex-induced vibrations on wind turbines.

## Experiment Setup

The BeRT wind turbine is a research turbine designed for the GroWiKa wind tunnel (Fig. 3) at the Hermann-Föttinger-Institut of Technische Universität Berlin [39]. The wind tunnel is a closed-loop wind tunnel with a turbulence level of less than 0.5% and a test section of 1.4 m × 2 m. The test subject was a single rotor blade from the BeRT Wind turbine [40]. The rotor blades were designed based on a Clark-Y airfoil with a chord length and twist distribution shown in Table 1. It should be noted that this airfoil is not a standard wind turbine airfoil but was chosen for the

[4] Read here: 3D Blade with twist and taper. The following references have analyzed a simple 2D airfoil with a 3D grid [35–37].

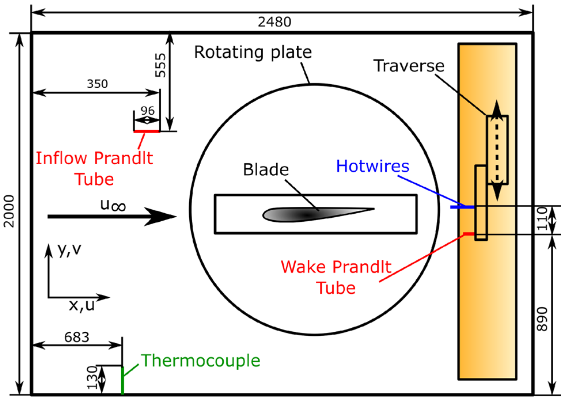


Fig. 4 Top view of the experimental setup (not to scale)

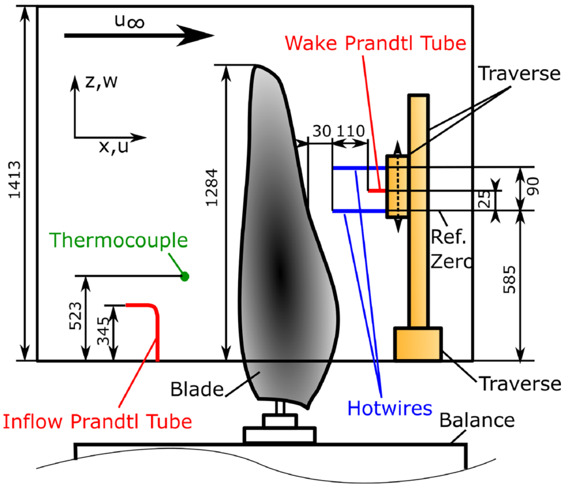


Fig. 5 Side view of the experimental setup (not to scale)

BeRT wind turbine in the context of a larger research project. This airfoil is thinner than a large modern wind turbine airfoil so some differences in the stall behavior can be expected due to the tendency of thin airfoils with sharper leading edges to stall hard [14]. Furthermore, the baseline test configuration is a clean blade, which again departs from the reality of wind turbine practice [41] where leading edge soiling causes a forced transition of the boundary layer.

A single blade was detached from the turbine and clamped vertically (using the standard hub connection) onto the rotating force balance. The blade passed through the rotating turntable plate with a small gap between the foil and the plate. Behind the blade, a two-axis traverse with two X configuration hot-wire probes was constructed. Prandtl tubes were used to measure the incoming and wake flow velocities as a backup for the velocities measured from the wind tunnel nozzle pressure difference and X wires, respectively. An overview of the setup is presented in Figs. 4 and 5.

For the vortex generator configuration, there were used two different sizes. The smaller VGs had a width of 7 mm and a height of 5 mm, the bigger VGs had a size of 15 mm and 10 mm, respectively. Ten of the smaller VGs were installed and seventeen of the large. The small VGs were fixed on the blade beginning at 70 mm beneath the blade tip and covering a range of 325 mm, the big VGs covering a range of 210 mm and beginning 25 mm beneath the lowest small VGs.

The gurney flap was constructed from a simple plastic L-profile. It was mounted along the trailing edge on the suction side of the blade beginning 65 mm beneath the tip and ranging to 988 mm beneath the tip. This covered the entire measurement range. The gurney flaps had a height of 5 mm based on previous parameter studies undertaken by Bach [28]. The configurations for the VGs and gurney flaps are shown in Fig. 6.

An initial fast sweep of the wake was undertaken to detect the edges of the wake at all angles of attack. An adaptive grid was then generated for the main tests to ensure that the regions of high gradient were well resolved with a high density of measurement points (see Fig. 7). The two hotwires were oriented perpendicularly and then each of the measurement points was repeated for each probe, thus resolving the velocities in all three directions. The hotwires were calibrated against the probe designated as the “Wake Prandtl Tube” with the traverse in a position well clear of any interference from the wake. The hotwires were used in

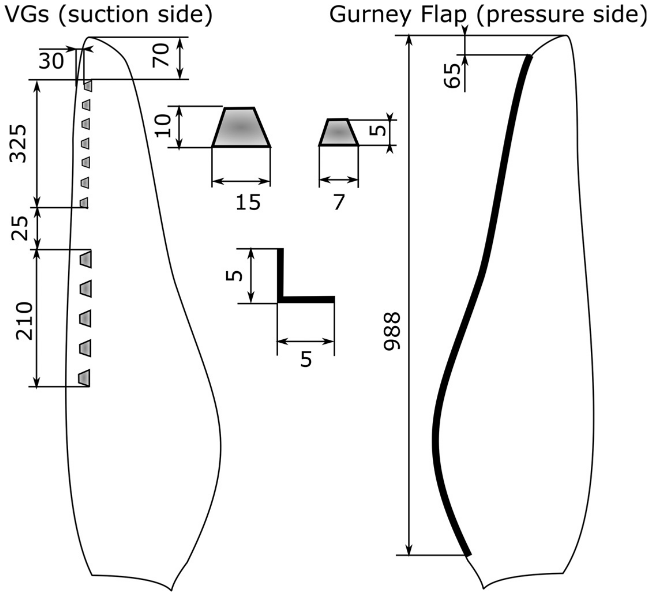


Fig. 6 The VG and GF configurations

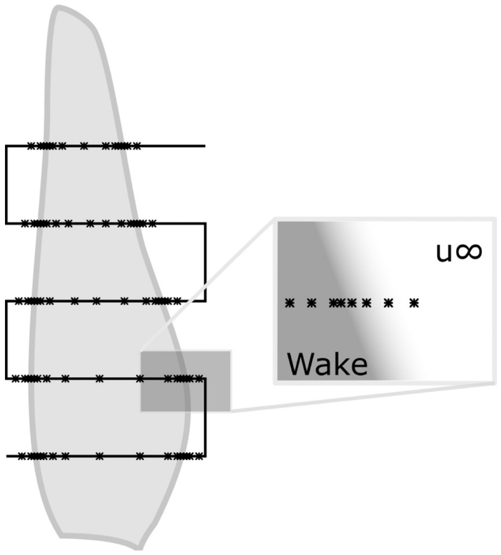


**Fig. 7 Measurement point density optimized to capture the edge of the wake**

constant temperature configuration with a 2 kHz low pass filter on the TSI IFA100 Anemometers (Table 2).

The data were sampled at a rate of 5 kHz using a NI 9188 cDAQ chassis equipped with NI-9215 modules. The signals were digitized synchronously at 16 bit and recorded using a custom LabVIEW program. Inflow speed and ambient data were obtained by additional sensors as depicted in Figs. 5 and 7.

The blade was tested in quasi-static conditions for a range of angles of attack and for two Reynolds numbers. For simple flow visualization, a case was run with flow tufts. The tufts and backing foil were then removed resulting in a clean configuration for the main baseline tests.

## Results

**Baseline Results.** For the scope of this paper, two data sets will be analyzed. The first set is the flow tuft visualization shown in Fig. 8. During the short wake measurements, photos were taken at each angle of attack. The photographs were post processed using a sequence of brightness adjustment, threshold filtering, and erosion thickening to leave only the tufts in black and white. One "dead" tuft was removed from the photograph after it was discovered posttest that it was stuck in reversed position. The outline of the blade was then superimposed back onto the diagrams giving the results shown in Fig. 8. These steps were taken to improve readability.

The flow tuft results show that for the higher angles of attack, as expected, the tufts seem to indicate separated flow. What is interesting is that different cross flow patterns can be observed over the span of the blade with a pattern change most noticeably at midspan. Video sequences showed that this midspan pattern

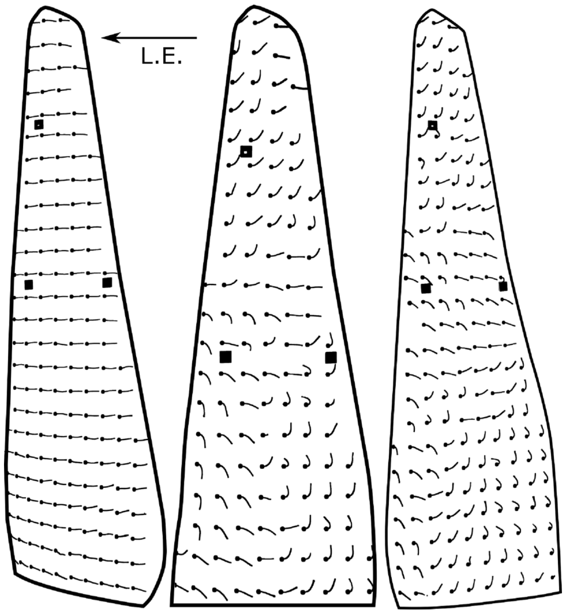


**Fig. 8 Flow tufts visualization on the suction blade surface at 6 deg, 30 deg and 70 deg from left to right**

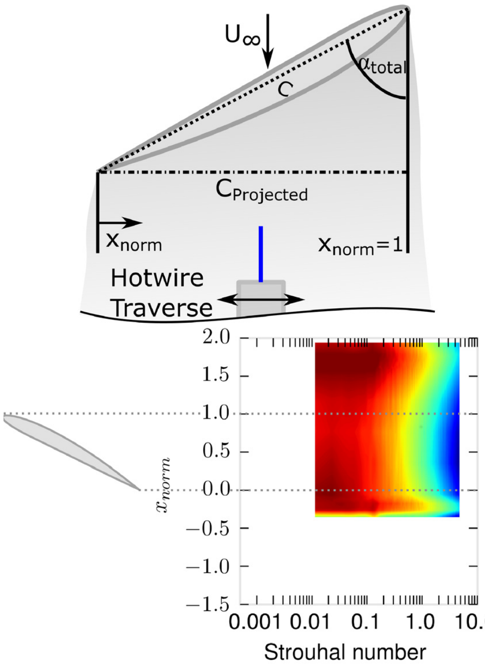


**Fig. 9 Wake coordinates normalized by projected chord**

**Table 2 Overview of the different run configurations**

| # | Surface | Velocity | Samples | AoA |
|---|---|---|---|---|
| 1 | Clean | 14 m/s | 262,144 | −6 deg, −3 deg, 0 deg, 3 deg, 6 deg, 9 deg, 12 deg, 15 deg, 20 deg, 30 deg, 40 deg, 50 deg, 60 deg, 90 deg |
| 2 | Clean | 20 m/s | 262,144 | |
| 3 | VG | 14 m/s | 262,144 | |
| 4 | Gurney | 14 m/s | 262,144 | |

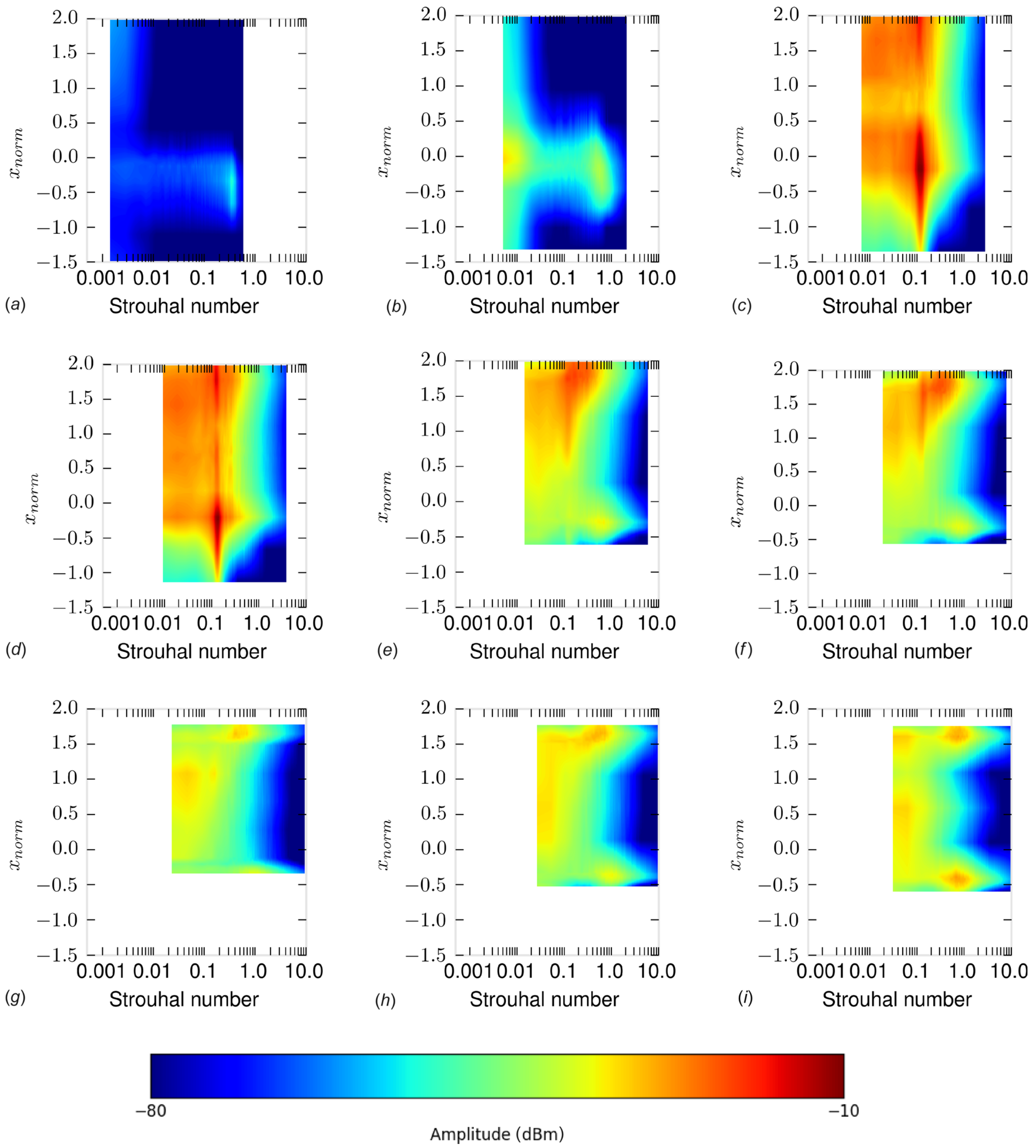


**Fig. 10 V component wake velocities at a height of 360 mm (above reference) for Angles of Attack from 0 deg to 90 deg at an inflow of 14 m/s: (*a*) $\alpha_{local}$ 2.6 deg, (*b*) $\alpha_{local}$ 9.4 deg, (*c*) $\alpha_{local}$ 12.4 deg, (*d*) $\alpha_{local}$ 17.4 deg, (*e*) $\alpha_{local}$ 27.4 deg, (*f*) $\alpha_{local}$ 37.4 deg, (*g*) $\alpha_{local}$ 47.4 deg, (*h*) $\alpha_{local}$ 57.4 deg, and (*i*) $\alpha_{local}$ 87.4 deg**

change seems to surge up and down the span of the blade. This surge pattern was not harmonic but seemed to have two distinct states. Unfortunately, surface pressure measurements were not possible but the simple visual result seems to show some kind of stall cell behavior like described by Manolesos et al. [6].

The hotwire data were post processed to yield the power spectral densities using overlapping and average methods with a Hanning Window. The signal was split into 255 segments using a window overlap of 50%. The coordinate corresponding to the width of the wake was normalized by the projected chord length for the local angle of attack taking into account the global pitch setting and local blade twist (see Fig. 9). For very low angles of attack, the normalization was based on the thickness of the section thus giving

$$c_{\text{projected}} = \max \begin{cases} t_{\max} \\ c\sin(\alpha) \end{cases} \quad (1)$$

where

$$t_{\max} = \text{maximum airfoil thickness} \quad (2)$$

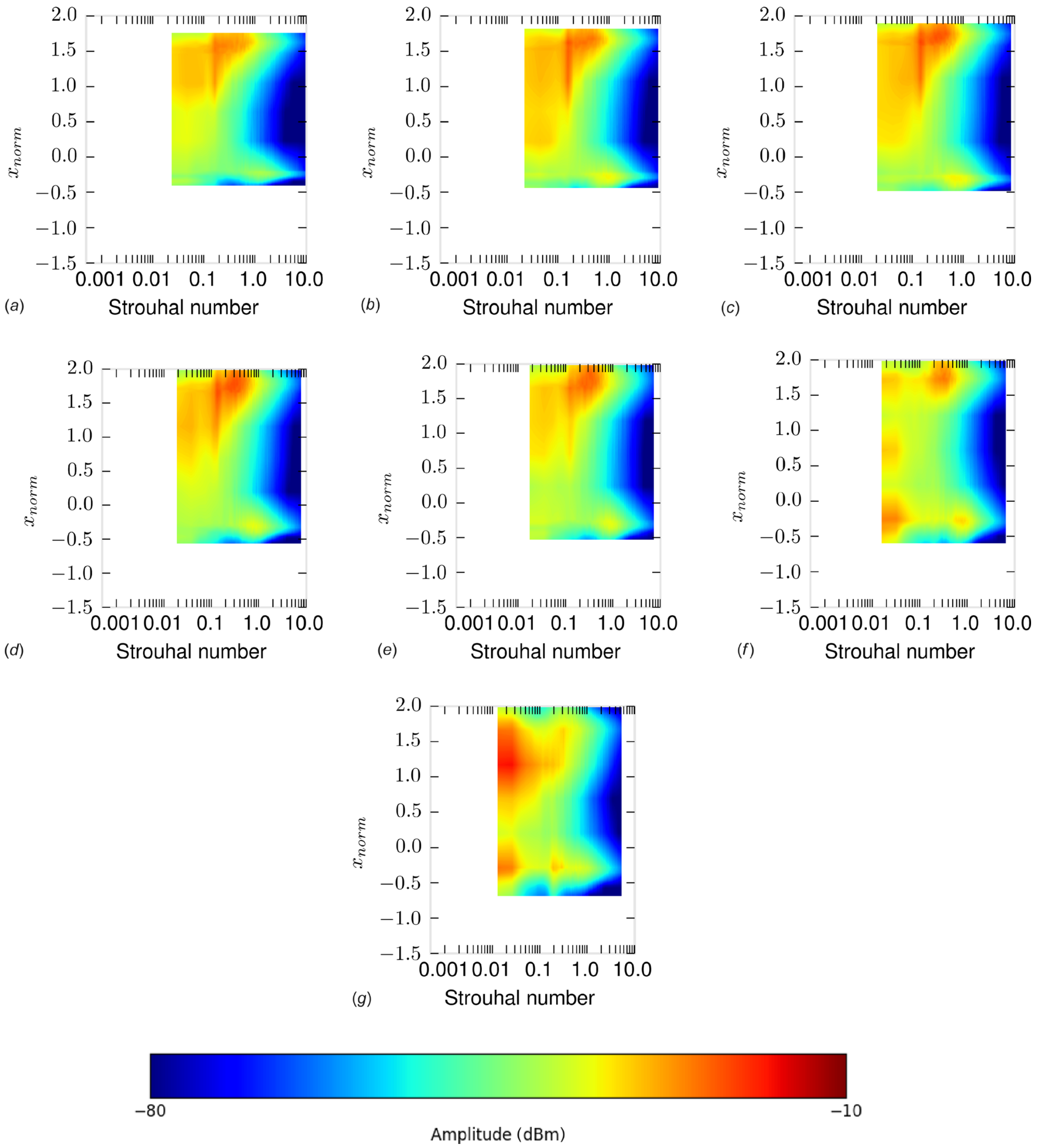


**Fig. 11 V component wake velocities across the blade span for a reference pitch 40 deg at an inflow of 14m/s: (*a*) $\alpha_{\text{local}}$ 34 deg at 90 mm from reference, (*b*) $\alpha_{\text{local}}$ 35 deg at 180 mm from reference, (*c*) $\alpha_{\text{local}}$ 36 deg at 270 mm from reference, (*d*) $\alpha_{\text{local}}$ 37 deg at 360 mm from reference, (*e*) $\alpha_{\text{local}}$ 38 deg at 450 mm from reference, (*f*) $\alpha_{\text{local}}$ 39 deg at 540 mm from reference, and (*g*) $\alpha_{\text{local}}$ 40 deg at 630 mm from reference**

and then

$$x_{\text{norm}} = \frac{x_{\text{absolute}}}{c_{\text{projected}}} - \frac{c(1 - x_{\text{pitchaxis}})}{c_{\text{projected}}} \tag{3}$$

The results were then offset to place the trailing edge at zero and the leading edge or thickness are placed at 1 (see Fig. 9 for an example graph). For ease of comparison, all graphs were given the same axis limits. The frequencies were normalized giving the results in terms of Strouhal number using the following formula based again on the projected chord length:

$$\text{St} = \frac{f c_{\text{projected}}}{U_\infty} \tag{4}$$

The hotwire measurements in Fig. 10 display four different wake regimes. (Item enumeration corresponds to the subfigure enumeration.) They are:

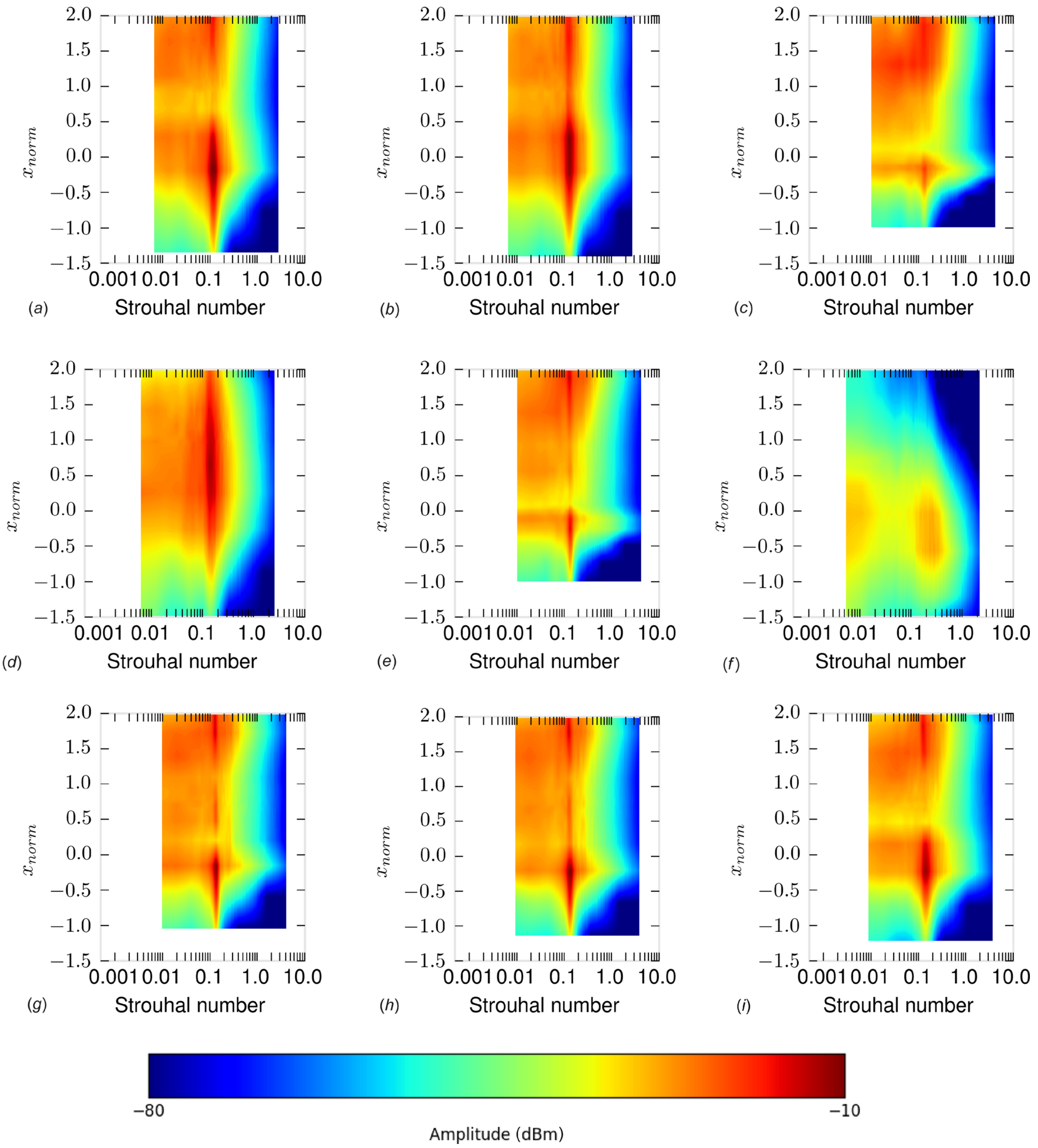


**Fig. 12 V component for angles of attack from 12 deg to 19 deg with varying spanwise positions and chord lengths at an inflow of 14m/s: (*a*) $\alpha_{local}$ 12.4 deg, spanwise position 360 mm, (*b*) $\alpha_{local}$ 13.3 deg, spanwise position 450 mm, (*c*) $\alpha_{local}$ 14 deg, spanwise position 90 mm, (*d*) $\alpha_{local}$ 14.4 deg, spanwise position 540 mm, (*e*) $\alpha_{local}$ 15.3 deg, spanwise position 180 mm, (*f*) $\alpha_{local}$ 15.6 deg, spanwise position 630 mm, (*g*) $\alpha_{local}$ 16.4 deg, spanwise position 270 mm, (*h*) $\alpha_{local}$ 17.4 deg, spanwise position 360 mm, and (*i*) $\alpha_{local}$ 18.3 deg, spanwise position 450 mm**

Figures 10(*a*)–10(*b*): Attached flow, slight presence of boundary layer noise at the trailing edge.

Figures 10(*c*) and 10(*d*): Trailing edge separation with a dominant frequency around St = 0.15. Increasing angle of attack extends the shed wake zone toward leading edge.

Figures 10(*e*) and 10(*f*): The dominant frequency content is now located toward the leading edge. The shedding frequencies are still centered about St = 0.15 but are more broadband. The structures are large extending more than 1 project chord length suggesting wake mode shedding and look the same as a dynamic stall vortex. Simultaneously, towards the trailing edge, what appears to be a trailing edge vortex.

Figures 10(*g*)–10(*i*): The trailing edge shear layer is still present and appears to be unaffected by further increases of angle of attack. The leading edge is no longer shedding the large scale structures seen in Figs. 10(*e*) and 10(*f*), the vorticity seems to be confined to the shear layer. The wake pattern at this distance from the foil looks like two shear layers with a large dead zone in the middle. The hotwires are too close to capture the kinds of shear layer interactions seen in Fig. 1. The leading edge shedding

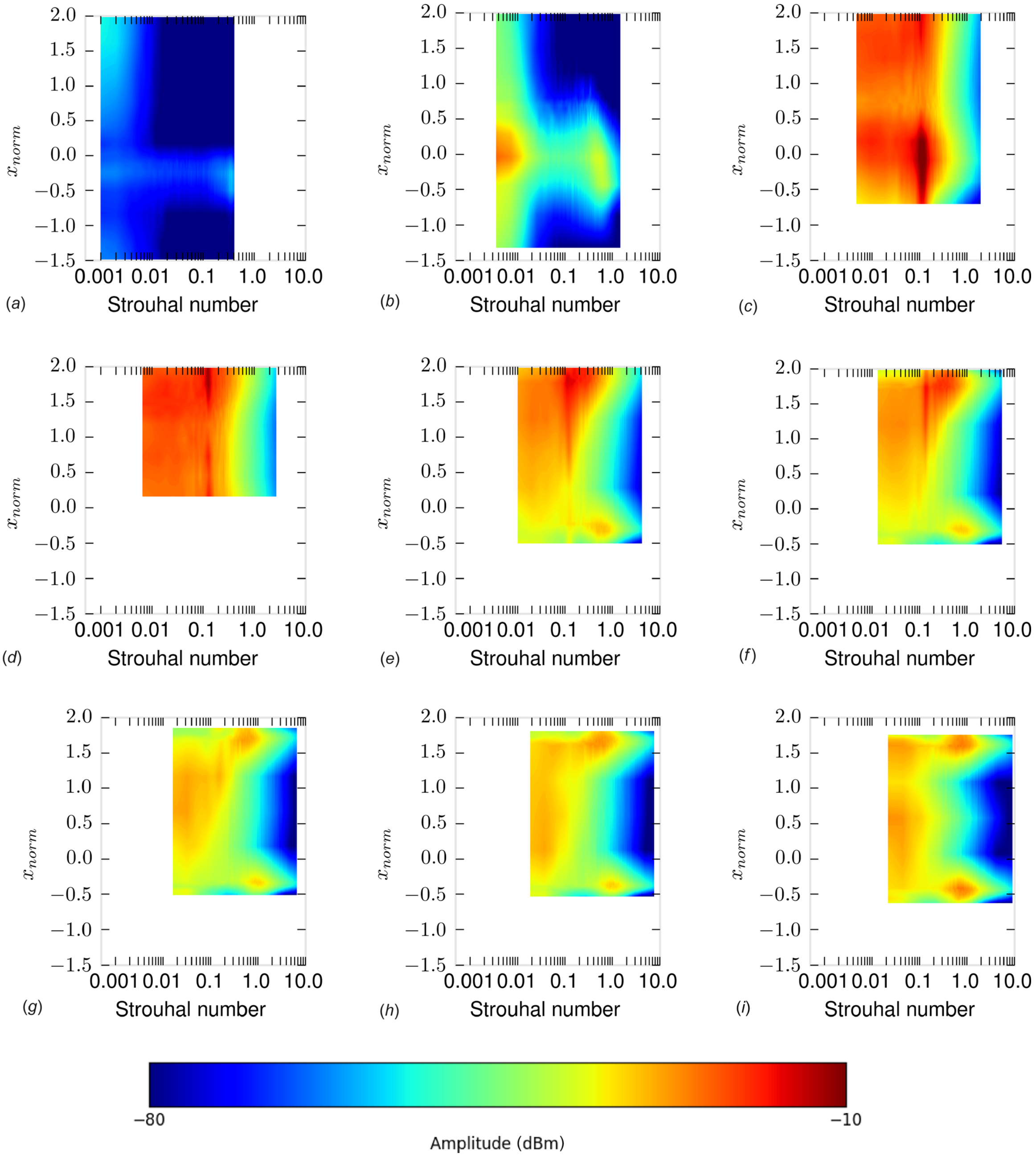


**Fig. 13 V component wake velocities at a height of 360 mm (above reference) for angles of attack from 0 deg to 90 deg at an inflow of 20m/s: (*a*) $\alpha_{local}$ 2.6 deg, (*b*) $\alpha_{local}$ 9.4 deg, (*c*) $\alpha_{local}$ 12.4 deg, (*d*) $\alpha_{local}$ 17.4 deg, (*e*) $\alpha_{local}$ 27.4 deg, (*f*) $\alpha_{local}$ 37.4 deg, (*g*) $\alpha_{local}$ 47.4 deg, (*h*) $\alpha_{local}$ 57.4 deg, and (*i*) $\alpha_{local}$ 87.4 deg**

frequencies increase with angle of attack closing the disparity between leading and trailing edge frequency bands. The leading edge has transitioned from wake mode shedding to a shear layer shedding pattern between 35 and 50 deg angle of attack.

The transition into the shear layer shedding mode occurs at roughly the same angles of attach as shown in Tang's results, which were surface pressure based [11]. Interestingly, at 35 deg, Tang's results show a clean peak at the higher frequency at a rather messy spectrum at the lower frequencies. As the angle of attack increases to 50 deg the high frequency peak is in fact a double peak, which could be attributed to the different wake frequencies at the leading and trailing edge. It appears that qualitatively, the behavior of the 3D case is very similar to the 2D case tested previously. Also, the shift in frequency content seen on the pressure sensors in Tang's results can be understood as a change of wake behavior.

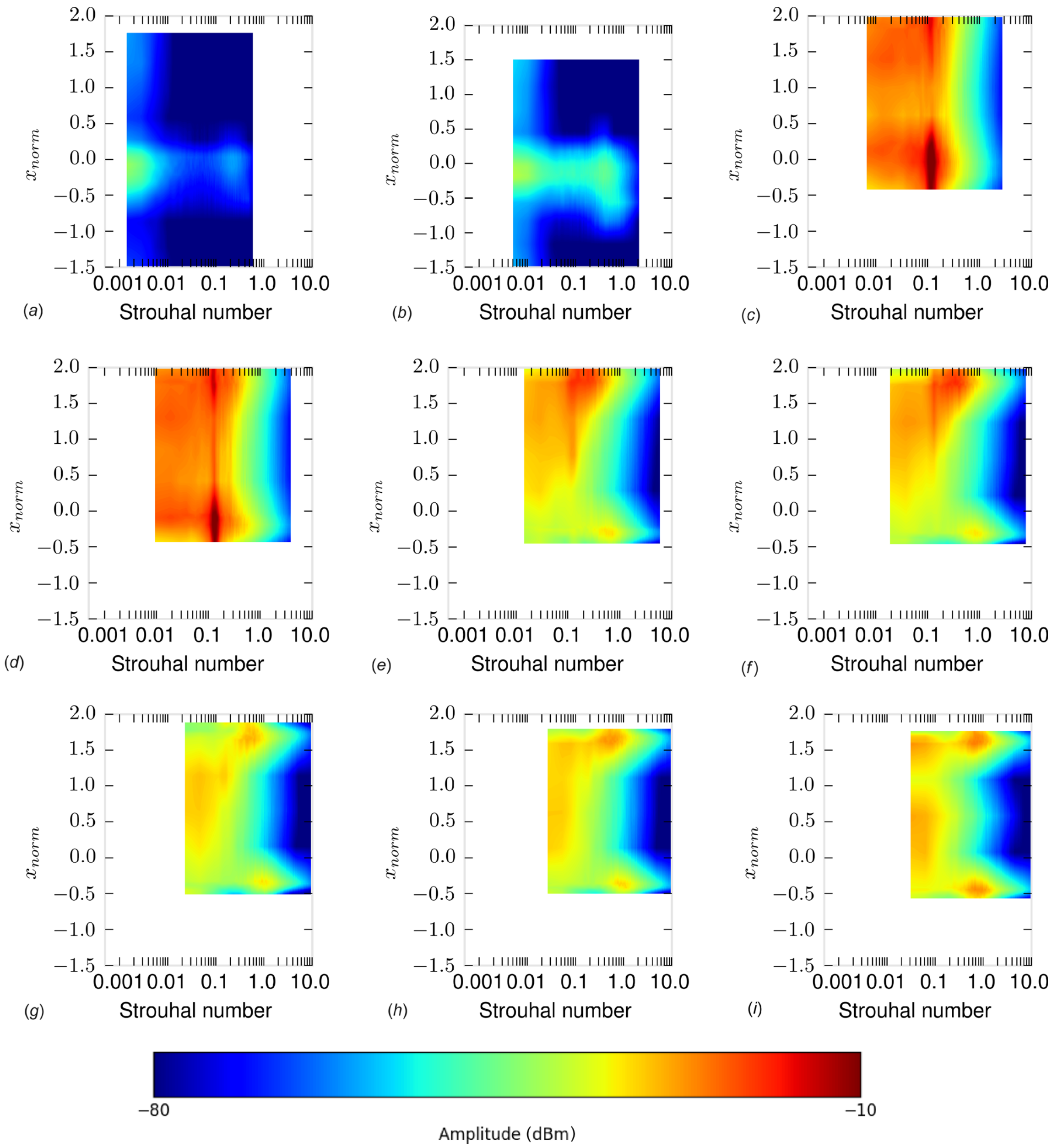


**Fig. 14 V component wake velocities at a height of 360 mm (above reference) for angles of attack from 0 deg to 90 deg at an inflow of 14m/s with VGs: (*a*) $\alpha_{\text{local}}$ 2.6 deg, (*b*) $\alpha_{\text{local}}$ 9.4 deg, (*c*) $\alpha_{\text{local}}$ 12.4 deg, (*d*) $\alpha_{\text{local}}$ 17.4 deg, (*e*) $\alpha_{\text{local}}$ 27.4 deg, (*f*) $\alpha_{\text{local}}$ 37.4 deg, (*g*) $\alpha_{\text{local}}$ 47.4 deg, (*h*) $\alpha_{\text{local}}$ 57.4 deg, and (*i*) $\alpha_{\text{local}}$ 87.4 deg**

If one could consider the wake to be spanwise independent, this would be very useful for analysis. The results discussed so far were for a single spanwise position at different pitch settings. Figure 11 shows different spanwise stations for a single pitch setting of 40 deg resulting in local angles of attack from 34 deg to 40 deg due to the twisted blade. As the angle of attack increases, trailing edge shear layer vorticity strengthens and the leading edge transitions eventually at 39 deg to shear layer type shedding. In Fig. 10, a selection of wake measurements was collected from different spanwise positions, different blade pitch angles but with similar local angles of attack (12–19 deg). From these data, it is possible to see that the spanwise location of the section is a rather small influence compared to the local angle of attack (Fig. 12). The one exception here is the sections close to the tip region where tip vorticity is present due to the sudden change in circulation. Furthermore, increasing the Reynolds number had no significant effect as can be seen by comparing Fig. 13 with an inflow of 20 m/s with Fig. 10 with an inflow of 14m/s.[5] It seems that in this case where changes in chord length and twist are gradual over the span, local angle of attack is determining the shedding behavior. Such an assumption would enable a method based on finding overlaps of

[5] This should be assessed case by case; this result in no way means that Reynolds number effects are never present.

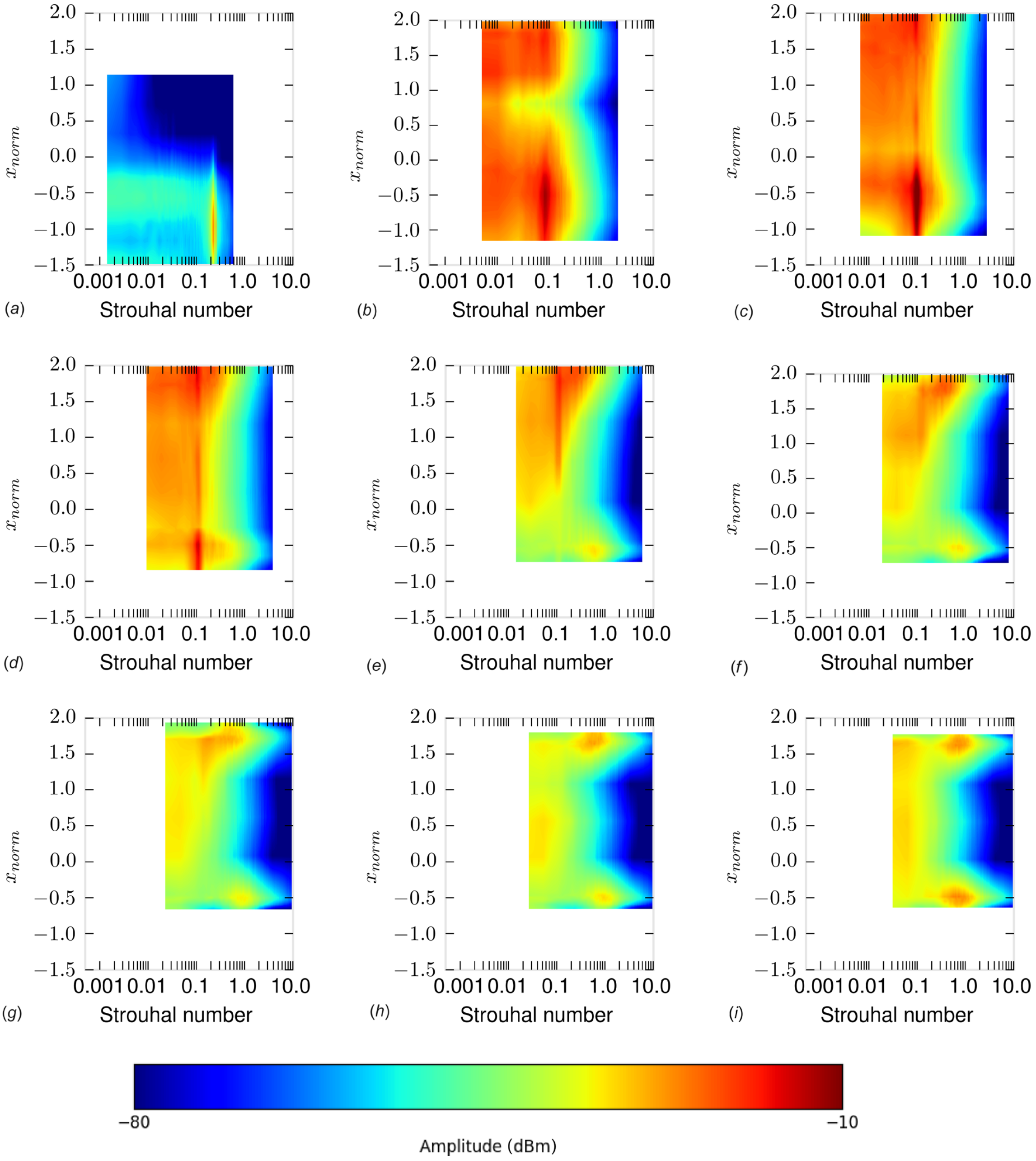


**Fig. 15 V component wake velocities at a height of 360 mm (above reference) for angles of attack from 0 deg to 90 deg at an inflow of 14m/s with Gurney Flaps: (*a*) $\alpha_{\text{local}}$ 2.6 deg, (*b*) $\alpha_{\text{local}}$ 9.4 deg, (*c*) $\alpha_{\text{local}}$ 12.4 deg, (*d*) $\alpha_{\text{local}}$ 17.4 deg, (*e*) $\alpha_{\text{local}}$ 27.4 deg, (*f*) $\alpha_{\text{local}}$ 37.4 deg, (*g*) $\alpha_{\text{local}}$ 47.4 deg, (*h*) $\alpha_{\text{local}}$ 57.4 deg, and (*i*) $\alpha_{\text{local}}$ 87.4 deg**

the shedding frequencies and the structural frequencies at each chordwise station. Such a method would be suitable to highlight cases that need further simulation during load case assessments.

**Passive Flow Control Elements.** While the authors had expected vortex generators to have an effect on the onset of wake mode stall and then transition to shear layer stall, this effect was not observed in the data due to the sparse test matrix. Figure 14 can be compared with the baseline measurements (Fig. 10) to demonstrate that no observable difference was made by the VGs. It is expected that the onset of wake mode type stall would be affected by the vortex generators but the dataset is too sparse to make any further revelations. However, unless the VGs are on the leading edge it seems safe to ignore them when assessing potential vortex-induced vibrations as the vorticity seems to be localized to the shear layer.

The gurney flaps made a pronounced change to the behavior over all of the angles of attack.

Figure 14(*a*): The flow is attached. At St = 0.2, the shedding frequency of the gurney flap is present. The wake is steered significantly lower than the clean airfoil.

Figures 14(*b*)–14(*d*): Trailing edge separation is already present at 9.4 deg with an St of roughly 0.8, which lower than the clean case. With increasing angle of attack, the st increases gradually to 0.1. Like the clean case, the vorticity is largely concentrated about the trailing edge extending towards the leading edge with increasing angle of attach. The trailing edge vorticity is lower than the clean case.

Figures 10(*e*) and 10(*f*): The leading edge wake mode shedding dominates like in the clean case. The trailing edge shear layer vorticity is located lower than in the clean case. The transition from wake mode to shear layer mode appears to begin a little earlier than the clean case already showing a middle stage in sub-figure (f).

Figures 10(*g*)–10(*i*): The transition to shear layer mode shedding completes and again the leading edge vorticity increases in frequency with angle of attack. The trailing edge vorticity is again located lower than in the clean airfoil.

Figures 1 and 2 already demonstrated how gurney flaps steer the wake and shear layer. This effect is present across all four flow regimes. For the shear layer mode of shedding, the flow visualizations (Fig. 1) and the wake measurements indicate that gurney flaps are acting like a virtual extension of chord length. The current data leave it inconclusive whether this results in a shift in the surface pressure frequencies finally seen by the airfoil as the hotwires were located in the dead zone of the flow for these very high angles of attack. But the results do seem to indicate (Fig. 15).

- Gurney flaps on the suction side of an airfoil can shift the shedding frequency in trailing edge and leading edge stall scenarios.
- Gurney flaps should be considered in the analysis of standstill vibrations as a potential method of shifting frequencies.

## Conclusions

In this study, a test wind turbine blade was tested in high angle of attacks to imitate wind turbine rotor blades in a state of stand still. Flow visualization showed the highly 3D flow pattern at high angles of attack. However, in spite of the 3D behavior, it seems that twisted tapered blade geometries can be for some analysis purposes treated as 2D for sections away from the tip where it appears that tip shedding dominates. Normalizing by Strouhal number was effective with certain flow regimes showing stable shedding frequencies independent of spanwise position, chord length, and Reynolds number(for the limited range tested). Treating each spanwise section as independent will enable wind turbine designers to use a Strouhal number versus structural frequency and mode shape-based approach in order to isolate the combinations of conditions that may create vortex induced vibration problems. This reduced set of parameters can then be investigated with higher order CFD tools.

Passive flow control devices were tested on the blade to see if any changes to the shedding behavior occurred. Vortex generators did not show any significant influence in this experiment and can probably be ignored for the very high angles of attack when conducting analysis. Gurney flaps modified the wake geometry by either moving the shear layer or wake downward. In some flow regimes, the shedding frequencies were also affected and the onset of trailing and leading edge stall was at slightly lower angles of attack than the clean airfoil. It seems that Gurney flaps may have an effect on vortex-induced vibrations and should be investigated further.

## References

[1] Mulleners, K., and Raffel, M., 2013, "Dynamic Stall Development," Exp. Fluids, **54**(2), p. 1469.

[2] Leishman, J. G., Martin, G. L., Aerospace, A., and Meeting, S., 2002, "Challenges in Modeling the Unsteady Aerodynamics of Wind Turbines," ASME Paper No. WIND2002-37.

[3] Gharib, M., and Roshko, A., 1987, "The Effect of Flow Oscillations on Cavity Drag," J. Fluid Mech., **177**(1), pp. 501–530.

[4] Hudy, L. M., and Naguib, A., 2007, "Stochastic Estimation of a Separated-Flow Field Using Wall-Pressure-Array Measurements," Phys. Fluids, **19**(2), p. 024103.

[5] Mulleners, K., Pape, A. L., Heine, B., and Raffel, M., 2012, "The Dynamics of Static Stall," 16th International Symposium on Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal, July 9–12, pp. 9–12.

[6] Manolesos, M., Papadakis, G., and Voutsinas, S. G., 2014, "Experimental and Computational Analysis of Stall Cells on Rectangular Wings," Wind Energy, **17**(6), pp. 939–955.

[7] Hoerner, S. F., and Borst, H. V., 1985, *Fluid-Dynamic Lift: Practical Information on Aerodynamic and Hydrodynamic Lift*, L. A. Hoerner, Berkeley, CA.

[8] Besem, F. M., Thomas, J. P., Kielb, R. E., and Dowell, E. H., 2016, "An Aeroelastic Model for Vortex-Induced Vibrating Cylinders Subject to Frequency Lock-In," J. Fluids Struct., **61**, pp. 42–59.

[9] Besem, F. M., 2015, "Aeroelastic Instabilities due to Unsteady Aerodynamics," Ph.D. thesis, Duke University, Durham, NC.

[10] Besem, F. M., Kamrass, J. D., Thomas, J. P., Tang, D., and Kielb, R. E., 2014, "Vortex-Induced Vibration and Frequency Lock-in of an Airfoil at High Angles of Attack," ASME Paper No. GT2014-25648.

[11] Tang, D., and Dowell, E. H., 2014, "Experimental Aerodynamic Response for an Oscillating Airfoil in Buffeting Flow," AIAA J., **52**(6), pp. 1170–1179.

[12] Mueller-Vahl, H., Pechlivanoglou, Georgios Nayeri, C., and Paschereit, C., 2012, "Vortex Generators for Wind Turbine Blades: A Combined Wind Tunnel and Wind Turbine Parametric Study," ASME Paper No. GT2012-69197.

[13] Dima, C., Manor, D., and Carter, R. L., 1994, "Further Study of Vortex Generators' Effect on Lift and Stall Angle of Attack," AIAA Paper No. 94-0625.

[14] Greenblatt, D., and Wygnanski, I., 2002, "Effect of Leading-Edge Curvature and Slot Geometry on Dynamic Stall Control," AIAA Paper No. 2002-3271.

[15] Bak, C., Skrzypiński, W., Gaunaa, M., Villanueva, H., Brønnum, N. F., and Kruse, E. K., 2016, "Full Scale Wind Turbine Test of Vortex Generators Mounted on the Entire Blade," J. Phys.: Conf. Ser., **753**, p. 022001.

[16] Castro, O., Lennie, M., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O., 2015, "The Use of a New Fatigue Tool (ALBdeS) to Analyse the Effects of Vortex Generators on Wind Turbines," ASME Paper No. GT2015-43198.

[17] Skrzypiński, W., Gaunaa, M., and Bak, C., 2014, "The Effect of Mounting Vortex Generators on the DTU 10 MW Reference Wind Turbine Blade," J. Phys.: Conf. Ser., **524**, p. 012034.

[18] Pechlivanoglou, G., Fuehr, S., Nayeri, C. N., and Paschereit, C. O., 2010, "The Effect of Distributed Roughness on the Power Performance of Wind Turbines," ASME Paper No. GT2010-23258.

[19] Greenblatt, D., and Wygnanski, I., 2001, "Dynamic Stall Control by Periodic Excitation—Part 2: NACA 0015 Mechanisms," J. Aircr., **38**(3), pp. 439–447.

[20] Greenblatt, D., and Wygnanski, I., 2001, "Dynamic Stall Control by Periodic Excitation—Part 1: NACA 0015 Parametric Study," J. Aircr., **38**(3), pp. 430–438.

[21] Pape, A. L., Costes, M., Joubert, G., David, F., and Deluc, J.-M., 2012, "Dynamic Stall Control Using Deployable Leading-Edge Vortex Generators," AIAA J., **50**(10), pp. 2135–2145.

[22] Mai, H., Dietz, G., and Geißler, W., 2005, "Dynamic Stall Control by Leading Edge Vortex Generators," American Helicopter Society 62nd Annual Forum, Phoenix, AZ, May 9–11, pp. 26–36.

[23] Heine, B., Mulleners, K., Gardner, A., and Mai, H., 2009, "On the Effects of Leading Edge Vortex Generators on an OA209 Airfoil," 10th ONERA-DLR Aerospace Symposium, Berlin, Oct. 6–8, pp. 1–12.

[24] Martin, P., Wilson, J., Berry, J., and Wong, T., 2008, "Passive Control of Compressible Dynamic Stall," AIAA Paper No. 2008-7506.

[25] Lennie, M., Wendler, J., Pechlivanoglou, G., Nayeri, C., Paschereit, C. O., and Greenblatt, D., 2017, "Development of a Partially Stochastic Unsteady Aerodynamics Model," AIAA Paper No. 2017-2002.

[26] Liebeck, R. H., 1978, "Design of Subsonic Airfoils for High Lift," J. Aircr., **15**(9), pp. 547–561.

[27] Dam, C. P. V., Chow, R., Zayas, J. R., and Berg, D. E., 2007, "Computational Investigations of Small Deploying Tabs and Flaps for Aerodynamic Load Control," J. Phys.: Conf. Ser., **75**, p. 012027.

[28] Bach, A., 2015, "Gurney Flaps and Micro-Tabs for Load Control on Wind Turbines," Technischen Universität Berlin, Berlin.

[29] Holst, D., Bach, A. B., Nayeri, C. N., Paschereit, C. O., and Pechlivanoglou, G., 2015, "Wake Analysis of a Finite Width Gurney Flap," ASME J. Eng. Gas Turbines Power, **138**(6), p. 062602.

[30] Lennie, M., Bach, A., Pechlivanoglou, G., Nayeri, C., and Paschereit, C. O., 2016, "The Unsteady Aerodynamic Response of an Airfoil With Microtabs and It's Implications for Aerodynamic Damping," AIAA Paper No. 2016-1006.

[31] Wang, K., Riziotis, V. A., and Voutsinas, S. G., 2016, "Aeroelastic Stability of Idling Wind Turbines," J. Phys.: Conf. Ser., **753**, p. 042008.

[32] Pirrung, G., Madsen, H., and Schreck, S., 2016, "Trailed Vorticity Modeling for Aeroelastic Wind Turbine Simulations in Stand Still," J. Phys.: Conf. Ser., **753**, p. 042007.

[33] Leishman, J. G., and Beddoes, T. S., 1989, "A Semi-Empirical Model for Dynamic Stall," J. Am. Helicopter Soc., **34**(3), pp. 3–17.

[34] Wendler, J., Marten, D., Nayeri, C. N., Pechlivanoglou, G., and Paschereit, C. O., 2016, "Implementation and Validation of an Unsteady Aerodynamics Model," ASME Paper No. GT2016-57184.

[35] Skrzypi, W., Gaunaa, M., Sørensen, N., Zahle, F., and Heinz, J., 2014, "Vortex-Induced Vibrations of a DU96-W-180 Airfoil at 90° Angle of Attack," Wind Energy, **17**(10), pp. 1495–1514.

[36] Skrzypiński, W. R., Guanna, M., Sorensen, N., Zahle, F., and Heinz, J., 2014, "Self-Induced Vibrations of a DU96-W-180 Airfoil in Stall," Wind Energy, **17**(4), pp. 641–655.

[37] Zanotti, A., Nilifard, R., Gibertini, G., Guardone, A., and Quaranta, G., 2014, "Assessment of 2D/3D Numerical Modeling for Deep Dynamic Stall Experiments," J. Fluids Struct., **51**, pp. 97–115.

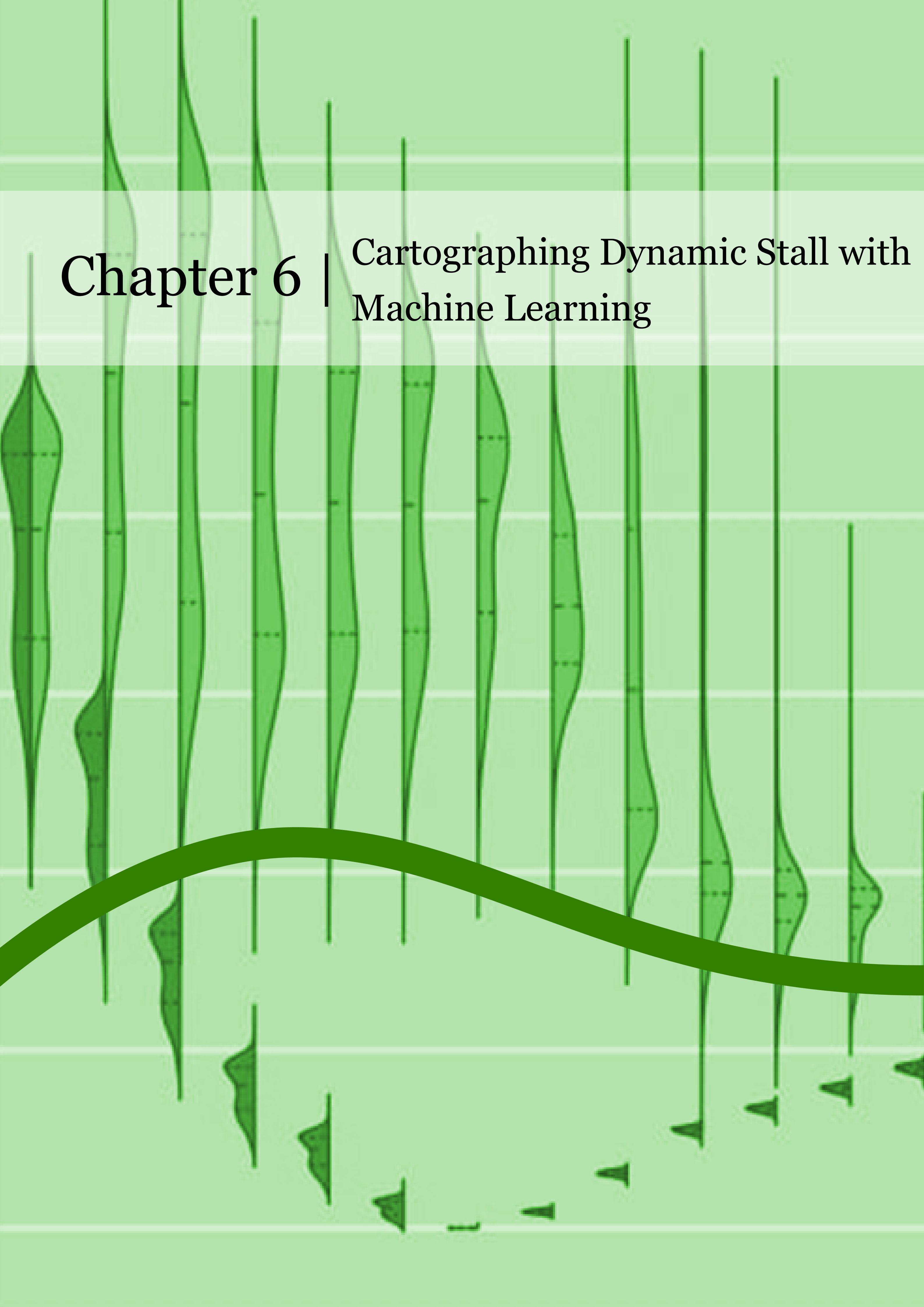
[38] Gaunaa, M., Heinz, J., and Skrzypiński, W., 2016, "Toward an Engineering Model for the Aerodynamic Forces Acting on Wind Turbine Blades in Quasisteady Standstill and Blade Installation Situations," J. Phys.: Conf. Ser., **753**, p. 022007.

[39] Pechlivanoglou, G., Fischer, J., Eisele, O., Vey, S., Nayeri, C., and Paschereit, C., 2015, "Development of a Medium Scale Research Hawt for Inflow and Aerodynamic Research in the TU Berlin Wind Tunnel," 12th German Wind Energy Conference (DEWEK), Bremen, Germany, May 19–20.

[40] Vey, S., Marten, D., Pechlivanoglou, G., Nayeri, C., and Paschereit, C. O., 2015, "Experimental and Numerical Investigations of a Small Research Wind Turbine," AIAA Paper No. 2015-3392.

[41] Mayda, E., Obrecht, J., Dixon, K., Zamora, A., Mailly, L., Sievers, R., and Singh, M., 2013, "Wind Turbine Rotor R & D—An OEM Perspective," AIAA Paper No. 2015-3392.

# Chapter 6 | Cartographing Dynamic Stall with Machine Learning



Wind Energ. Sci., 5, 1–20, 2020
https://doi.org/10.5194/wes-5-1-2020

Open Access
WIND
ENERGY
SCIENCE

# Cartographing dynamic stall with machine learning

**Matthew Lennie**[1], **Johannes Steenbuck**[1], **Bernd R. Noack**[2], **and Christian Oliver Paschereit**[1]

[1]Technische Universität Berlin, Institut für Strömungsmechanik und Technische Akustik, Berlin, Germany
[2]LIMSI, CNRS, Université Paris-Saclay, Bât 507, rue du Belvédère,
Campus Universitaire, 91403 Orsay, France

**Correspondence:** Matthew Lennie (matthew.lennie@tu-berlin.de)

Received: 27 June 2019 – Discussion started: 1 July 2019
Revised: 22 April 2020 – Accepted: 5 May 2020 – Published:

**Abstract.** Once stall has set in, lift collapses, drag increases and then both of these forces will fluctuate strongly. The result is higher fatigue loads and lower energy yield. In dynamic stall, separation first develops from the trailing edge up the leading edge. Eventually the shear layer rolls up, and then a coherent vortex forms and then sheds downstream with its low-pressure core causing a lift overshoot and moment drop. When 50+ experimental cycles of lift or pressure values are averaged, this process appears clear and coherent in flow visualizations. Unfortunately, stall is not one clean process but a broad collection of processes. This means that the analysis of separated flows should be able to detect outliers and analyze cycle-to-cycle variations. Modern data science and machine learning can be used to treat separated flows. In this study, a clustering method based on dynamic time warping is used to find different shedding behaviors. This method captures the fact that secondary and tertiary vorticity vary strongly, and in static stall with surging flow the flow can occasionally reattach. A convolutional neural network was used to extract dynamic stall vorticity convection speeds and phases from pressure data. Finally, bootstrapping was used to provide best practices regarding the number of experimental repetitions required to ensure experimental convergence.

## 1 Introduction

Beyond small angles of attack, airfoil boundary layers have to contend with strong adverse pressure gradients. When the boundary layer does not have enough momentum, a flow reversal occurs and eventually the flow separates from the surface of the airfoil (Abbott and Doenhoff, 1959). Once this occurs, viscous effects dominate and any assumption of potential flow falls apart (Schlichting and Gersten, 2016). This means that modeling separated flows has always been a challenging part of designing wind turbines or even understanding experimental and field data. Even in the age of computational fluid dynamics (CFD), attempts to simulate stall with unsteady Reynolds-averaged Navier–Stokes (URANS) equations have not yet yielded good-quality results (Strangfeld et al., 2015; Rumsey, 2008; Rumsey and Nishino, 2011). Large-eddy simulations (LESs) show promise but are still too computationally expensive to be used as an ordinary design and analysis tool (Rumsey and Nishino, 2011). In the wind industry, semiempirical models (Andersen et al., 2007; Wendler et al., 2016; Holierhoek et al., 2013) are still the main analysis tools for stalled airfoil flows. These models have to make simplifications to be viable in terms of available computational power and input boundary conditions. The key questions are as follows. What information is lost? If we had better understanding and better models, how much could we improve wind turbine designs?

Fortunately, the recent surge in development of machine learning techniques has provided a new set of tools to answer these types of questions. The foundational idea at the basis of this paper is that modern machine learning approaches are accessible to aerodynamics practitioners and can help us better understand experimental data and better recreate that physics in simulations. We will provide a number of demonstrations on how machine learning can help dissect stalled airfoil data. We will also provide a road map for creating a machine-learned semiempirical dynamic stall model. It should be obvious by the end of this paper not only that

Published by Copernicus Publications on behalf of the European Academy of Wind Energy e.V.

these new methods are powerful and accessible but also that they are of vital importance for dealing with airfoil stall.

Stall is the term used to describe a broad range of phenomena that occur during boundary layer separation. There are two broad characteristics that help us provide a loose definition.

1. A flow reversal in the boundary layer results in the stream-wise streamline no longer following the surface of the airfoil (Abbott and Doenhoff, 1959). The region of flow reversal will usually have a neutral pressure.

2. Instabilities, such as shear layer instabilities or wake mode (vortex shedding) instabilities (Hudy and Naguib, 2007), are present. These instabilities make the pressure footprint on the airfoil highly unsteady.

While the following explanations of the categories of stall will dive deep into details, these two features remain the basic underlying phenomena.

Let us begin by considering a stationary airfoil. As the angle of attack increases, the airfoil will encounter trailing-edge (light) stall (McCroskey, 1981). Light stall will develop at moderate angles of attack and is more likely to be present on airfoils with a well-rounded leading edge (Greenblatt and Wygnanski, 2002; Leishman, 2006). The adverse pressure gradient overcomes the momentum of the boundary layer somewhere downstream of the point of minimum pressure (Abbott and Doenhoff, 1959). The vertical size of the viscous region will be on the order of the airfoil thickness (McCroskey, 1982). A well-rounded leading edge will result in a smooth development of trailing-edge stall, whereas a sharp leading edge may cause trailing-edge stall to be bypassed rapidly (Leishman, 2006). The separated region will not contribute to the lift, implying a smooth roll off of the lift, increase in drag and a nose-up moment. Even on a stationary airfoil, the boundaries of the separated region will be unsteady (Mulleners and Rütten, 2018) and will vary along the span and chord.

At higher angles of attack, deep stall will develop on the airfoil (McCroskey, 1982). Deep stall is characterized by separation occurring at the leading-edge region. As the angle of attack increases, the point of minimum pressure will move closer to the leading edge as the stagnation point moves more towards the pressure side of the airfoil (Abbott and Doenhoff, 1959). Here the airfoil leading-edge geometry is critical as a tight radius will cause a stronger adverse pressure gradient which can lead to deep stall initiating from the leading edge, thus bypassing light stall. Even though the stall occurs at the leading edge, the definition of "leading-edge stall" usually involves a laminar bubble bursting but the mechanism can more simply be trailing-edge stall that engulfs the entire suction side of the airfoil (Leishman, 2006). In the steady case, deep stall will cause a plummet in the lift being produced and a sharp increase in drag. The vertical size of the viscous region will be on the order of the airfoil chord (McCroskey, 1982). The viscous region will be home to various instabilities such as shear layer mode or wake mode shedding (Hudy and Naguib, 2007), essentially different types of shedding phenomena leading to fluctuating airfoil forces.

Flow that detaches from the leading edge can reattach due to transition of the shear layer (Abbott and Doenhoff, 1959) or a re-thickening of the airfoil; for example, wind turbine airfoils can have dents due to manufacturing (Madsen et al., 2019). This phenomenon is called a separation bubble. Bubbles are a sensitive phenomenon and small changes to boundary conditions can make them disappear completely (Ward, 1963). Inflow turbulence, leading-edge surface erosion, fouling or ice will often cause forced transition (Pires et al., 2018). Earlier transition will tend to reduce or remove bubbles (Ward, 1963). Even without outside influences, bubbles are an unstable phenomena due to shear-layer disturbances which lead to transition and eventual reattachment or bursting (Kirk and Yarusevych, 2017). For certain older airfoil families, i.e., NACA 63-2nn, the presence or lack of a bubble may cause an airfoil to switch between leading and trailing-edge stall; this phenomenon is known as double stall (Bak et al., 1998). While double stall might no longer be as relevant in new generations of airfoils on wind turbines with pitch regulation, bubbles can affect stall behavior and the eventual performance of the airfoil.

What happens when the airfoil starts moving? When an airfoil moves from low angles of attack into light stall regimes, there will be a phase lag between the angle of attack and the separation. This effect becomes stronger as the airfoil pitches faster and can be seen as a resistance to stall when compared to the stationary case. One can interpret this effect in a few ways:

1. The wake has not yet forgotten the previous flow arrangement, meaning the effective angle of attack is still catching up with the geometric angle of attack, i.e., circulatory lift delay.

2. The current boundary layer still has the higher momentum from the former more favorable flow state.

3. The surface of the airfoil accelerates the boundary layer during the motion.

When moving from light stall angles of attack down to attached flow, the flow attachment is delayed for the same reason. This appears in polar diagrams as hysteresis loops but can also be interpreted as a dangerous phase difference between the angle of attack and the lift, moment and drag. In this context, phase differences mean that the structure will absorb or dissipate energy (Bowles et al., 2014; Lennie et al., 2016). In short, this phase difference can lead to single-degree-of-freedom pitch flutter also known as stall flutter (McCroskey, 1982). If the unstable nature of separated flows leads to the extent and phase of light stall being variable between cycles of pitching, then it follows that the aeroelastic damping of the airfoil will also be variable between cycles[1].

When an airfoil moves rapidly from attached flow into deep stall, it creates an effect known as dynamic stall. The separation moves rapidly from the trailing edge up to the leading edge; the shear layer becomes unstable and then rolls up into a vortex with a strong low-pressure core (Mulleners and Raffel, 2013). The vortex then travels downstream, causing a spike in low pressure across the airfoil, which presents as a strong spike in lift and a strong dump in the moment. A full description of dynamic stall would be extraneous here but excellent reviews can be found in McAlister et al. (1978), McCroskey (1982), McCroskey (1981), Leishman (2002) and Carr (1987). More modern experimental works can be found in Granlund et al. (2014), Mulleners and Raffel (2013), Mulleners et al. (2012), Mulleners and Raffel (2012), Müller-Vahl et al. (2017), Müller-Vahl et al. (2015), Strangfeld et al. (2015), Balduzzi et al. (2019), and Holst et al. (2019). For the discussion here, it is sufficient to note that as the strength and phase of the leading vortex varies, so will the aeroelastic stability.

To review the previous section,

1. there are different types of stall that occur differently in static or dynamic conditions,
2. the spatiotemporal variation is in both span and chord, and
3. differences in stall behavior will also lead to changes in aeroelastic stability.

So how are these variations treated? Treating stall as a stochastic process is a relatively recent idea. As early as 1978, one sees acknowledgment that stall is variable in literature such as McAlister et al. (1978), an experimental report that described taking measurements of 50 cycles of a pitching airfoil undergoing dynamic stall to ensure convergence of the lift. While these researchers did acknowledge the variability of the data, they still used a simple average to represent the data. This was a reasonable choice at the time given that many of the more advanced tools now available did not exist nor was the requisite computational power available. Only more recently have researchers begun to address the spatial and temporal variability of stall in experimental work. Mulleners and Raffel (2013) were able to show that dynamic stall could be described by two stages of a shear layer instability and that the development of these instabilities varied across cycles. In light stall, it was shown that the trailing-edge separation region had two modes, resulting in either a von Kármán shedding pattern or a stable dead water zone (Mulleners and Rütten, 2018). The separation pattern fluctuates unreliably and when vorticity is present, the vortex convection speed is also variable.

Experimental data from Manolesos serve as a detailed reminder that stall happens in three dimensions (Manolesos et al., 2014; Manolesos, 2014). Even on a simple 2D wind section, flow visualization showed four different separation patterns (Manolesos, 2014). These patterns are referred to as stall cells, and they create complicated vortex patterns on and behind the airfoil. Even more complicated still are the separation patterns on wind turbine blades due to the changes of airfoil shape, twist and chord length (various surface visualizations can be found in Manolesos, 2014; Lennie et al., 2018b; Vey et al., 2014). Wind turbines uniquely experience very high angles of attack, where the spatial patterns create further complications (Skrzypinski et al., 2014; Skrzypinski, 2012; Gaunaa et al., 2016; Lennie et al., 2018b). The picture that should now be clear is that stall is a continuum of behaviors rather than a small number of defined cases.

So variability is rampant in stall. How should we measure and interpret airfoil stall behavior? This paper will attempt to demonstrate that machine learning has provided a new set of tools that can be helpful for these very tasks. This paper will demonstrate

1. a clustering method to group similar time series together,
2. a computer vision method for extracting vortex convection speeds from pressure data, and
3. how to detect outliers and inspect the convergence of the dataset.

Furthermore, we will provide a future perspective on the way that machine learning may help us in modeling airfoil stall in simulations. While the specific methods used in this paper should prove to be useful and while we will point out some specific aerodynamic effects in the examples section, these are only examples. This paper is trying to communicate that machine learning more broadly is approachable and useful for unsteady aerodynamics, wind energy and other adjacent fields.

Before jumping into the new methods we should establish what kind of techniques have been used previously. It should be clear given the discussion so far that simple averaging or even phase averaging will remove important data (Riches et al., 2018). In dynamic stall, for example, phase-averaged flow visualizations and pressure data appear vastly cleaner and more coherent than a single cycle. The cycle-to-cycle variations and outliers are an important part of the dataset and should not be smeared out. Manolesos (2014) suggested conditional averaging to produce better airfoil polar diagrams. Mulleners and Rütten (2018) also performed a kind of conditional averaging using the orbits of POD coordinates displayed onto recurrence plots. Furthermore, Holst et al. (2019) also suggest a binning approach, especially when considering very deep stall. Conditional averaging is an interesting

[1] While we may normally consider an operating state to be stable or unstable on a long range trajectory, we may have to consider that each operating state can display short-term behaviors that appear unstable.

approach, but the important question becomes the following: what rules should we use to split the data and is it possible to automate this process to some degree?

Fluid dynamics has always been a natural case for dimensionality reduction. In particular, there is abundant literature using singular value decomposition (SVD) methods such as proper orthogonal decomposition– principle component analysis (POD–PCA) (Taira et al., 2017), dynamic mode decomposition (DMD) (Schmid, 2010; Kutz et al., 2015; Brunton et al., 2015) and spectral proper orthogonal decomposition (SPOD) (Sieber et al., 2015). These methods generally do not perform well in cases with any kind of traveling wave behavior (Taira et al., 2017; Riches et al., 2018; Hosseini et al., 2016). The reason for this lies in the creation of fixed spatial functions and basis functions. If the shedding is consistent, the system will be sparse, a sensible reduced-order system can be found. However, introduce phase jitter and the small number of basis functions no longer does a good job in representing the shedding; so more mode shapes are needed. Even for a simple cylinder shedding, up to 50 modes were required to represent the system reasonably well (Loiseau et al., 2018). Dynamic stall convection velocities vary continuously (Mulleners and Rütten, 2018); therefore we cannot expect a sparse set of spatial functions to represent the system well.

Fortunately the SVD and simple averaging-type methods are not the only forms of dimensionality reduction techniques available. It turns out the dimensionality reduction is a cornerstone technique of machine learning; an interactive summary can be found on Christopher Olah's website (Olah, 2019). In this paper, we will show how multidimensional scaling (MDS) (O'Connell et al., 1999) and clustering (Maimon and Rokach, 2006) can be used as a reliable analysis technique for airfoil stall. Nair et al. (2019) have demonstrated one approach to clustering for separated flows in the context of cluster-based feedback control. Cao et al. (2014) also demonstrated the use of time series clustering in the context of combustion. The advantage of cluster-type methods is that they break the data down into similar neighborhoods rather than assuming that a set of global basis functions can describe the whole domain. Both Loiseau et al. (2018) and Ehlert et al. (2019) have demonstrated that local linear embedding (LLE), a neighborhood-type method, can create a sparse representation of the system. In this paper, we will focus on clustering and MDS, although other methods also show promise.

The MDS and clustering methods rely on a distance metric to gauge the similarity between the time series of lift of various experimental repetitions. As already discussed, the data will contain phase jitter which may cause simple distance metrics such as Euclidean metrics to overestimate the difference between cycles (Ratanamahatana and Keogh, 2004). The problem is amplified by the strong gradients present around the time of vortex convection. This is a common time series problem, and dynamic time warping (DTW) was created for this purpose (Morel et al., 2018; Ratanamahatana and Keogh, 2004). DTW allows for the time series to be stretched and squashed a small amount to allow for an effective comparison between experimental repetitions. The approach of using a cycle-to-cycle distance metric (in this case DTW) is different to making time-independent clusters used in the work of Nair et al. (2019). The difference in approach comes from intended application. In this paper, we will create clusters and MDS plots by comparing the entire time series of separate pitch cycles.

Methods such as clustering and MDS belong to a branch of machine learning called unsupervised learning, i.e., learning from the data without having the answer ahead of time. Conversely, supervised learning uses a labeled dataset to learn a mapping between input and outputs. Once a model is trained, we can then map new data. This is the nature of our second example, extracting the vortex convections from pressure data. We manually create a small set of examples by clicking on the vortex patterns. We then use these data to train a model that can do the same job over the whole dataset efficiently. Manually clicking on the patterns is a laborious, time-wasting and unpleasant task. For these reasons, we want to do this only for the bare minimum number of examples. Fortunately, we can leverage the concept of transfer learning to minimize the effort.

The concept of transfer learning exploits the fact that once a model has been trained for one task, it can be easily remolded to complete similar tasks (Brownlee, 2017). In practice this means that a neural network can be trained for a specific computer vision task and then easily be reused; i.e., a network originally trained for classifying breeds of dogs within photographs can be easily reused on aerodynamics data (the FASTAI project has a lecture series expanding at length on this theme; Howard et al., 2019). This may seem like an exotic claim but there is solid reasoning underpinning the claim. Pictures are displayed in pixels, which is an incredibly high-dimensional space (modern cameras have a 10 MP range). If we randomly choose pixel values, the chances of getting a sensible picture are almost zero; we would usually only get noise. This means that sensible pictures with geometric features such as lines and circles exist in an incredibly small neighborhood. That is to say, any real picture (of an elephant, a calculator, a cloud or even a plot of our pressure vs. time) is more similar to any other real picture than it is to a picture of the kind of random static noise we know from old television sets. Why does this matter? It means that we can use any general picture dataset to get our neural network to the right neighborhood, that is, being able to recognize real geometry. It turns out that as far as the neural network is concerned, the pressure plots look close enough to real world pictures that it only needs a small amount of retraining. Therefore, instead of requiring millions of training data examples, we only needed roughly 700.

In this paper, we will demonstrate the utility of transfer learning by using a pre-trained convolutional neural network (CNN) to extract vortex convection speeds from airfoil pressure plots. A huge challenge of working with experimental data is that it is exceptionally difficult to extract features from data in an automated fashion. One example of this is extracting the convection speed of a vortex from pressure data. To the human eye it is a fairly obvious stripe in the pressure plot; however it is challenging to extract this feature automatically based on basic rules. Computer vision machine learning is perfect for such cases. While the vortex convection speeds are themselves an interesting result, the example should demonstrate to readers the incredible power of using pre-trained neural networks for extracting features from data. Deep neural networks are becoming increasingly used within the wind industry for applications, e.g., for predicting rotor icing (Yuan et al., 2019), power-curve estimation (Kulkarni et al., 2019) or even for rotor–blade inspections (Shihavuddin et al., 2019). We hope to demonstrate with this paper that modern machine learning tools and infrastructure can provide a useful boost to research in unsteady aerodynamics, wind energy and other adjacent fields.

## 2 Experimental data

Most machine learning methods are heavily reliant on an initial dataset[2]. The analyses shown in the rest of this paper rely on two existing datasets. We use these two datasets to demonstrate different approaches as they feature different interesting effects. The wind tunnel dataset is the primary data source and unless explicitly stated will be used in all figures, graphs and discussions. The towing-tank dataset provides a great example for outlier detection. The following introductions aim to provide some context but do not exhaustively describe the experimental setups or the data they retrieved. The original references provide a far more detailed view into the setups.

### 2.1 Wind tunnel

The first dataset was collected by Müller-Vahl (2015). Extensive unsteady aerodynamic experiments were conducted in a blowdown wind tunnel powered by a 75 kW backward bladed radial blower. The test section is depicted in Fig. 1 and is 610 mm per 1004 mm. The model is mounted on two circular, rotatable plexiglas windows and the wind speed is measured with two hot-wire probes. The pressure around the model is captured by 20 pressure sensors on both suction and pressure sides (40 in total). The NACA 0018 airfoil model has two control slots at 5 % and 50 % chord for additional blowing. The model has a chord length of 347 mm and a span of 610 mm. More information about the tunnel can also be found in Greenblatt (2016), and excerpts of the dataset can be found at https://www.flowcontrollab.com/data-resource (last access: 13 September 2019).

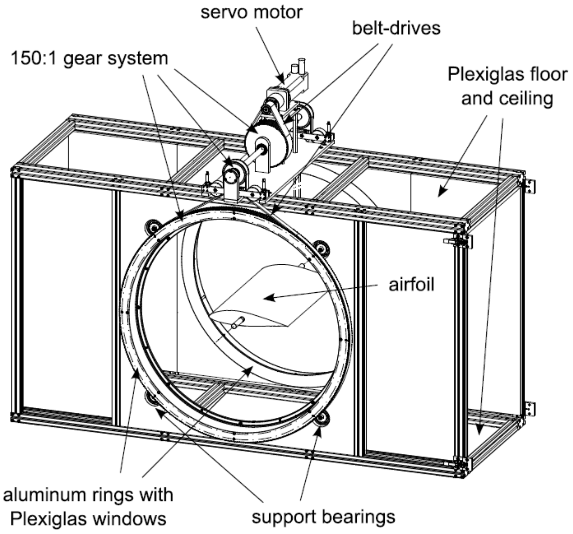

servo motor
belt-drives
150:1 gear system
Plexiglas floor and ceiling
airfoil
aluminum rings with Plexiglas windows
support bearings

**Figure 1.** View of the test section showing the pitching mechanism and the approximate location of the airfoil model. From Müller-Vahl (2015).

The wind tunnel data cover a comprehensive collection of experiments with varying boundary conditions. The dataset has been thoroughly explored in previous publications and appears to be of good quality. It ranges from static baseline investigations over oscillating pitching and variation in free-stream velocity (and a combination of both). In order to manipulate the boundary layer, blowing was added. One peculiarity of this dataset is that boundary layer tripping can be induced by the taped-over blowing slots on the suction side of the airfoil. For the purposes of our analysis, this detail was not critical.

### 2.2 Towing tank

The second dataset comes from a large towing-tank facility at the Technische Universität Berlin. This dataset is used to demonstrate outlier detection as the test configuration used in these data did have some peculiar stall behavior on some cycles. The water tank dimensions are 250 m in length, 8.1 m in width, and about 4.8 m in average depth. A carriage runs on rails, towing a rig (and the model) through the water with a maximum speed of 12.5 m s$^{-1}$. On it the complete measuring system is installed. The rig consists of two side plates with a length of 1.25 m, a height of 1 m and a thickness of 0.035 m prohibiting lateral flow around the model. In between the side plates, the model, with a span size up to 1 m, can be inserted at arbitrary angles of attack. The model resembles a flat plate with an elliptical nose and blunt trailing edge. It has a span of 0.95 m, 0.5 m chord and a thickness of 0.03 m. The surface

[2] Most but not all. For example, reinforcement learning can use self-play as a training mechanism.

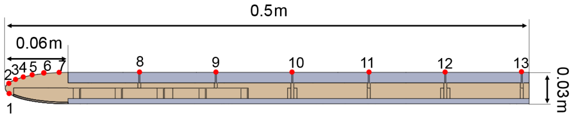


**Figure 2.** Cross section of the mounted flat plate. Red dots indicate position of pressure sensors. From Jentzsch et al. (2019).

is covered in aluminum, and 12 pressure ports are inserted at the specified locations in Fig. 2. The airfoil model is an unusual form but only some qualitative demonstrations are made with this dataset. A more detailed description is given in Jentzsch et al. (2019).

## 3 Machine learning approaches

In this paper, we aim to provide a demonstration of a few machine learning methods and how they can be applied to unsteady aerodynamics data. A brief overview of the algorithms is provided to give a sense of what each of the algorithms is doing. The first algorithm demonstrates how to train and use a relatively simple machine learning algorithm, clustering, from scratch. The second example demonstrates the more advanced deep-learning approach and shows a few tricks to make it possible to do so with a modest amount of data and computational power. Usually each task will call for a different algorithm and different approach, but many of the principles discussed in the following section should transfer well onto other problems. This is especially true for the deep-learning training tricks.

### 3.1 Dynamic time warping, clustering and multidimensional scaling

In this section, we will describe a method of grouping similar data together called clustering. For clustering to work we need two parts, a distance metric/measurement and a clustering algorithm. The distance metric gives us a measurement of similarity between our data[3]. The clustering algorithm takes the distances and groups the data into clusters.

Dynamic time warping (DTW) is a distance measurement that allows for squashing and stretching of the time series in order to reach a best fit. In practice, it is comparable to taking a winding path through a grid where each box corresponds to a time step from the two paths being compared (see Fig. 5). The dynamic time warping algorithm is particularly useful in this case because it will still indicate that time series are similar even if there is a slight phase difference in vortex shedding or other stall phenomena (this effect is shown in the wave form in Fig. 5). The general rule of thumb is that a small amount of warping is a good thing; a lot can end up distorting reality. Therefore, DTW algorithms are usually implemented with either global or local constraints, and these constraints have a bonus of increasing the computational efficiency.

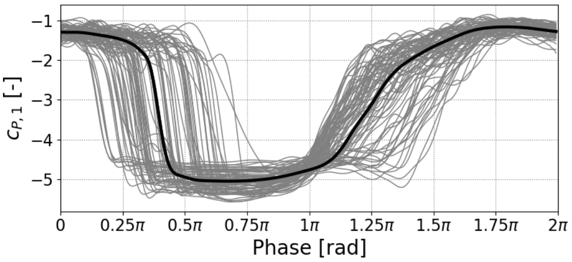


**Figure 3.** Soft-DTW centroid for clustered time series with strong phase jitter. (Example data from pressure sensor reading from towing tank.)

A useful extension to the DTW algorithm creates a composite of multiple time series called a centroid (see Fig. 3). Normally the problem with dynamic stall time series is that the vortex shedding is smeared out when simple means are taken. The onset of static stall can also appear to be a smooth process rather than a sudden separation that occurs at variable phases across different cycles of the experiments (see Fig. 3). The barycenter extension to DTW creates an average that preserves these features. This means that the resulting centroid will be far more representative of a real stall process. In short, it is just a pseudo-average using different mathematics in the background, but it provides a better answer to the following question: for these boundary conditions, what does the stall process typically look like?

For this research, the soft-DTW algorithm was used to compute the barycenter and was taken from the Python module *tslearn* by Tavenard (2008). The algorithm was first proposed by Cuturi and Blondel (2017). To create the clusters, it is necessary to compare every time series within a group to each other. This means the complexity of the algorithm is $O(N^2)$. Two steps were taken to scale the process; firstly the data were downsampled, thus reducing "$N$", and secondly the code was scaled using DASK (Dask Development Team, 2016). DASK is a Python library designed to parallelize standard Python functions onto cluster architecture. The second step may at first appearance seem extreme. In practice the power required was more than a standard desktop but one or two compute nodes were more than sufficient. For the examples computed in this paper, one to two workers (nodes with eight cores each) would process a single experiment within a few minutes. A combination of parallelization and downsampling was used in this study[4].

[3] In this example, we are comparing a time series of a single experimental repetition against another. Clustering can also work with much more simple distance metrics.

[4] Combining the soft-DTW algorithm and the DASK module did require some programming effort, but as both tools were well developed, the effort was smaller than it perhaps first appears. In particular, tools like DASK allow people with very modest programming skill to run cluster-scale code.

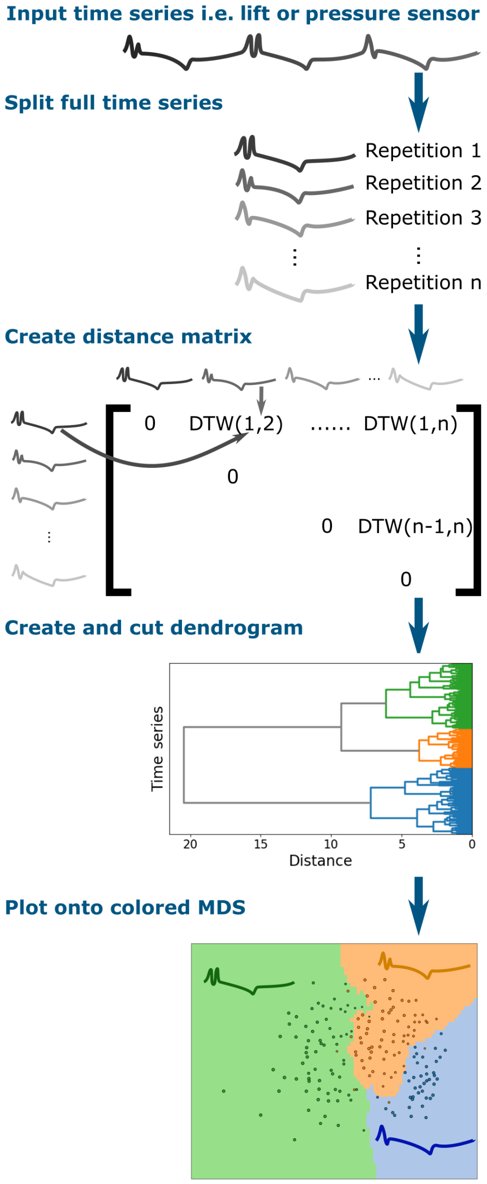


**Figure 4.** Time series clustering algorithm.

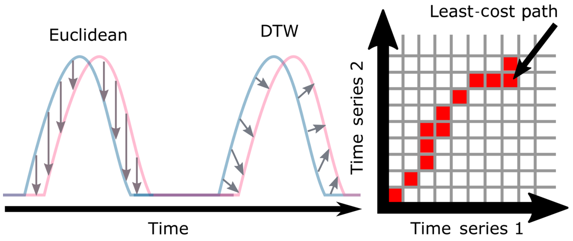


**Figure 5.** Euclidean distance vs. DTW distance between two time series.

Reducing the number of samples gives a significant speed boost as the complexity of the distance measurement is based on the number of time steps. While reducing the sample size, the spectral resolution is reduced about the same factor. The frequency of the expected phenomena limits the amount of downsampling. In order to improve the cluster results the data are, in addition to downsampling, filtered. Dynamic time warping is noise sensitive as the algorithm shifts and bends the time series in order to match similar values. Fortunately, tuning these steps is not difficult as a visual inspection of the resulting data will indicate whether the algorithm is making sensible groups or not. This topic is explored in greater detail in the related work from Steenbuck (2019).

Clustering is a method of dimensionality reduction based on the principle that the dataset can be efficiently described by a set of subgroups. These subgroups are formed on the assumption that the description of the cluster is a useful enough generalization for each member of the cluster. This means that the groups are formed on the basis of similarity. Clustering is an unsupervised method in the sense that there is no correct answer defined ahead of time. Usually unsupervised methods will reveal underlying data structures. This is not to say that we can just passively use these algorithms and useful results will ensue. Each clustering algorithm will perform well for some datasets and will deliver nonsense for others; care is required. To ensure good results, users will usually have to tune hyper-parameters for the dataset, and the simplest of these parameters is the number of clusters. A first estimation about a reasonable number can be made from inspecting the dendrogram or the MDS plot as described below. Another approach is to calculate the mean silhouette score of all elements for a range of cluster numbers (Fig. 6). The silhouette score is calculated by comparing the distance from a data element to its own cluster center to the distance to the center of the closest neighboring cluster (Raschka, 2015). By calculating the mean silhouette score for a number of different clusters, we can see that once we get to four clusters, we only marginally change the quality of the clusters (shown in Fig. 6). This means that breaking the dataset up into further smaller pieces is not going to improve our analysis.

For this application hierarchical clustering turned out to produce groups that were physically meaningful and shared features. Hierarchical clustering creates links between data points (in our case a single cycle of a dynamic stall test) to form a dendrogram as seen in Fig. 4. This process essentially takes the distances that we previously calculated and starts collecting similar data together recursively which is what is shown in the dendrogram. The dendrogram is then cut at a height which results in a given number of clusters. As longer branches indicate bigger differences, the height of cutting should be chosen so that the longest branches are cut. The clustering was implemented using

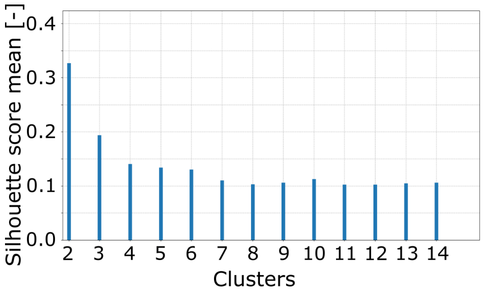


**Figure 6.** Mean of silhouette scores per cluster number.

SciPy's (Jones et al., 2019) hierarchical clustering algorithm (scipy.cluster.hierarchy) with the ward method as a measure for distances between newly formed clusters. Hierarchical clustering was chosen after exploratory analysis showed that other basic algorithms such as *k*-means tended to perform poorly for these data. Fortunately, well-developed machine learning libraries such as Sci-Kit Learn make it very simple to trial different algorithms.

Another way of presenting the data is to use multidimensional scaling (MDS) (O'Connell et al., 1999). MDS essentially takes a cloud of data points with high dimensionality and squashes the points onto a low-dimension plane while attempting to maintain the distance between the points. In our case, each time step of a single series represents a dimension or feature which results in dimensionality that is incredibly difficult to interpret. Now take each series as a single data point and then squash it onto a 2D plane, and the data reveal an underlying structure. We can then color each point and use a *k* nearest-neighbor classifier to color the background as seen in Fig. 4. The resulting point cloud (hopefully) inherits distinct clusters. The number of clusters encountered here gives a good first estimation about a reasonable cluster number for further analysis. So instead of creating a chaos of overlapping time series, the data appear as a low-dimensional representation image with each color representing time series with similar behavior. In some circumstances, the coordinates of the image will even have a clear physical meaning; i.e., dimension 1 could correlate with the Reynolds number. A broad overview of the algorithm used in this paper can be found in Fig. 4.

An example of the cluster analysis is depicted in Fig. 4. In this figure, we summarize all of the time series of a single cluster by displaying only its centroid. We can see that each of the centroids represents a slightly different behavior, particularly during the secondary vortex shedding. Each cluster has a small uncertainty band shown by the standard deviation. As the dataset can be represented by three centroids instead of trying to compress the entire data into a single average, the representation is concise but still provides a more accurate view of the process.

### 3.2 Convolutional neural networks

In the previous section, we looked at how we can cluster together similar experimental samples. This section aims to see if we can extract some interesting features from our data using machine learning. For this example, we will attempt to use computer vision (machine learning applied to pictures) to extract information about the dynamic stall vortex.

Convolutional layers are the special trick that have turned neural networks into a wildly effective computer vision tool (Krizhevsky et al., 2012). Convolutional layers allow pictures to maintain their structure as a grid of pixels. Convolution operations are applied over the picture as a kind of moving window shape filter. The shape filters are learned and often end up resembling recognizable patterns. In the first layer of the network, the filters will be detecting edges, slow gradients and color changes (Zeiler and Fergus, 2013). As we proceed deeper into the neural network, the filters begin to look like natural features such as a birds eye, a bicycle wheel or a doorframe. Each of these filters is created during the training process where large datasets are fed through the network, and the error is propagated backwards through the network to allow for incremental improvement. It is helpful to note that as pictures are just made up of a grid of pixels, a 2D matrix structure (for single channel), in a great deal of cases, data can be represented in this form. This means that computer vision tools can be used on data that can be structured like a picture. Convolutional neural networks are most effective when features are local.

We have discussed neural networks here with a high number of layers. This is referred to as deep learning. Deep learning is a field that has seen rapid innovation due to the abundance of graphical processor units (GPUs) and more recently tensor processing units (TPUs). Platforms such as PyTorch or TensorFlow provide high-level front ends in Python. These front ends abstract away much of the complexity, meaning that users avoid much of the low-level matrix algebra and optimization. Furthermore, it is common practice to publish well-performing neural network architectures that are already pre-trained (transfer learning). Many of the once difficult decisions, such as choosing a learning rate, have now been made simpler with tools such as learning rate finder (Howard et al., 2019). Cheap computational power, easy high-level coding and the advent of transfer learning means that these incredibly powerful tools are now available for aerodynamic applications like detecting boundary layer transition from microphone data (see Fig. 7). These innovations mean that non-machine learning specialists can use deep learning with a low barrier to entry.

In this paper, we will provide an example of turning aerodynamic data into a picture and then using a convolutional neural network to extract useful information. Dynamic stall vortices have a strong low-pressure core which causes a lift overshoot and moment dump. When dynamic stall vortex data are averaged over 50+ cycles, it tends to show dynamic

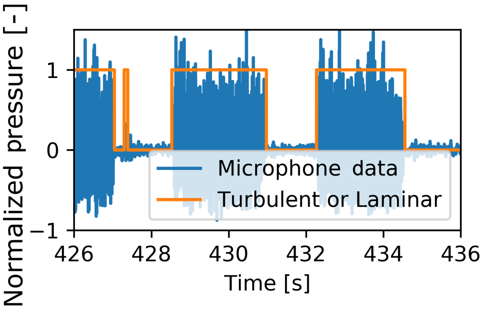


**Figure 7.** Identification of a boundary layer state using a recurrent neural network (data from Bak et al., 2010) (see code example https://github.com/MatthewLennie/Aerodynamics, last access: 13 September 2019).

stall vorticity as far more clean and coherent than is the case for a single cycle. The strength of each vortex, its convection speed and onset of convection vary between cycles. This leaves the following questions. How much do dynamic stall vortices convect differently? Do boundary conditions like the reduced frequency affect the variability?

The dynamic stall vortex feature of a pressure vs. time plot is easily distinguished by the human eye; however, pulling this feature from the data is rather difficult. The authors attempted the task with a number of more simple approaches such as simply finding the peak at each chord-wise position, a Hough transform or even Bayesian linear regression with the pressure plot interpreted as a probability distribution. They all worked for a few cases but failed to generalize and in the end did not perform well enough to be usable. Each vortex is different and therefore manually creating a rule to automatically pull the dynamic stall vortex feature from the data was not trivial. However, this is a standard computer vision task very similar to a driverless car identifying a cyclist in a picture. Fortunately, heavy development in the computer vision field has resulted in some incredibly powerful pre-trained models such as the RESNET family of models (He and Sun, 2016)[5]. The model is a convolution neural network that has been pre-trained on a massive dataset of real world images. This means that the convolutional layers of the network already have a set of shape filters that are broadly applicable to all natural pictures. This means that with a relatively small amount of training data and computational effort, we are able to simply remold the convolutional layers to identify dynamic stall vortices's and give the convection speed and phase.

Pre-trained neural networks can be built and retrained using any of the typical frameworks such as PyTorch, Keras or TensorFlow. In this case, we used a RESNET50 model within the FASTAI architecture which is a high-level interface built on top of PyTorch (Howard et al., 2019). The FASTAI architecture implements several current best practices as defaults such as cyclical learning rates, drop-out, training data augmentation and data normalization. We can think of the pre-trained neural network as a template: most of the training has already been done, and we only need to retrain the network to react correctly to our dataset. This approach is cheap in terms of data volume and computational power.

The final layer of the neural network was replaced with two outputs to represent a linear fit of the vortex convection (slope, offset). For this analysis, acceleration of the vortex was ignored, though the code could be easily extended. The pressure data were represented as a picture where the horizontal dimension represents phase, and the vertical dimension represents the suction side of the airfoil with the bottom of the picture being the leading edge (an example of an already processed picture is in Fig. 8, where the training data do not have the blue line identifying the vortex but are otherwise the same). Training data were created by manually clicking (and storing) the positions of the vortex on 733 images (an attempt with only 300 pictures tended to over-fit on RESNET50 or have high bias on smaller models). The manual clicking does introduce some measurement error, but a few practice runs showed that the error was much smaller than the effect of the physical phenomena. The images were selected from a wide range of cases with randomized test training splitting within each case to ensure good generalization of the fitted model. However, data were limited to examples with a strong wake mode shedding, meaning that the vorticity is easily visible on the pressure footprint. The training was done in two stages, first with the internal layers of the RESNET model frozen. This means we train only the very last layers that output the slope and onset. Once the training error reached stopped improving, the internal layers were unfrozen to mold the internal layers for a small number of epochs (training repetitions).

Initially 80 % of the data were taken as the training set and the training was completed with 20 epochs with the convolutional layers frozen so that the newly added layers could quickly converge. The training was stopped at 20 epochs once the validation error began to increase. The convolutional layers were then unfrozen and the training was continued for a further 20 epochs. During training no geometrical augmentations on the training were undertaken, but the brightness of the images was augmented[6]. The error statistics were still unsatisfactory and additional training did not improve the performance further. However, the current settings of the hyper-parameter settings and training procedure had seemed to extract the best model given the available data. The training procedure was repeated exactly the same a sec-

[5] Note that while we were able to get acceptable results from the RESNET models, a higher level of accuracy may be obtained by network architectures that were built specifically for this kind of localization task.

[6] Geometric augmentations would have been the next method to improve the model if the process had not worked well enough.

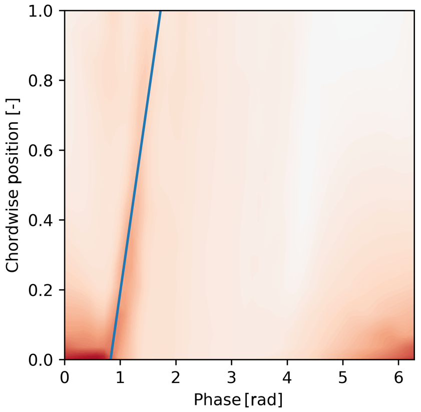


**Figure 8.** Example CNN output. Color intensity refers to suction pressure, and the blue line is regressed fit. Pressure is standardized; therefore the colors represent $Z$ scores. No units or color bar are provided for this reason.

ond time, with the same hyper-parameters and the same number of epochs; however this time the dataset was not split into test and training sets, thus neglecting validation error (a practice described by Goodfellow et al., 2016). This may be perceived as opening up the risk of over-fitting; however the training procedure and hyper-parameters were already tested and the neural network did not over-fit. Furthermore, usually additional data will help reduce over-fitting. We therefore have confidence that with this procedure the validation error will not increase and that the training error is representative across the dataset. Thankfully the additional data did reduce the training error enough to make the model usable (see an example results in Fig. 8). The residuals of both the slope and constant were distributed roughly as Gaussians with standard deviations of 0.15. In total, the training took on the order of 30 min of computational time on a GPU. Readers are encouraged to view the source code at https://github.com/MatthewLennie/VortexCNN (last access: 13 September 2019). The repository contains training sets and final data used to produce the following analysis.

The resulting model incurs a small measurement error so the resulting distributions have be adjusted. Fortunately, the measurement error could be quantified. Both the error and the resulting vortex convection values can be approximated as Gaussian. The real distribution is sought by guessing a distribution, running a Gaussian convolution filter over the distribution and then measuring the difference between the resultant distribution and the data. Essentially, we knew the measurement error distribution roughly, we knew the output distribution, and we can work backwards on a statistical basis. This error term is fed into a optimizer, thus giving an estimation of the real data distribution without the error incurred by the neural network inference. In practice, this reduces the standard deviations of both the slope and intercept by roughly 30 %. We should note that we can not "repair" the measurement data and locate the true convection speed of each measurement, but on a statistical basis, we can get closer to a true estimation. It is also worth mentioning that this neural net will find the speed that the vortex footprint travels across the airfoil, and the vortex will usually have an additional component normal to the airfoil.

The procedure described above represents a first iteration of such an approach, a feasibility demonstration. With some more effort, a better neural network architecture could be chosen and the clicking procedure could be replaced with comparison to flow visualization. With these improvements, we could potentially avoid the final step where we attempt to repair the distributions. We would prefer to remove this final step which forces us to assume that the distribution is Gaussian. Nonetheless, the current model is workable enough for our purposes.

## 4 Examples

So far we have explored the idea that stall is variable as well as a few machine learning methodologies that could help interpret the data. We will now provide demonstrations of both of the algorithms. While the specific results are interesting and we will briefly discuss the physical effects observed by the algorithms, the aim of this section is to provide illustrative examples of the approaches in use. The description of the physical effects is provided merely to motivate that the methods appear to be finding sensible phenomena.

### 4.1 Extracting vortex convection with a convolutional neural network

In this section, we provide a demonstration of the neural network extracting dynamic stall vortices's from the surface pressure of the airfoil–time series data. The time series data come from the wind tunnel dataset which sees an airfoil in a wind tunnel. The airfoil pitches sinusoidally, the free-stream velocity can be changed and the leading-edge blowing is installed. A number of test configurations with dynamic stall were chosen and pushed through the neural network. The first case is relatively complicated, as it features an oscillating inflow velocity (sinusoidal with a variation of 50 % in the mean inflow), pitching into the dynamic stall range (up to 25°) with leading-edge blowing active. Four example tests were compared with different phase differences between the angle-of-attack motion and the inflow velocity. The pitch and blowing phases for each case are shown in Fig. 9. Medina et al. (2018) made a very similar analysis and found that decelerating flow tended to destabilize the boundary layer and encourage earlier separation. With the convection speed and onset data retrieved by the neural network, it is possible to show that this is true in the specific detail of the dynamic stall vortex. Figure 10 shows that for cases where the inflow speed is in phase with the angle of attack, the shedding occurs later.

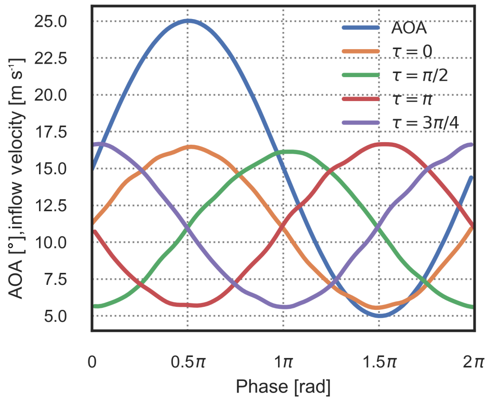


**Figure 9.** Inflow and angle of attack for Figs. 10 and 11.

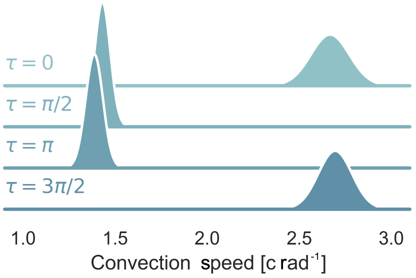


**Figure 10.** Probability distributions of the convection speed of dynamic stall with airfoil blowing different phases of harmonic inflow ($\tau$). $\frac{U_{\text{amp}}}{U} = 0.5$, $k = 0.08$, $Re = 2.5 \times 10^5$ and $\alpha_0 = 15°$, $\alpha_{\text{amp}} = 10°$.

However, when it does finally occur, the vortex will shed at a higher velocity (see Fig. 11). Interestingly the results seem to indicate a much higher variability in the cases where the flow is decelerating during the vortex convection. Figure 14 also shows the relationship between the onset of the vortex shedding and the convection speed. There is a weak correlation ($\sim 0.3$ Pearson metric) but not strong enough with the existing data to make conclusions about the relationship between the two. This first example shows us that we can use a machine learning tool to better understand how our boundary conditions such as inflow velocity affect the physical process. We were able to take a large set of test repetitions and summarize them in a compact yet descriptive manner without having to resort to averaging.

A second example shows the effect of varying only the Reynolds number with constant inflow velocity (see Figs. 13 and 12). We can see that the mean vortex convection velocity scales with Reynolds number as we should expect. The vortex convection onset has a constant variance across both examples (see Fig. 13). However, interestingly the variance of the convection velocity grows with Reynolds number (see Fig. 12). This example shows us to be very careful about how we think about variability and how it applies to each part of the physical process. While these results and the first example's results are interesting and can be expanded upon, the important lesson is that a small data, low computational cost machine learning method was able to help extract a richer set of information from the dataset.

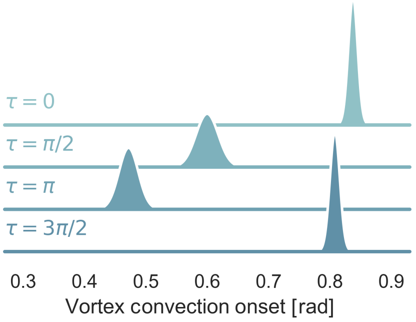


**Figure 11.** Probability distributions of the onset of dynamic stall with airfoil blowing and different phases of harmonic inflow ($\tau$). $\frac{U_{\text{amp}}}{U} = 0.5$, $k = 0.08$, $Re = 2.5 \times 10^5$ and $\alpha_0 = 15°$, $\alpha_{\text{amp}} = 10°$.

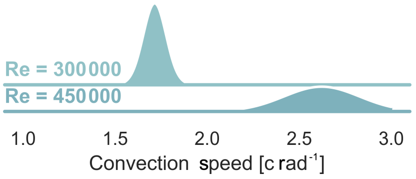


**Figure 12.** Probability distributions of the convection speed of dynamic stall with different Reynolds numbers, $k = 0.09$, and $\alpha_0 = 18°$, $\alpha_1 = 7°$.

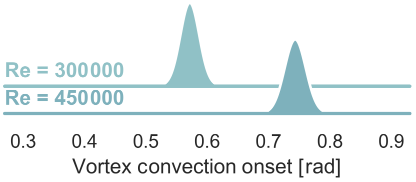


**Figure 13.** Probability distributions of the onset of dynamic stall with different Reynolds numbers, $k = 0.09$, and $\alpha_0 = 18°$, $\alpha_1 = 7°$.

### 4.2 Dynamic stall clustering

In the following section, we have chosen a few examples purely for the purpose of demonstrating the clustering approach and their usefulness in analyzing dynamic stall. In particular, we would like to see if there are distinct behaviors possibly stemming from stall cells or other complex phenomena. By using clustering, we hope to split our dataset into clusters of different airfoil behaviors as far as they exist. This provides us with a way of inspecting the data without having

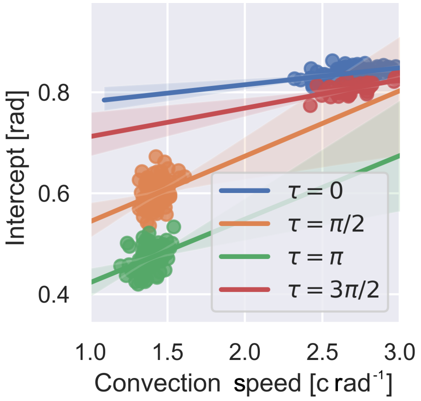


**Figure 14.** Relationship between the onset of shedding and convection speed for a range of blowing cases.

to laboriously compare each time series or to simply inspect averaged data that will hide these effects.

At high angles of attack ($\alpha_0 = 21.25°$ and $\alpha_{\mathrm{amp}} = 8.25°$), we can observe the different kinds of stall behaviors that can occur. Figures 15 and 16 show contrasting behaviors for the same angles of attack. In Fig. 15, a quasi-periodic shedding appears. Without flow visualization it is hard to determine the shedding type, but the pressure footprint shows the vortex as weak and smeared. This kind of footprint would indicate that the vorticity is not close to the surface of the airfoil or is large and not very coherent. This probably indicates that we are seeing a shear layer instability rather than very clear wake mode examples seen in the previous section. The clusters seem to indicate that the shedding behavior is not reliable, with cluster 3 (green) and cluster 4 (red) showing amplitudes of oscillation dissipating rapidly. However, the other two clusters show a more sustained shedding pattern.

Now let us consider a second case with a different Reynolds number and reduced frequency but with the same angle of attack range (Fig. 15). The airfoil moves into stall, releases one (cluster 2 – orange) or two coherent vortices (cluster 1 – blue cluster) and then resolves into weaker small-scale shedding. That is, we are seeing two different shedding patterns for the same boundary conditions.

In Figs. 17 and 18 we can observe the effect of changing the reduced frequency while holding the Reynolds number and angle of attack constant. The first most obvious difference is that the period between the primary and secondary vorticities remains constant. The data do otherwise follow the general wisdom that the lift overshoot will increase with reduced frequency, but it does not happen uniformly. Furthermore, the lower reduced frequency seems to create a much wider variance in the primary stall vortex compared to the higher reduced frequency where both clusters display a strong primary vortex with only a barely visible change in primary stall. Using the clustering method we are also able to reveal that, in both cases, one cluster has a strong secondary vorticity and the other has a nearly nonexistent secondary vorticity (read carefully, colors do not match). Interestingly the higher reduced frequency in Fig. 18 seems to suppress the secondary vortex as Fig. 17 shows strong secondary vorticity in 55.8 % of the cycles and a somewhat weaker secondary vortex for the other cycles.

We have observed with these four example cases that differences in reduced frequency and Reynolds number will resolve into a quite different type of vortex shedding. Furthermore, even within the same case we can see a strong variation in the strength of the shedding mechanism. The instability mechanism driving this shedding is very sensitive to the small variations in input conditions. The shedding mechanisms shown in these four examples are just one of the variety of shedding behaviors.

A quick visual inspection of the time series data would be unlikely to uproot the variable shedding behaviors seen in these two examples. However, the cluster centroids or even simply the MDS plots (i.e., Fig. 20) make the differences clear and easy to interpret. In any of the example cases, phase-averaged results would have been a poor representation of the dataset because we would not have any way of seeing the variable nature of the results. Better information leads to better decisions. For example, when we calibrate simulation tools, particularly empirical unsteady aerodynamic models, we should be aware of where our models will perform poorly because the underlying flow physics is highly stochastic, even showing distinct behaviors. Our model design choices can be more well informed, i.e., choosing to fit a model on the most commonly occurring cluster only or even trying to recreate the variability. We should also be aware that standard measurements of a model's performance such as mean-squared error are only valid for homoscedastic regimes; that is, we expect the same amount of variance throughout the whole range of the model's validity. If we violate this condition, the models will tend to be a poor representation of reality. This is true for fitting machine learning models and also the semiempirical models commonly used in unsteady aerodynamics. Finally, one can easily find examples of experimental field data where clustering would be a powerful data analysis tool, e.g., the double stall measurements from Bak et al. (1998).

## 5 Convergence and outliers

The clustering and MDS can also be used together to qualify outliers that may corrupt the quality of the dataset. For instance in wind tunnels, the first cycles of a test will often be different to later cycles due to the wake effects and dynamics of the tunnel. Similar start-up effects can also be seen in the towing tank. However, more broadly speaking, test data are often plagued with test data poisoned by some sort of external influence. Figure 19 is an example of a single leading-

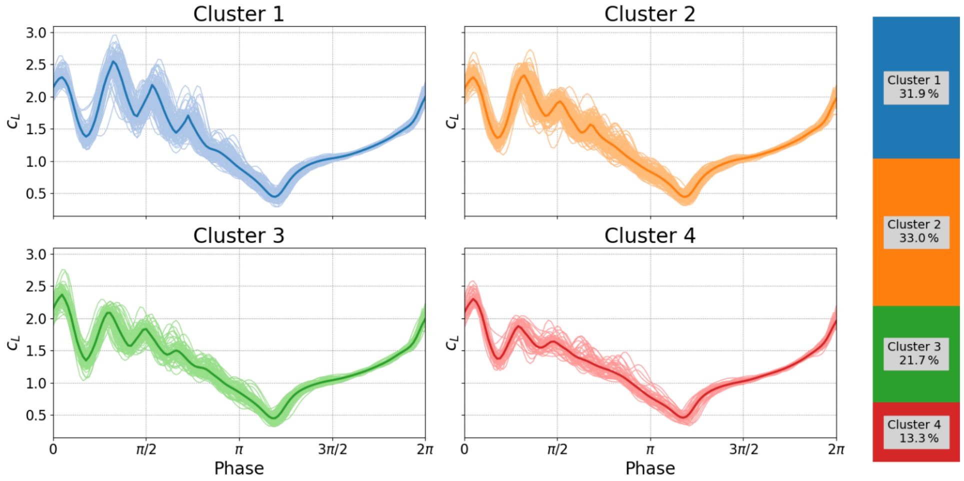


**Figure 15.** Deep stall investigations: cluster analysis for boundary conditions: $k = 0.0992$, $Re = 3.3 \times 10^5$ and $\alpha_0 = 21.25°$, $\alpha_{\mathrm{amp}} = 8.25°$.

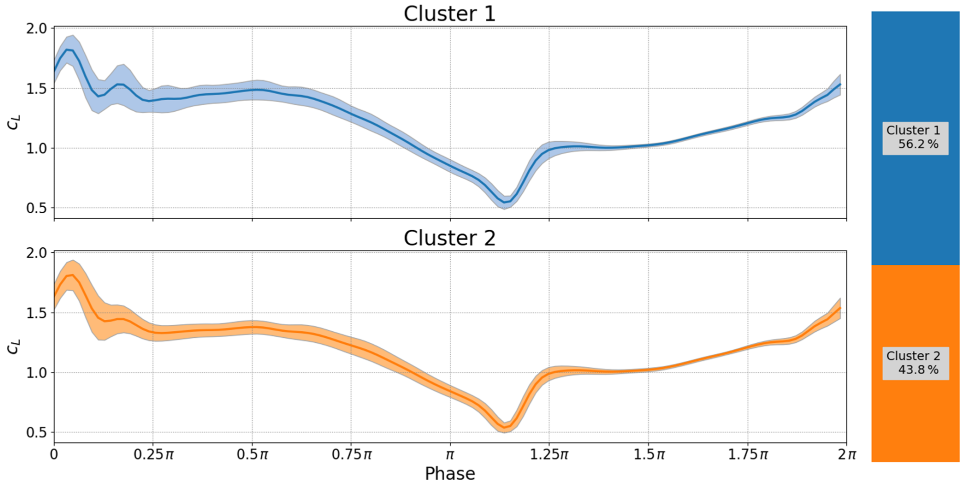


**Figure 16.** Deep stall investigations: cluster analysis for boundary conditions: $k = 0.0574$, $Re = 5.7 \times 10^5$ and $\alpha_0 = 21.25°$, $\alpha_{\mathrm{amp}} = 8.25°$.

edge pressure sensor from the towing tank where obvious outliers are present. The pressure values in the main cluster (blue) show detached flow over the entire cycle. However, a small number of cycles in the green and orange clusters actually reattach. The MDS representation alone (Fig. 20) indicates that it is worth inspecting the data further. Such an obvious representation could speed up the task of possibly pruning the dataset where outliers are created by known effects such as startup or a measurement failure.

It would also be possible to remove outliers automatically based on the cluster data. In practice, this level of automation is not necessary on most experimental setups and the visual inspection provided by MDS and clustering was enough to find outliers quickly and efficiently. On a practical level, it is possible to put the MDS plots into a folder and view the image thumbnails an efficient quality assurance step.

While in this paper we have broadly recommended making cluster-based centroids rather than a mean of the whole dataset, the reality is that the latter is still common practice. McAlister et al. (1978) made the recommendation of taking at least 50 cycles of data to ensure convergence of cases with dynamic stall. The methods used in that paper were limited by available computational power.

Bootstrapping is a method of uncertainty estimation which uses resampling. The concept is quite simple: stick the data in a bucket, resample with replacement until you reach the size of the dataset, and then find your mean, variance and other statistics required. This process is then repeated until a probability distribution of the values is found, very similar to

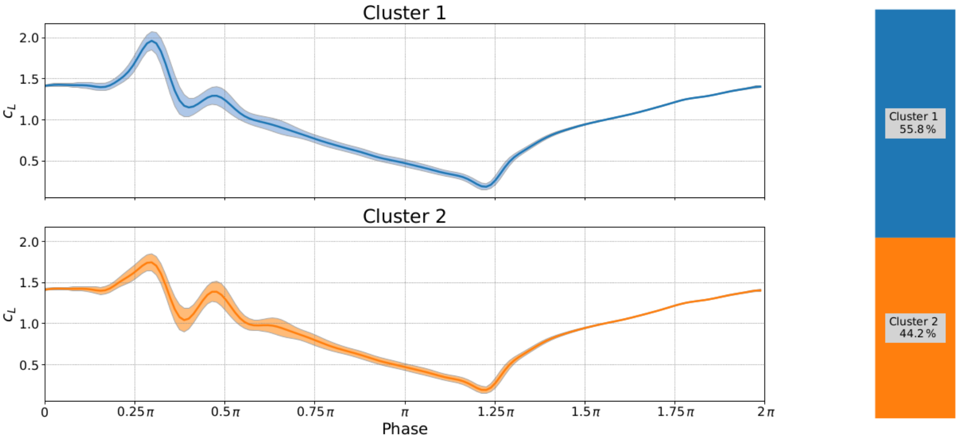


**Figure 17.** Deep stall investigations: cluster analysis for boundary conditions: $k = 0.0992$, $Re = 3 \times 10^5$ and $\alpha_0 = 18°$, $\alpha_{\text{amp}} = 7°$.

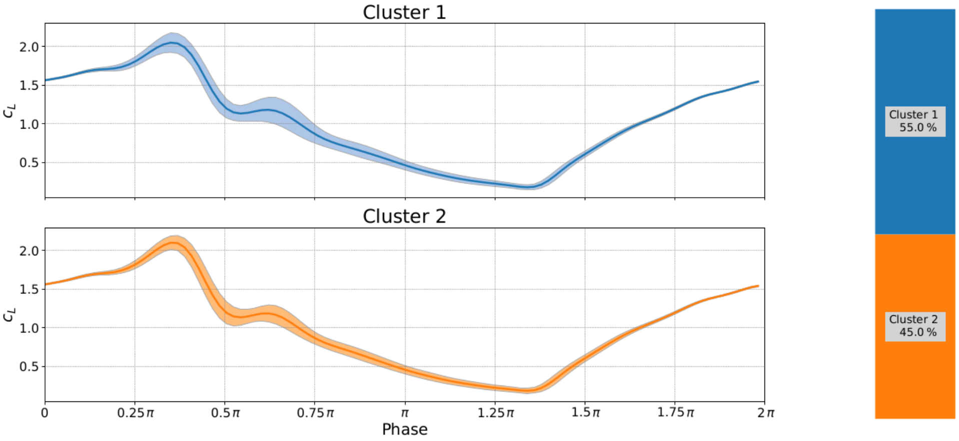


**Figure 18.** Deep stall investigations: cluster analysis for boundary conditions: $k = 0.1346$, $Re = 3 \times 10^5$ and $\alpha_0 = 18°$, $\alpha_{\text{amp}} = 7°$.

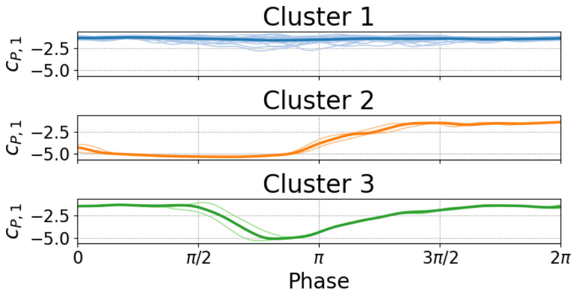


**Figure 19.** Clustered time series from towing-tank surge experiment. Boundary conditions: $U_\infty(\phi) = 2.5\,\text{m}\,\text{s}^{-1} + 0.7\,\text{m}\,\text{s}^{-1}\sin(\phi)$, $\overline{Re} = 1.25 \times 10^6$, $f = 0.21$ Hz, $\alpha = 10°$.

the concept of confidence intervals. This provides us a quantitative statement such as “the existing data indicate 90 % of the time that the mean lies between 0 and 1”. Bootstrapping has some nice mathematical properties mostly propagating from central limit theory. A good treatment of the subject is given by Chernick (2008).

In our case, we would like to see how the uncertainty of our population estimates decreases as we collect more data. To do this, we repeat the bootstrapping process, pretending at each step that we only have a given number of cycles. This results in a graph comparing uncertainty to number of cycles available (see Figs. 21 and 22). One will note that the variance and interquartile ranges converge slower than the mean and median. This is due to the simple fact that the central moments of the distribution will collect more data more quickly and will therefore converge with fewer data. In practice this

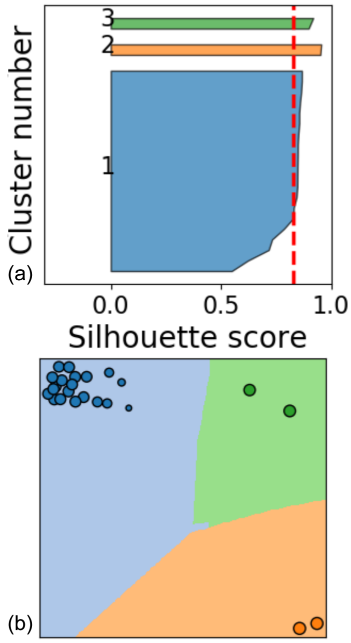


**Figure 20.** **(a)** Silhouette samples per cluster. **(b)** MDS representation. Data from towing-tank surge experiment.

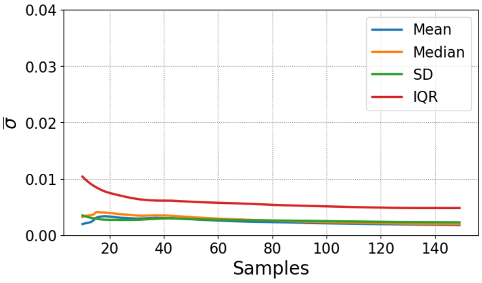


**Figure 21.** Convergence of the population estimates for a light stall case as the number of tests increases.

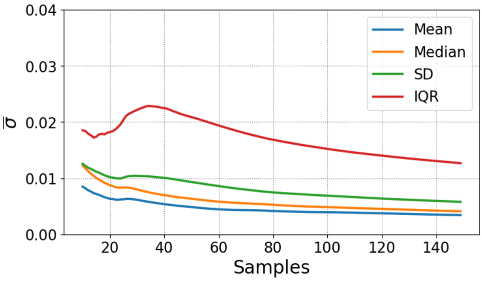


**Figure 22.** Convergence of the population estimates for a deep stall case as the number of tests increases.

means how much data you need will depend on whether you need the central moments or the extreme events.

Lennie et al. (2017) demonstrated that when considering stall, it is probably best to avoid using mean and variance due to the non-Gaussian spread of the data. Median and interquartile range will serve better in cases of stall. All of the population estimates are presented here, as percentile based estimates such as median and interquartile are still rarely used in literature. Representing the variability with a non-parametric distribution (kernel density estimate) gives the best representation and can be achieved with violin plots (see examples in Lennie et al., 2017). The error itself is based on the temporal mean of the respective estimate throughout the time series. A similar convergence approach was used in Lennie et al. (2018a).

A number of test cases were chosen with varying degrees of separation. In deep stall cases, as seen in Fig. 22, the error of the standard deviation drops below 2 % after ∼ 60 repetitions. The light stall case in Fig. 21 shows quick convergence at low values. Already after 20 repetitions all errors are below 1 %. In cases with unsteady inflow, the normalization of aerodynamic coefficients with the inflow speed can amplify experimental noise and therefore converge slower than expected. It may be possible to converge the inflow speed and lift values separately and then apply normalization to speed up convergence. Of course different levels of confidence would require more or fewer repetitions; however, for general purposes the following principles can be made:

1. For deep stall use < 60–100 cycles.
2. For light stall use < 20 cycles.
3. Be careful in cases with unsteady inflow; even attached flow can take up to ∼ 40 cycles to converge.

These principles should be read in the context of the limited example given here. In most of the examined cases, the variability and thus the rate of convergence were reduced with higher Reynolds numbers. The higher the angle of attack, the more pronounced the effect. The convergence may be influenced further by the reduced frequency and the addition of flow control elements. It is always best practice to conduct the bootstrapping for each new test configuration.

While a main recommendation from this paper is to use clustering to represent data, simple averaging will remain a popular analysis tool. However, we advocate using bootstrapping to at least help quantify the uncertainty of the averaging and clustering to find outliers. Even if the final analysis will be conducted on averaged data, the steps outlined in this section will still help isolate problems with the dataset[7].

[7] An extended set of results can be found in the master's thesis of Steenbuck (2019).

## 6 Potential new dynamic stall modeling approaches

The marriage of data science and aerodynamics presented in this study has so far been an exercise of data visualization. However, machine learning tools can also be useful for other tasks such as robust dynamic prediction (Brunton and Noack, 2015). The natural extension of this study would be to create a new generation of unsteady aerodynamics models using machine learning techniques. A challenge to the current stable of unsteady aerodynamic models is modeling vortex-induced vibrations. Wind turbines can be exposed to very high angles of attack, particularly during construction and shutdown, and furthermore the blades are relatively flexible, giving rise to vortex-induced vibration problems (Lennie et al., 2018b). There are complex relationships between the operating regime and the wake structure (Lennie et al., 2018b). The empirical models[8], such as the Beddoes–Leishman model, will be very difficult to extend to handle vortex-induced vibrations given the fact that shedding behavior varies strongly with many of the input conditions and therefore will be hard to encode into a readable set of equations. The authors of the Beddoes–Leishman model even hint that the model they developed was difficult to extend without amplifying noise (Leishman, 1988). Essentially it becomes too difficult for a human creator to write down a complex enough model that is well behaved over all operating regimes. This is a recognized problem in machine learning, that models with enough capacity to learn a complex system tend to memorize the training data and perform poorly on new examples. This is called over-fitting.

Machine learning provides another path to improving aerodynamic models, as it provides the tools and techniques to fit high-capacity models while simultaneously handling the problem of over-fitting. Such an approach would perform much the same role as the current models but would be machine learned. It is important to note here that neural networks are not a look-up table. In the same way that our convolutional neural network learned more complex features as we moved deeper into the network, a neural network would begin to learn abstractions that are useful in the context of unsteady aerodynamics, i.e., relationships between angle of attack and lift.

Of course, the network has to be trained to learn these abstractions. Using the concept of transfer learning it would be possible to train the model in stages. We outline a potential recipe for creating such a new model with the disclaimer that this is speculative, and we fully expect some of the stages to require modification. Nonetheless, the recipe discusses the principle of training in stages, first with larger quantities of computationally cheap data and then again with smaller quantities of higher-quality data. The machine-learned model training process could be achieved with the following steps:

1. Generate a huge set of “cheap” training data using a standard unsteady aerodynamic model.
2. Train the machine learning model on these data until it performs as well as the standard model.
3. Generate unsteady CFD and experimental training data for a single airfoil.
4. Use the smaller amount of higher-fidelity data to further train the machine learning model.
5. For each airfoil, generate a small amount of CFD data.
6. Recalibrate the machine-learned model to each airfoil.

This approach has the advantage that the model can be constrained with a nearly endless supply of cheap data from the standard unsteady aerodynamics models. We would now have confidence that over nearly all operating conditions the model would not diverge too far from reality. In this first stage, we have trained the network to learn a useful set of abstractions that apply to unsteady aerodynamics. The model can then be remolded just enough to represent the higher-fidelity data from experiments and CFD without losing the constraints set in the previous step. This would produce a base model. For each new airfoil a new sub-model could be spawned off with a small amount of training data and computational effort. This means we have the robustness of the engineering model with an improved ability to match high-quality data.

This concept does come with some challenges. The current low-fidelity unsteady aerodynamics models are not designed to produce results at very high angles of attack. Furthermore, at very high angles of attack it is usually required to use 3D CFD to get high-quality results. Finally, the shedding modes are affected by the flexibility of the structure, that is to say the full 3D structure. A possible approach is to use very rough approximations for the cheap training data (just based on the Strouhal number) followed by 2D CFD. While these two approaches are unlikely to be accurate, it will pre-train the model to reproduce the rough physics. This would then reduce the amount of 3D CFD with structural interaction that would be required to represent the very high angles of attack. This approach would treat the final version of the model as a blade dynamic stall model rather than an airfoil. These simulations would still require large amounts of computational power given current standards but will be the cheaper (if not cheap) approach. While the method described here does not provide a final approach, it hopefully demonstrates a useful machine learning principle of refining models in stages to make the best use of the data available.

[8] Holierhoek et al. (2013) have a good comparison of the models.

## 7 Conclusions

This paper has attempted to bridge the gap between unsteady aerodynamics and the field of data science and machine learning. In particular we have attempted to provide some use cases of machine learning in unsteady aerodynamics. Stall is a complex phenomenon which varies in both time and space, and the data have shown strong variations between cycles of the experiments. The combination of clustering, dynamic time warping and multidimensional scaling allows us to effectively cluster cycles together, making the data easy to interpret and reveal patterns that were previously difficult to inspect visually. Convolutional neural networks allow us to use computer vision on pressure data to find dynamic stall vortex convection. Using neural networks to extract complex features from data has an incredible potential within aerodynamics, especially due to the advent of transfer learning.

Even the few examples analyzed in this study demonstrate that stall behavior is complex. The clustering results demonstrated that the shedding behavior varies across cycles, especially in the secondary and tertiary vorticities. The neural network was able to extract the vortex convection feature from the pressure plots to show that the onset of dynamic stall and the convection speed vary with the inflow conditions as well as cycle to cycle. The approaches described in this paper are just examples of the potential approaches that can be used to provide detailed insights into unsteady aerodynamics data.

The results of this study already provide a number of recommendations about stall and data science.

1. Means are not a sufficient description of stall. Data science and machine learning provide good ways of investigating cycle-to-cycle variations.

2. Multidimensional scaling and clustering with DTW as a distance metric is an effective way of examining data for different shedding modes or experimental outliers.

3. Dynamic stall behaviors vary significantly even within the same test conditions.

4. It is unlikely the traditional empirical models are the solution to modeling stall more accurately, and machine learning may be the better option.

5. Dynamic stall vortices will convect at different times and with different speeds. A neural network can retrieve this information from pressure data with a reasonable amount of training data and computational resources.

6. The bootstrapping method will help with determining the number of cycles needed to reach a given level of confidence.

7. The examples in this paper did not require huge datasets (though they can be used on larger datasets) or large computational resources, nor did they require significant amounts of specialized knowledge.

Finally, we hope that the demonstrations provided in this paper will communicate that there is a rich family of machine learning methods available for use in wind energy, unsteady aerodynamics and other adjacent fields.

**Code availability.** A distillation of the codes used in this paper is available at https://doi.org/10.5281/zenodo.3909028 (Lennie and Steenbuck, 2020). The data used for the convection plots are also in the repository. An example file is provided for the time series clustering with the MDS plot.

**Author contributions.** ML prepared the manuscript with the help of all co-authors. The computer vision model was constructed by ML. JS constructed the clustering code with assistance and supervision from ML. BRN provided code review and technical advice. COP provided assistance with the paper review.

**Competing interests.** The authors declare that they have no conflict of interest.

**Special issue statement.** This article is part of the special issue “Wind Energy Science Conference 2019”. It is a result of the Wind Energy Science Conference 2019, Cork, Ireland, 17–20 June 2019.

**Acknowledgements.** The wind tunnel data were taken by Hanns-Mueller Vahl and David Greenblatt at the Technion – Israel Institute of Technology in conjunction with the Technische Universität Berlin. The towing-tank data were measured by Marvin Jentzsch and Hajo Schmidt at the Technische Universität Berlin. The time series microphone data used in the recurrent neural network example were provided by Helge Madsen under the DanAero project. The DanAero projects were funded partly by the Danish Energy Authorities (EFP2007. Journal no. 33033-0074 and EUDP 2009-II. Journal no. 64009-0258) and partly by the project partners Vestas, Siemens, LM, Dong Energy and DTU. These datasets required extensive measurement effort and ongoing consulting to make the data useful; for this the authors acknowledge and thank the contributors. The authors would also like to thank Kenneth Granlund and George Pechlivanoglou for providing interesting feedback. The researchers would also like to acknowledge the support received from NVIDIA through their GPU grant program.

**Financial support.** We acknowledge support by the German Research Foundation and the Open Access Publication Fund of TU Berlin.

**Review statement.** This paper was edited by Katherine Dykes and reviewed by Tuhfe Gocmen and two anonymous referees.

## References

Abbott, I. H. and Doenhoff, A. E. V.: Theory of Wing Sections, Including a Summary of Airfoil Data, 1st Edn., Dover Publications, Dover, 1959.

Andersen, P. B., Gaunaa, M., Bak, C., and Hansen, M. H.: A Dynamic Stall Model for Airfoils with Deformable Trailing Edges, J. Phys.: Conf. Ser., 75, 012028, https://doi.org/10.1088/1742-6596/75/1/012028, 2007.

Bak, C., Madsen, H. A., Fuglsang, P., and Rasmussen, F.: Double stall, in: vol. 1043, available at: http://orbit.dtu.dk/fedora/objects/orbit:90308/datastreams/file_7731788/content (last access: 13 September 2019), 1998.

Bak, C., Madsen, H. A., Paulsen, U. S., Gaunaa, M., Fuglsang, P., Romblad, J., Olesen, N. A., Enevoldsen, P., Laursen, J., and Jensen, L.: DAN-AERO MW: Detailed aerodynamic measurements on a full scale MW wind turbine, in: European Wind Energy Conference and Exhibition (EWEC), 20–23 April 2010, Warsaw, Poland, 1–10, 2010.

Balduzzi, F., Bianchini, A., Church, B., Wegner, F., Ferrari, L., Ferrara, G., and Paschereit, C. O.: Static and Dynamic Analysis of a NACA 0021 Airfoil Section at Low Reynolds Numbers Based on Experiments and Computational Fluid Dynamics, J. Eng. Gas Turb. Power, 141, 1–10, https://doi.org/10.1115/1.4041150, 2019.

Bowles, P. O., Corke, T. C., Coleman, D. ., Thomas, F. O., and Wasikowski, M.: Improved Understanding of Aerodynamic Damping Through the Hilbert Transform, AIAA J., 52, 2384–2394, https://doi.org/10.2514/1.J052630, 2014.

Brownlee, J.: A Gentle Introduction to Transfer Learning for Deep Learning, available at: https://machinelearningmastery.com/transfer-learning-for-deep-learning/ (last access: 13 September 2019), 2017.

Brunton, S. L. and Noack, B. R.: Closed-Loop Turbulence Control: Progress and Challenges, Appl. Mech. Rev, 67, 050801, https://doi.org/10.1115/1.4031175, 2015.

Brunton, S. L., Proctor, J. L., Tu, J. H., and Kutz, J. N.: Compressed sensing and dynamic mode decomposition, J. Comput. Dynam., 2, 165–191, https://doi.org/10.3934/jcd.2015002, 2015.

Cao, Y., Kaiser, E., Borée, J., Noack, B. R., Thomas, L., and Guilain, S.: Cluster-based analysis of cycle-to-cycle variations: application to internal combustion engines, Exp. Fluids, 55, 1837, https://doi.org/10.1007/s00348-014-1837-y, 2014.

Carr, L. W.: Progress in Analysis and Prediction of Dynamic Stall, J. Aircraft, 25, 6–17, https://doi.org/10.2514/3.45534, 1987.

Chernick, M. R.: Bootstrap Methods: A Guide for Practitioners and Researchers, in: Wiley Series in Probability and Statistics, Wiley, Newtown, PA, 2008.

Cuturi, M. and Blondel, M.: Soft-DTW: a Differentiable Loss Function for Time-Series, available at: http://arxiv.org/abs/1703.01541 (last access: 13 September 2019), 2017.

Dask Development Team: Dask: Library for dynamic task scheduling, available at: https://dask.org (last access: 13 September 2019), 2016.

Ehlert, A., Nayeri, C. N., Morzynski, M., and Noack, B. R.: Locally linear embedding for transient cylinder wakes, preprint: http://arxiv.org/abs/1906.07822, last access: 13 September 2019.

Gaunaa, M., Heinz, J., and Skrzypiński, W.: Toward an Engineering Model for the Aerodynamic Forces Acting on Wind Turbine Blades in Quasisteady Standstill and Blade Installation Situations, J. Phys.: Conf. Ser., 753, 022007, https://doi.org/10.1088/1742-6596/753/2/022007, 2016.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep Learning, 1st Edn., MIT Press, Cambridge, USA, 2016.

Granlund, K., Monnier, B., Ol, M., and Williams, D.: Airfoil longitudinal gust response in separated vs. attached flows, Phys. Fluids, 26, 027103, https://doi.org/10.1063/1.4864338, 2014.

Greenblatt, D.: Unsteady Low-Speed Wind Tunnels, AIAA J., 54, 1817–1830, https://doi.org/10.2514/1.J054590, 2016.

Greenblatt, D. and Wygnanski, I.: Effect of Leading-Edge Curvature and Slot Geometry on Dynamic Stall Control, in: 1st Flow Control Conference, 24–26 June 2002, St. Louis, Missouri, https://doi.org/10.2514/6.2002-3271, 2002.

He, K. and Sun, J.: Deep Residual Learning for Image Recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las 27–30 June 2016, Vegas, NV, USA, 1–9, https://doi.org/10.1109/cvpr.2016.90, 2016.

Holierhoek, J., de Vaal, J., van Zuijlen, A., and Bijl, H.: Comparing different dynamic stall models, Wind Energy, 16, 139–158, https://doi.org/10.1002/we.548, 2013.

Holst, D., Church, B., Wegner, F., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O.: Experimental Analysis of a NACA 0021 Airfoil Under Dynamic Angle of Attack Variation and Low Reynolds Numbers, J. Eng. Gas Turb. Power, 141, 1–10, https://doi.org/10.1115/1.4041146, 2019.

Hosseini, Z., Martinuzzi, R. J., and Noack, B. R.: Modal energy flow analysis of a highly modulated wake behind a wall-mounted pyramid, J. Fluid Mech., 798, 717–750, https://doi.org/10.1017/jfm.2016.345, 2016.

Howard, J. and others: FASTAI, available at: https://github.com/fastai/fastai, last access: 13 September 2019.

Hudy, L. M. and Naguib, A.: Stochastic estimation of a separated-flow field using wall-pressure-array measurements, Phys. Fluids, 19, 1–18, https://doi.org/10.1063/1.2472507, 2007.

Jentzsch, M. P., Schmidt, H.-J., Woszidlo, R., Nayeri, C., and Paschereit, C. O.: Development of a Setup and Measurement Procedure for Unsteady Model Velocities in a Large Water Towing Tank, in: AIAA Scitech 2019 Forum, San Diego, California, https://doi.org/10.2514/6.2019-2164, 2019.

Jones, E., Oliphant, T., Peterson, P., and Others: {SciPy}: Open source scientific tools for {Python}, availiable at: http://www.scipy.org/, last access: 13 September 2019.

Kirk, T. M. and Yarusevych, S.: Vortex shedding within laminar separation bubbles forming over an airfoil, Exp. Fluids, 58, 43, https://doi.org/10.1007/s00348-017-2308-z, 2017.

Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, in: Advances in Neural Information Processing Systems 25, edited by: Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., Curran Associates, Inc., 1097–1105, available at: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural (last access: 13 September 2019), 2012.

Kulkarni, P. A., Dhoble, A. S., and Padole, P. M.: Deep neural network-based wind speed forecasting and fatigue analysis of a large composite wind turbine blade, Proc. Inst. Mech. Eng. Pt. C, 233, 2794–2812, https://doi.org/10.1177/0954406218797972, 2019.

Kutz, J. N., Fu, X., and Brunton, S. L.: Multi-Resolution Dynamic Mode Decomposition, J. Appl. Dynam. Syst., 15, 713–735, 2015.

Leishman, J. G.: Validation of approximate indicial aerodynamic functions for two-dimensional subsonic flow, J. Aircraft, 25, 914–922, https://doi.org/10.2514/3.45680, 1988.

Leishman, J. G.: Challenges in Modeling the Unsteady Aerodynamics of Wind Turbines, in: 21st ASME Wind Energy Symposium and the 40th AIAA Aerospace Sciences Meeting, Reno, NV, 2002.

Leishman, J. G.: Principles of helicopter aerodynamics, 2nd Edn., Cambridge University Press, Cambridge, 2006.

Lennie, M. and Steenbuck, J.: VortexCNN, Zenodo, https://doi.org/10.5281/zenodo.3909028, 2020.

Lennie, M., Bach, A., Pechlivanoglou, G., Nayeri, C., and Paschereit, C. O.: The Unsteady Aerodynamic Response of an Airfoil with Microtabs and it's Implications for Aerodynamic Damping, in: 34th Wind Energy Symposium, 4–8 January 2016, San Diego, California, USA, 1–12, https://doi.org/10.2514/6.2016-1006, 2016.

Lennie, M., Wendler, J., Pechlivanoglou, G., Nayeri, C., Paschereit, C. O., and Greenblatt, D.: Development of a Partially Stochastic Unsteady Aerodynamics Model, in: 35th Wind Energy Symposium, 9–13 January 2017, Grapevine, Texas, 1–14, https://doi.org/10.2514/6.2017-2002, 2017.

Lennie, M., Marten, D., Pechlivanoglou, G., Paschereit, C. O., and Dominin, S.: Simulating Wind Turbine Ice Throw: QBlade and Statistical Analysis, in: Proceedings of the ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition, Vol. 9: Oil and Gas Applications, Supercritical $CO_2$ Power Cycles, Wind Energy, Oslo, Norway, https://doi.org/10.1115/gt2018-76485, 2018a.

Lennie, M., Selahi-moghaddam, A., Holst, D., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O.: GTP-17-1616 Vortex shedding and Frequency lock in on stand still wind turbines, a baseline experiment, J. Eng. Gas Turb. Power, 140, 112603, https://doi.org/10.1115/1.4039818, 2018b.

Loiseau, J. C., Noack, B. R., and Brunton, S. L.: Sparse reduced-order modelling: Sensor-based dynamics to full-state estimation, J. Fluid Mech., 844, 459–490, https://doi.org/10.1017/jfm.2018.147, 2018.

Madsen, H. A., Özçakmak, Ö. S., Bak, C., Troldborg, N., Sørensen, N. N., and Sørensen, J. N.: Transition characteristics measured on a 2 MW 80 m diameter wind turbine rotor in comparison with transition data from wind tunnel measurements, in: AIAA Scitech 2019 Forum, 7–11 January 2019, San Diego, California, https://doi.org/10.2514/6.2019-0801, 2019.

Maimon, O. and Rokach, L.: Data Mining and Knowledge Discovery Handbook, Springer, Boston, MA, 2006.

Manolesos, M.: Testing airfoils for wind turbines The 3D challenges, in: 10th EAWE PhD Seminar on Wind Energy in Europe, National Technical University of Athens – Fluids Section, Orleans, France, 2014.

Manolesos, M., Papadakis, G., and Voutsinas, S. G.: An experimental and numerical investigation on the formation of stall-cells on airfoils, J. Phys.: Conf. Ser., 555, 012068, https://doi.org/10.1088/1742-6596/555/1/012068, 2014.

McAlister, K. W., Carr, L. W., and McCroskey, W.: Dynamic stall experiments on the NACA 0012 Airfoil, Tech. rep., NASA Aeromechanics Laboratory Ames Research Center, Moffett Field, California, https://doi.org/10.1007/BF00575335, 1978.

McCroskey, W. J.: The Phenomenon of Dynamic Stall, Tech. rep., Ames Research Center, NASA, https://doi.org/10.1080/6008555886, 1981.

McCroskey, W. J.: Unsteady Airfoils, Annu. Rev. Fluid Mech., 14, 285–311, https://doi.org/10.1146/annurev.fl.14.010182.001441, 1982.

Medina, A., Ol, M. V., Greenblatt, D., Müller-Vahl, H., and Strangfeld, C.: High-Amplitude Surge of a Pitching Airfoil: Complementary Wind- and Water-Tunnel Measurements, AIAA J., 56, 1–7, https://doi.org/10.2514/1.J056408, 2018.

Morel, M., Achard, C., Kulpa, R., and Dubuisson, S.: Time-series averaging using constrained dynamic time warping with tolerance, Pattern Recog., 74, 77–89, https://doi.org/10.1016/j.patcog.2017.08.015, 2018.

Mulleners, K. and Raffel, M.: The onset of dynamic stall revisited, Exp. Fluids, 52, 779–793, https://doi.org/10.1007/s00348-011-1118-y, 2012.

Mulleners, K. and Raffel, M.: Dynamic stall development, Exp. Fluids, 54, 1469, https://doi.org/10.1007/s00348-013-1469-7, 2013.

Mulleners, K. and Rütten, M.: Analysis of intermittent trailing edge vortex shedding using recurrence plots, AIAA J., 56, 571–580, 2018.

Mulleners, K., Pape, A. L., Heine, B., and Raffel, M.: The Dynamics of Static Stall, in: vol. 6, 16th International Symposium on Applications of Laser Techniques to Fluid Mechanics, Lisbon, Portugal, 9–12, 2012.

Müller-Vahl, H. F.: Wind Turbine Blade Dynamic Stall and its Control, Doctoral thesis, Technische Universität Berlin, Berlin, 2015.

Müller-Vahl, H. F., Strangfeld, C., Nayeri, C. N., and Paschereit, C. O.: Control of Thick Airfoil, Deep Dynamic Stall Using Steady Blowing, AIAA J., 53, 1–34, https://doi.org/10.2514/1.J053090, 2015.

Müller-Vahl, H. F., Pechlivanoglou, G., Nayeri, C. N., Paschereit, C. O., and Greenblatt, D.: Matched pitch rate extensions to dynamic stall on rotor blades, Renew. Energy, 105, 505–519, https://doi.org/10.1016/j.renene.2016.12.070, 2017.

Nair, A. G., Yeh, C.-A., Kaiser, E., Noack, B. R., Brunton, S. L., and Taira, K.: Cluster-based feedback control of turbulent post-stall separated flows, J. Fluid Mech., 875, 345–375, https://doi.org/10.1017/jfm.2019.469, 2019.

O'Connell, A. A., Borg, I., and Groenen, P.: Modern Multidimensional Scaling: Theory and Applications, in: vol. 94 of Springer Series in Statistics, Springer, New York, https://doi.org/10.2307/2669710, 1999.

Olah, C.: Visualizing MNIST: An Exploration of Dimensionality Reduction, available at: https://colah.github.io/posts/2014-10-Visualizing-MNIST/, last access: 13 September 2019.

Pires, O., Munduate, X., Boorsma, K., Ceyhan Yilmaz, O., Aa Madsen, H., and Timmer, W. A.: Experimental investigation of Surface Roughness effects and Transition on Wind Turbine performance, in: vol. 1037, The Science of Making Torque from Wind (TORQUE 2018), Milan, https://doi.org/10.1088/1742-6596/1037/5/052018, 2018.

Raschka, S.: Python Machine Learning, 1st Edn., Packt Publishing, 2015.

Ratanamahatana, C. and Keogh, E.: Everything you know about dynamic time warping is wrong, in: Third Workshop on Mining Temporal and Sequential Data, Citeseer, 22–25, https://doi.org/10.1097/01.CCM.0000279204.24648.44, 2004.

Riches, G., Martinuzzi, R., and Morton, C.: Proper orthogonal decomposition analysis of a circular cylinder undergoing vortex-induced vibrations, Phys. Fluids, 30, 105103, https://doi.org/10.1063/1.5046090, 2018.

Rumsey, C. L.: Successes and Challenges for Flow Control Simulations (Invited), in: 4th Flow Control Conference, 1–26, https://doi.org/10.2514/6.2008-4311, 2008.

Rumsey, C. L. and Nishino, T.: Numerical study comparing RANS and LES approaches on a circulation control airfoil, Int. J. Heat Fluid Flow, 32, 847–864, https://doi.org/10.1016/j.ijheatfluidflow.2011.06.011, 2011.

Schlichting, H. and Gersten, K.: Boundary-Layer Theory, 9th Edn., Springer, Berlin, Heidelberg, 2016.

Schmid, P.: Dynamic mode decomposition of numerical and experimental data, J. Fluid Mech., 656, 5–28, https://doi.org/10.1017/S0022112010001217, 2010.

Shihavuddin, A. S., Chen, X., Fedorov, V., Christensen, A. N., Riis, N. A. B., Branner, K., Dahl, A. B., and Paulsen, R. R.: Wind turbine surface damage detection by deep learning aided drone inspection analysis, Energies, 12, 1–15, https://doi.org/10.3390/en12040676, 2019.

Sieber, M., Oberleithner, K., and Paschereit, C. O.: Spectral proper orthogonal decomposition, J. Fluid Mech., 792, 798–828, https://doi.org/10.1017/jfm.2016.103, 2015.

Skrzypinski, W. R.: Analysis and modeling of unsteady aerodynamics with application to wind turbine blade vibration at standstill conditions, PhD thesis, DTU, Copenhagen, 2012.

Skrzypinski, W. R., Gaunaa, M., Sørensen, N., Zahle, F., and Heinz, J.: Vortex-induced vibrations of a DU96-W-180 airfoil at 90° angle of attack, Wind Energy, 17, 1495–1514, https://doi.org/10.1002/we.1647, 2014.

Steenbuck, J.: Machine Learning approach to identify dynamic stall variability, MS thesis, Technische Universität Berlin, Berlin, 2019.

Strangfeld, C., Rumsey, C. L., Müller-Vahl, H., Greenblatt, D., Nayeri, C. N., and Paschereit, C. O.: Unsteady thick airfoil aerodynamics: Experiments, computation, and theory, in: 45th AIAA Fluid Dynamics Conference, 22–26 June 2015, Dallas, TX, 1–19, https://doi.org/10.2514/6.2015-3071, 2015.

Taira, K., Brunton, S. L., Dawson, S. T. M., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V., and Ukeiley, L. S.: Modal Analysis of Fluid Flows: An Overview, AIAA J., 55, 1–46, 2017.

Tavenard, R.: tslearn, available at: https://tslearn.readthedocs.io/en/latest/index.html (last access: 13 September 2019), 2008.

Vey, S., Lang, H. M., Nayeri, C. N., Paschereit, C. O., and Pechlivanoglou, G.: Extracting quantitative data from tuft flow visualizations on utility scale wind turbines, J. Phys.: Conf. Ser., 524, 012011, https://doi.org/10.1088/1742-6596/524/1/012011, 2014.

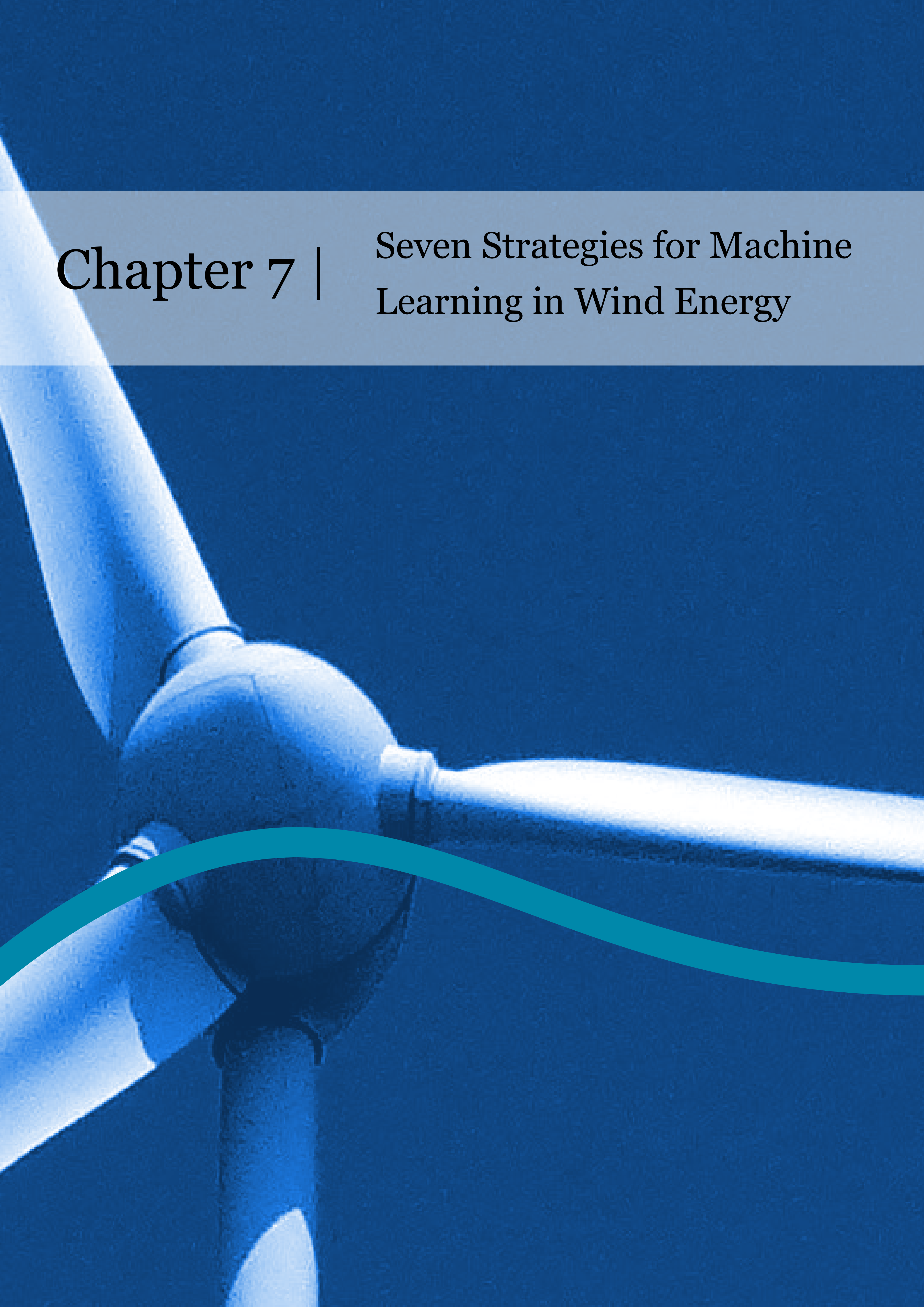
Ward, J. W.: The Behaviour and Effects of Laminar Separation Bubbles on Aerofoils in Incompressible Flow, Aeronaut. J., 67, 783–790, https://doi.org/10.1017/S0001924000061583, 1963.

Wendler, J., Marten, D., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O.: GT2016-57184 An Unsteady Aerodynamics Model for Lifting Line Free Vortex Wake Simulations of HAWT and VAWT in QBlade, in: Proceedings of ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition, Seoul, South Korea, 2016.

Yuan, B., Wang, C., Jiang, F., Long, M., Yu, P. S., and Liu, Y.: WaveletFCNN: A Deep Time Series Classification Model for Wind Turbine Blade Icing Detection, CoRR, abs/1902.0, http://arxiv.org/abs/1902.05625, 2019.

Zeiler, M. D. and Fergus, R.: Visualizing and Understanding Convolutional Networks, CoRR, abs/1311.2, http://arxiv.org/abs/1311.2901, 2013.

# Chapter 7 | Seven Strategies for Machine Learning in Wind Energy



# Chapter 7 Seven strategies for machine learning in wind energy

In the introduction to this thesis, we saw that wind energy is in need of:

1. Improved simulation tools to match the expanding needs of wind energy;
2. Better engineering models and analysis methods that meet modern challenges; and
3. Better utilisation of machine learning to help mine and model data.

Good examples of all three ideas have been laid down. The FVLLT model proved to be a good simulation method for floating-platform wind turbines - an otherwise challenging case for previous models. The derivation of the instantaneous aerodynamic rotor damping was a perfect example of how a traditional derivation approach can still yield powerful, closed-form engineering models. Conversely, airfoil deep stall modelling was beyond the traditional engineering approaches, and a data-driven machine-learning approach proved to be more powerful. The key achievements can be found in Figure 7.2. The outcomes of this study will impact load case simulation most heavily especially in previously difficult simulations domains such as stand still vibrations or floating platform wind turbine simulations. However, I believe this study should serve as a strong motivation for machine learning and data science in wind energy.

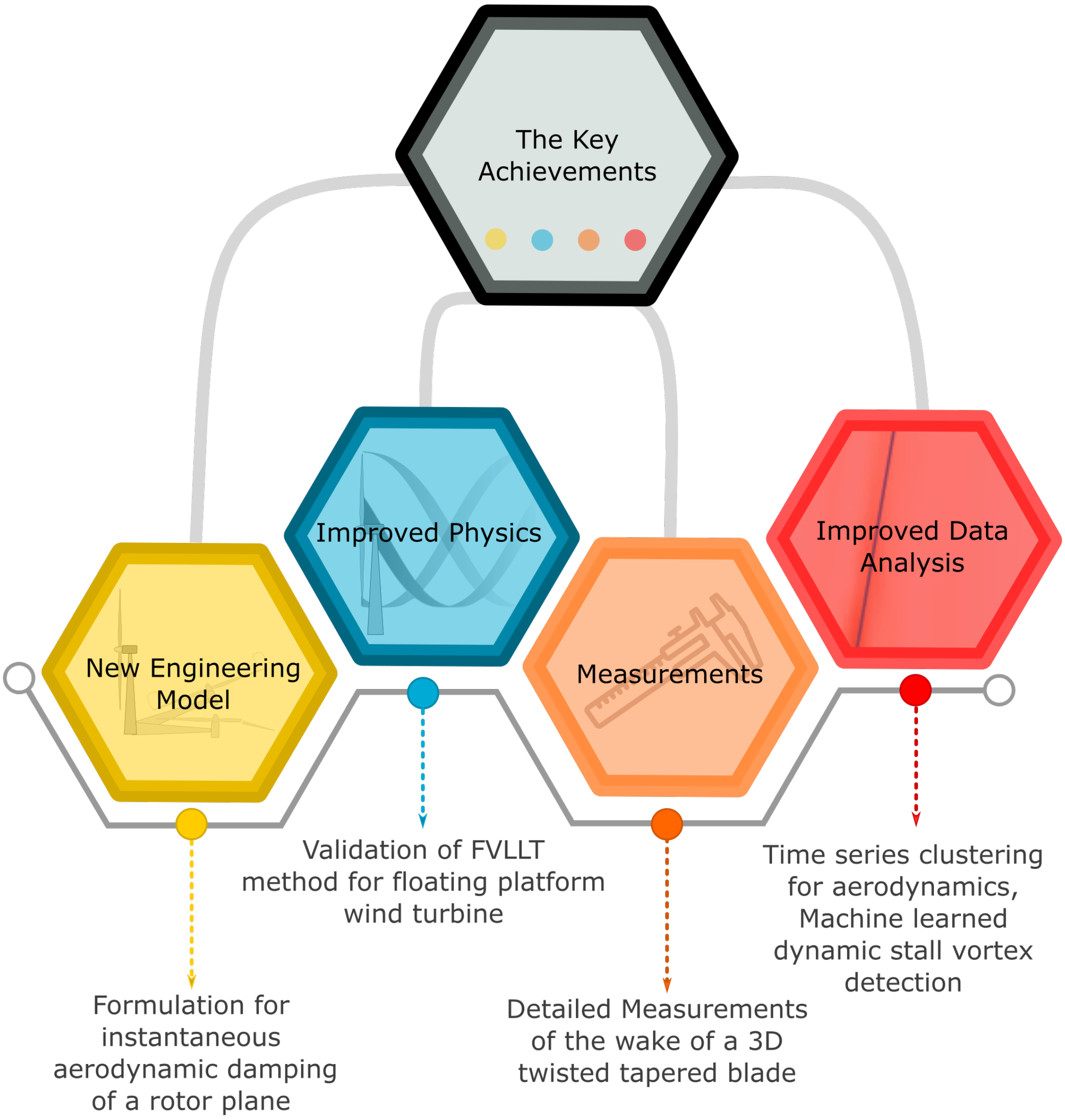

The Key Achievements
New Engineering Model
Improved Physics
Measurements
Improved Data Analysis
Validation of FVLLT method for floating platform wind turbine
Time series clustering for aerodynamics, Machine learned dynamic stall vortex detection
Formulation for instantaneous aerodynamic damping of a rotor plane
Detailed Measurements of the wake of a 3D twisted tapered blade

**Figure 7.1:** The key achievements from the three journal papers

Machine learning and data science have great potential in the context of wind energy, but there are road-blocks standing in the way. The largest data source in wind energy comes from SCADA systems, and the data is not high quality, meaning that it presents a challenging and labour-intensive data science problem - not ideal to enthuse an industry about the power of these methods. Engineering models, such as BEM, have a long history and extensive testing. BEM does not work well in a lot of situations, but the designers are comfortable with this because it is easy to understand what is wrong when it does not work, and why. Finally, BEM is incredibly computationally cheap. It is hard to persuade someone to replace the model they know well with a black-box.

However, this is simply a failure in marketing. In my opinion, machine learning will change wind energy not by creating an omniscient AI that can design a wind turbine without any human input. Data science and machine learning will add value to wind energy through the following seven strategies;

1. Low fidelity models and transfer learning;
2. Feature engineering and data mining;
3. Machine learning to move between scales;
4. Sparse sensing;
5. Efficiency through convergence criteria;
6. Machine learn a small part of the system;
7. Machine learn the whole task.

Note here that only the last strategy is to machine-learn a whole task. Machine learning will help stepwise, by replacing an assumption with a model or, an expensive computation with a cheaper, machine-learned, version, by helping us understand our data or by simply reducing the number of sensors needed to measure the state of a machine.

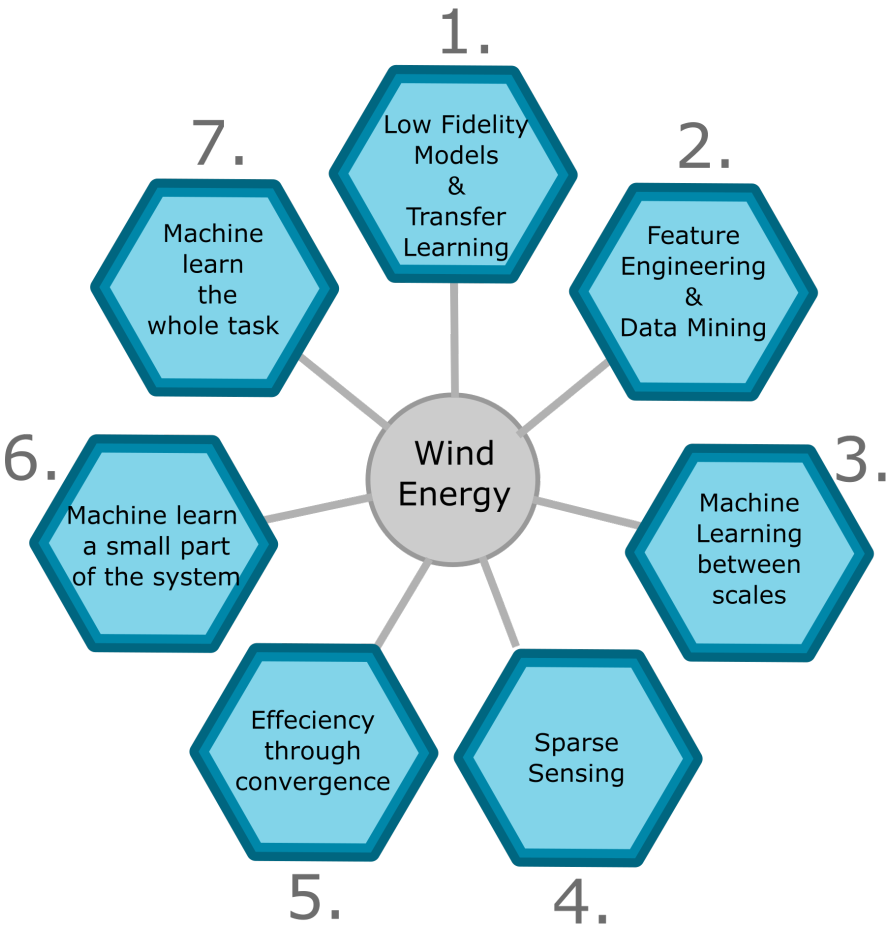

1.
Low Fidelity Models & Transfer Learning
2.
Feature Engineering & Data Mining
3.
Machine Learning between scales
4.
Sparse Sensing
5.
Effeciency through convergence
6.
Machine learn a small part of the system
7.
Machine learn the whole task
Wind Energy

**Figure 7.2:** Seven Strategies for Machine Learning in Wind Energy

In the following section, I describe these seven strategies for using data science and machine learning in the context of wind energy. These strategies derive from lessons learned from the studies in the three papers presented. For some of the strategies, there are already examples of the wind industry embracing these approaches. The important point is that machine-learning approaches do not discard past developments, but rather build on them. These seven strategies provide a road map for integrating engineering models and data science into wind energy.

## 7.1 Strategy 1 - Low fidelity models and transfer learning

Transfer learning is probably the single most important concept that needs to penetrate the minds of people not involved in machine learning. Transfer learning can be defined as:

**Definition 6.** *"Transfer learning is the improvement of learning in a new task through the transfer of knowledge from a related task that has already been learned." [60]*

The implications of this concept are deeper than they first appear. The key is, what is a related task? It turns out that, especially for neural networks, many datasets are very closely related.

Modern convolutional neural network architectures, such as the ResNet models [79], have a vast amount of flexibility. With such architecture, it's possible to build a classifier that will correctly label (with unprecedented accuracy) most common objects, such as trees, cars, dogs, a chessboard or a computer. The way we represent pictures is useful for projecting onto a computer screen, but the method inefficient. Almost every possible value combination of a photograph will be noise, and only an extremely small subset of pixel combinations will give a picture that we would expect to find in nature. This means that real geometries are much closer to each other than non-real pixel combinations. That is to say, a picture of a dog and a salt shaker are much closer to each other than a picture of a dog and the noise of a disconnected television set [200][19]. The important point is that, whatever pictures we can generate in science, (i.e. a pressure plot versus time) will be much closer to the picture of the dog and the salt shaker than to the picture of noise. Our scientific plots sit in the same subset of possible geometries as all pictures in nature. Once we understand this, we can see that 'related' applications has a broad definition. There was an example of this in the computer vision model built to extract dynamic stall vortex convection speeds - a model simply remolded from its pretrained weights.

Let us shift to an example applicable to wind energy, such as building a working replacement of the unsteady aerodynamics models. The model should generalisable across airfoils, maintain the reliability of the current models, but provide improved performance in very deep stall. Transfer learning provides a road map for undertaking this task (Figure 7.3).

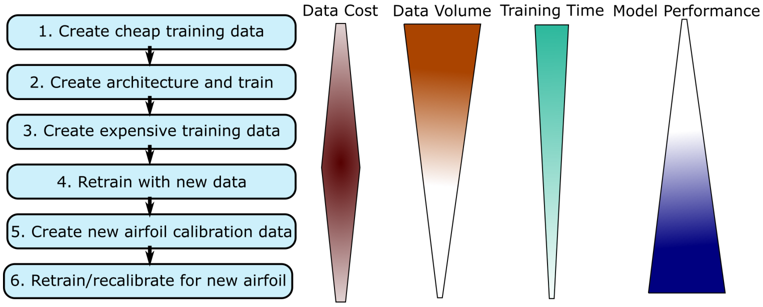

Data Cost
Data Volume
Training Time
Model Performance
1. Create cheap training data
2. Create architecture and train
3. Create expensive training data
4. Retrain with new data
5. Create new airfoil calibration data
6. Retrain/recalibrate for new airfoil

**Figure 7.3:** The transfer learning approach

We want to work with low-fidelity data in the early stages, because it is computationally cheap and little effort is required to produce large amounts of it. The Leishman-type models tend towards bias (i.e. being wrong, but stable), thus avoiding overfitting and numerical instabilities [78, 158, 73, 105]. Such models provide a rough sketch of what unsteady aerodynamics look like (see Figure 7.4). We can create very dense data for all of the input variables, even including

[19] Jeremy Howard details this point at length on the TWIML podcast [200]

some turbulence for data augmentation.

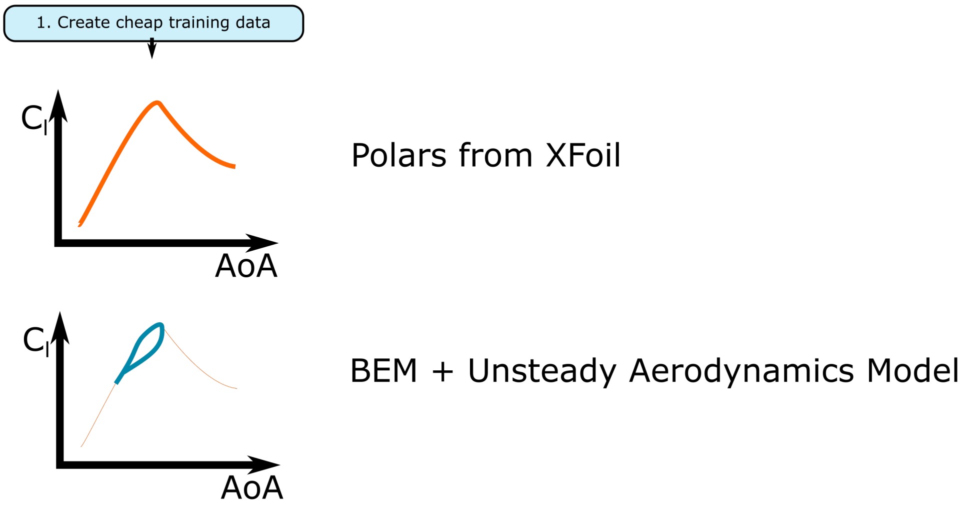

1. Create cheap training data
$C_l$
AoA
Polars from XFoil
$C_l$
AoA
BEM + Unsteady Aerodynamics Model

**Figure 7.4:** Low cost training data

We now have a cheap dataset that we can use to build up a model, but what kind of model? For time-series prediction with multiple inputs and as complicated complicated behaviour as unsteady aerodynamics, a recurrent neural network would be an appropriate choice. The first choice would be to take a pretrained model or architecture from a different application, such as econometrics, but if we need to generate our own architecture, there now exist efficient approaches to do this. Architecture generating algorithms exist in various forms. SethBling has demonstrated a nice implementation of such a process [179] (Figure 7.5). In this implementation, a neural network based agent learned to play Nintendo's Mario. In each generation, the architecture was trained, and then the best-performing architectures spawned children, and so on. Over many generations, the agent learned how to overcome each obstacle in the level.

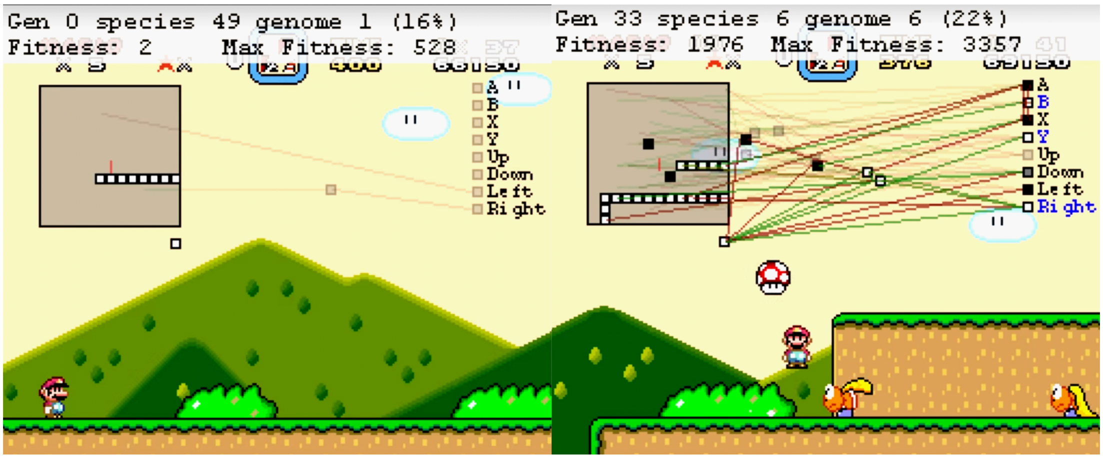

Gen 0 species 49 genome 1 (16%)
Fitness: 2 Max Fitness: 528
A
B
X
Y
Up
Down
Left
Right
Gen 33 species 6 genome 6 (22%)
Fitness: 1976 Max Fitness: 3357
A
B
X
Y
Up
Down
Left
Right

**Figure 7.5:** An example of evolved network architectures as described by Stanley and Miikkulainen [187] and implemented by SethBling [179]. The further Mario gets in the level after training, the better the architecture is ranked. (Mario is a trademark of Nintendo)

In our case, the same general approach can apply. There are many techniques, but they all concentrate on generating good neural network architectures. A useful review of the modern approaches can be found in Elsken et al. [60]. The good thing is that, we can do this data-hungry work using a cheap dataset to improve the efficiency of the project [60]. At the end of the process, we have a neural network architecture that represents the low fidelity training data (Figure 7.6).

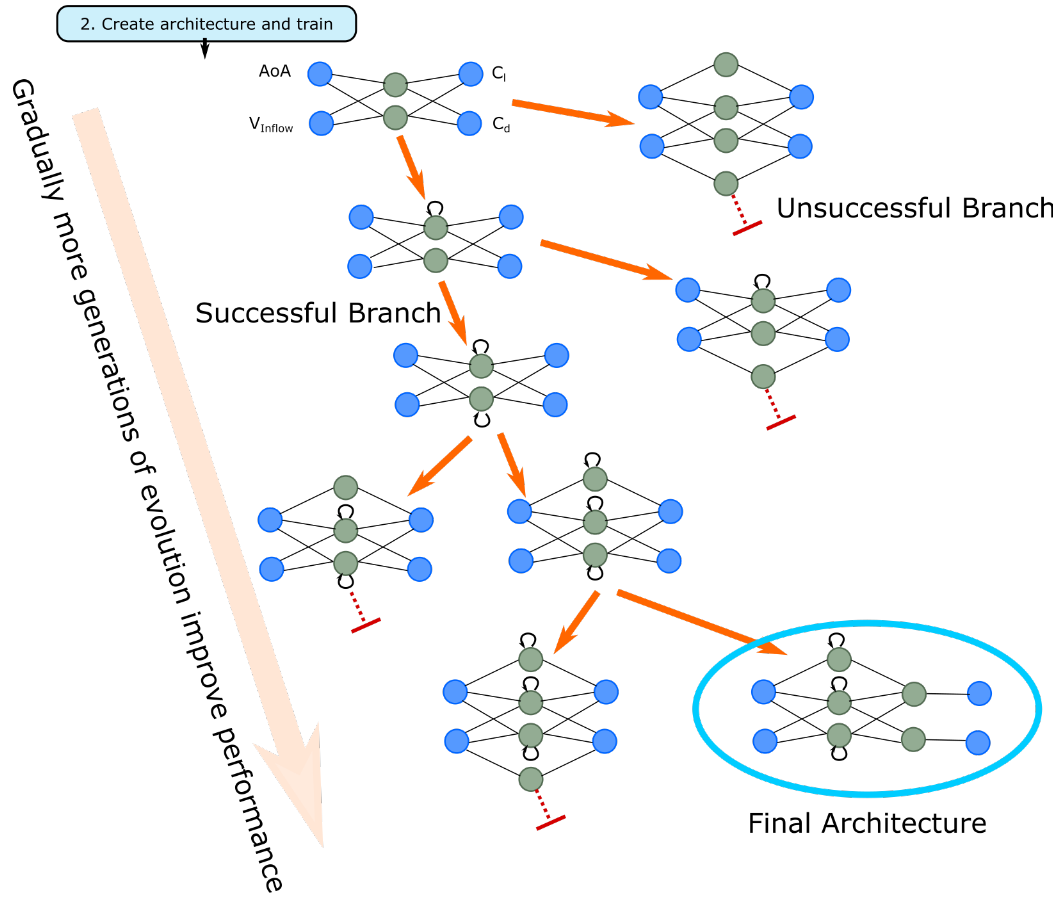

2. Create architecture and train
AoA
$V_{Inflow}$
$C_l$
$C_d$
Unsuccessful Branch
Successful Branch
Gradually more generations of evolution improve performance
Final Architecture

**Figure 7.6:** Designing the neural network architecture through evolution

At this stage, we need some higher fidelity data in order to improve the model further. Experiments and CFD both have their problems, but a combination of both these sources represents the best ground truth possible.

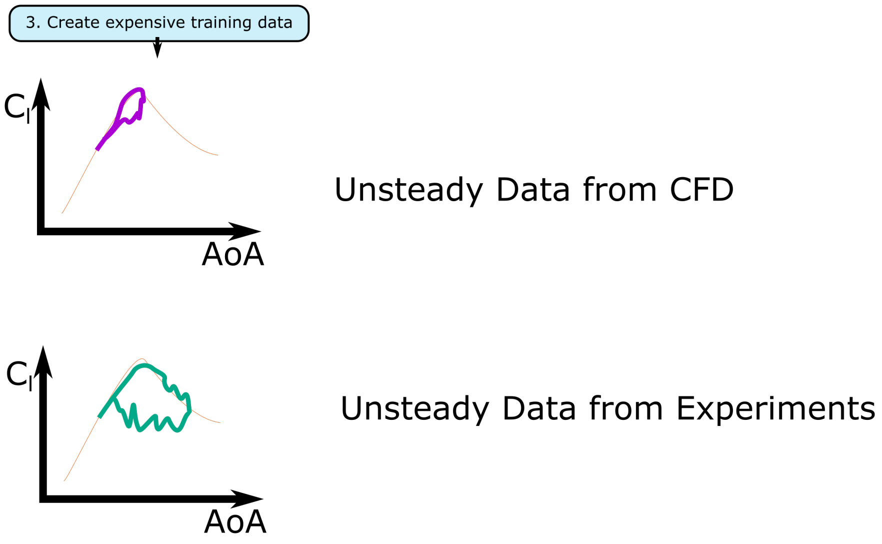

3. Create expensive training data
C_l
AoA
Unsteady Data from CFD
C_l
AoA
Unsteady Data from Experiments

**Figure 7.7:** High-fidelity data

Finally, we are able to leverage transfer learning. Out of all of the possibilities that can be represented by our neural network, the difference between the low-fidelity data and the high-fidelity data will be relatively modest. This means we can think of the high-fidelity data as adding the finishing touches on the model. It is critical in this case that this stage is performed with care. As the high-fidelity data is pushed through the neural network more times, the model will become more and more like the small amount of high quality data, however, if the training is too long, the neural network will begin to *only* represent the high-quality dataset and thus will generalise poorly. This means we would need to employ early stopping - the practice of stopping a model mid-training to prevent overfitting. In order to apply this model to different airfoils, we could leverage the same retraining procedure, as, the performances of airfoils are relatively similar.

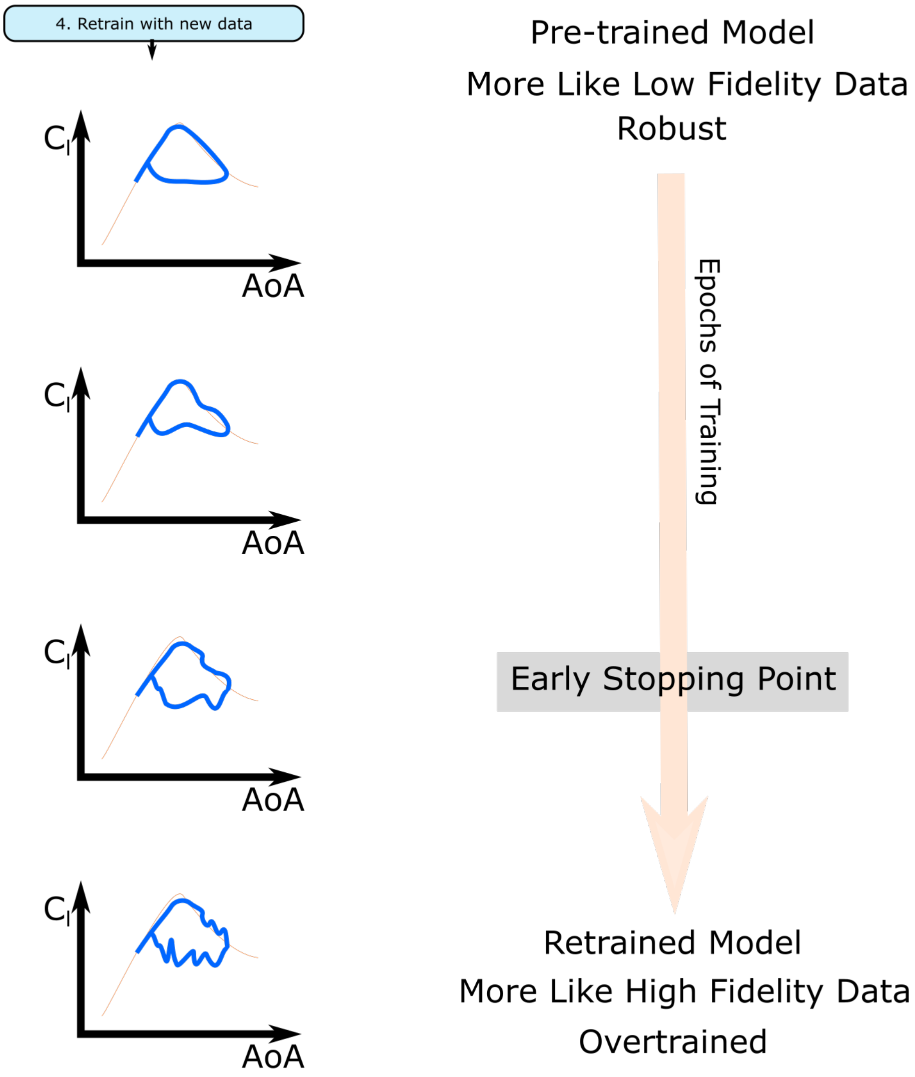


**Figure 7.8:** Retraining with high-fidelity data

Unsteady aerodynamics is just one application. If we think back to the introduction, there are multiple examples of low-order models that could be enhanced by high-fidelity data in this way. For example, we could use the Morrison equation for the first round of training in a floating-platform model, and expensive CFD or panel methods for the final tuning. Other possible applications might include controllers, inflow-measurement data models or structural monitoring models. This approach provides a smooth departure from engineering models into machine learning models.

## 7.2 Strategy 2 - Feature engineering and data mining

Feature engineering is the concept that data should be preprocessed before trying to predict on it [152]. It is the process of transforming data so that a learning algorithm can easily perform a task. Common examples include dimensionality reduction or clustering. In feature engineering, we are trying to make it easier for a model to make predictions. We can achieve this with dimensionality reduction techniques such as, principle component analysis (PCA) or linear discriminant analysis (LDA). Figure 7.9 shows how PCA can rotate axes into the directions of highest variance, thus making regression easier. Figure 7.9 also shows how LDA can rotate the axes into a direction that creates a maximum separation of classes, making classification easier. Feature engineering can be achieved by adding new features.

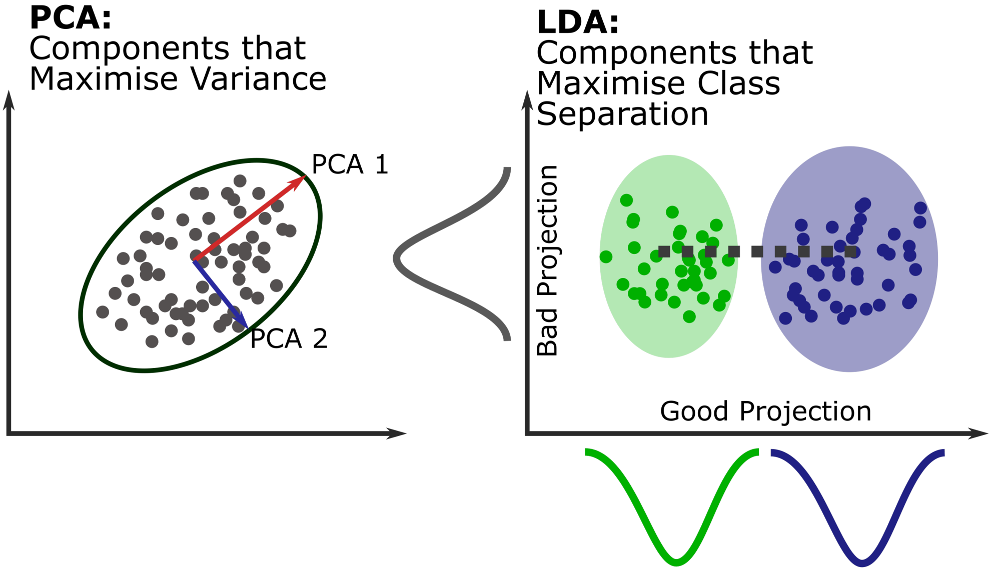


**Figure 7.9:** Using dimensionality reduction as feature engineering (adapted from Raschka [166])

Usually, the idea of machine learning is to let the model training process determine what to do with the features. However, as we retreat from influencing the process, we leave more things for the model to learn. In the unsteady aerodynamics example given in the last section, we forced the neural network to learn all physical behaviours such as attached flow lift, detachment of the boundary layer and system unsteadiness. We can afford to do this in this case because we have a very cheap model with which we can create endless amounts of data. However, sometimes, we may want to constrain the problem slightly to make the model easier to train with the data on hand.

What is a model constraint? A good example is in a simple logistic regression. A logistic regression is the same as a linear regression, but we push the result through a sigmoid function at the end to squash the signal between zero and one [17]. Having the sigmoid function constrains the output so the regression problem no longer has to learn first that the outputs will exist in a small sub-space of the domain before then learning the correct output. The constrained problem will require less data and less training time [20]. Prior knowledge about the problem to applies pre-constraints to the output.

We can also leverage engineering knowledge on the input features. In the following example, I would like to identify whether each microphone on an airfoil is experiencing a laminar or turbulent boundary layer. The input feature is a single time-series. We can see very clearly two regions, the areas with strong high-frequency content are turbulent. The key is that, our engineering knowledge tells us that high frequency content means turbulent.

If we put a single time-series into the model - a recurrent neural network in this case - the neural network has to first learn that high frequencies are different from low frequencies, and then that high frequencies mean a turbulent boundary layer. Of course, the neural network will abstract these learning steps in a way that we cannot understand, but regardless, the input and output data are further apart.

It is relatively easy to lift a single time-series into different spectral coefficients. In this example, the input time series was lifted into Mel-frequency cepstral coefficients using the Librosa library by Ellis et al. [59]. Mel-frequency cepstral coefficients (MFCC) [211] are usually used as a model for what humans can hear, and not for a technical application, but the method worked quite well for this case. Training the recurrent neural network with only the time-series signal was difficult, given the restricted dataset available. With the MFCC features, the classifier worked well. For this example, domain knowledge about the boundary layer and signal processing helped engineer a feature that was very effective to use in a model.

[20] This advice is one of the practical tips given in the Practical Deep Learning course by Howard and Others [83]

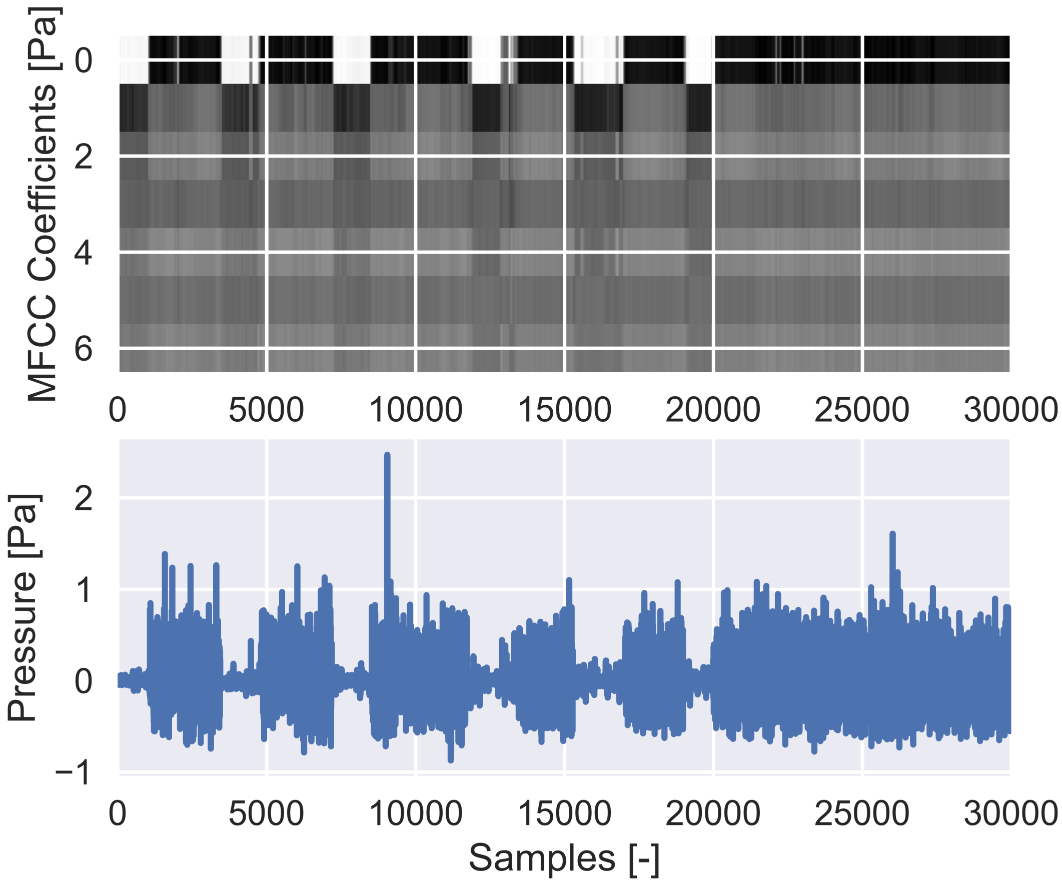


**Figure 7.10:** A pressure time series lifted into MFCC

As we can see, the resulting classifier performed well on a task that was reasonably simple, once the correctly engineered features were available.

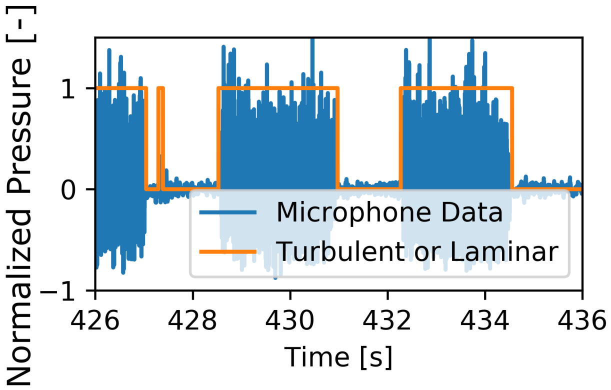


**Figure 7.11:** The predictions resulting from the recurrent neural network

In *'Modern methods for investigating the stability of a pitching floating platform wind turbine'*, an example of another engineering feature was given - the instantaneous aerodynamic damping of a pitching wind turbine. A machine-learned controller could eventually learn the relationship between the thrust and the motion, and how it affects the pitch stability. The controller can be handed a shortcut for the process by including this feature from the beginning, with the controller now trying to learn in a world where thrust and motion have a relationship to the stability of the turbine. The world looks simpler with well engineering features.

Bowles et al. [22] and Lennie et al. [112][21] both demonstrated a similar formulation in action, with the instantaneous aerodynamic damping formulation was derived for an airfoil by Bowles et al. [22]. The formulation identifies the negative effects of trailing edge stall on flutter stability. Any machine-learned model would benefit from this knowledge.

In the context of wind energy, there are many first-order equations and derivations that do have limitations but can still make fantastic input features into machine-learned approaches. This is at the heart of strategy two - use the domain knowledge built over the last generation of wind-energy development as input features in the data-driven approaches.

Another feature engineering approach is to cluster our operating domain into smaller chunks. In *"Cartographing dynamic stall with machine learning"*, we saw and example of clustering being used to help understand the data. An additional step would be to build a flow control system where each cluster has its own unique control [145]. Clustering methods don't just help us understand our data but actively provide us with a reduced-order system with which we can train models.

## 7.3 Strategy 3 - Machine Learning to move between scales

Wind-energy aerodynamics simulations will always suffer from the effects of being multi-scale problems. Simulations at the scale of wind farms have to simplify the appearance of wind turbines in the field. Simulations of a single wind turbine will usually have to simplify the representation of the inflow conditions based on statistics. Running a precursor simulation to generate inflow turbulence is possible, but costly. It would be better to use a cheap model that produces good results.

Augmenting a low resolution inflow field to include turbulence, looks a lot like the image super-resolution approaches that have become popular in the machine learning community. Wang et al. [208] provides a thorough overview of the specific approaches. Here, it is enough say that it is possible to turn a low-resolution photograph into a high-resolution photograph using an intelligent, machine-learned interpolation. A wind speed over a plane of a wind turbine field is essentially a photograph, with three wind components instead of three color channels.

[21] Associated publication

To train such a model in a supervised manner, it would be necessary to have a set of low-/high-resolution image pairs. The training data would have to cover a range of scenarios, including different terrain, roughness and the influence of the other wakes from other machines. The training data would have to be resolved at quite a high level, and this would present a challenge. However, once achieved, it might be possible to take a large-scale wind simulation with a coarse mesh (for an individual CFD simulation) and upgrade it to include much smaller scales of turbulence. Given that wind field data is structurally the same as photographic data, it would also be possible to use pretrained networks from general photographs, thus reducing the data and computational requirements.

Such an approach would be most useful for the investigation of wind-farm-level control. Using a plant-level controller has the advantage of making it possible to run one wind turbine suboptimally in order to optimize the performance of the entire plant. Several experimental and numerical studies have investigated the concept of wake steering; that is, deliberately yawing a wind turbine so that the wake steers away from the rotor of the second row of the plant [165, 64]. The upstream rotor will experience higher loads due to the yaw misalignment but the downstream rotor will experience a cleaner inflow, leading to high power and lower loads. For numerical simulations, it is critical to conduct the large wind farm simulations in a way that resolves atmospheric effects which can strongly influence the wake structure and therefore the effectiveness of wake steering [165]. But this level of simulation may not provide enough resolution to calculate the difference in loads experienced by each wind turbine. A machine learned approach could convert the coarse wind field simulation data into high resolution inflow data for individual wind turbines. The machine learning approach would be initially computationally intensive to train, but, usually, inference is orders of magnitude computationally cheaper than training, at least with neural networks, which are the most likely candidates for such an approach.

## 7.4 Strategy 4 - Sparse Sensing

Sparse sensing is a method used for intelligently extrapolating a full-state estimation from a small numbers of features. The assumption is that there are strong underlying patterns in the data that are repeatable and predictable [126]. We can use these known patterns to our advantage, placing sensors only where necessary to measure these patterns. Sensors are expensive to install and manage, therefore its a great advantage to obtain a full state estimation from a limited number.

A good example of this would be for an airfoil. Let us assume for now that the airfoil is not experiencing stall. If we measure pressure on the airfoil surface, we notice that there is a coherent pattern. We notice that as the flow accelerates around the curvature of the suction side of the airfoil, resulting in a strong suction pressure. At the trailing edge, we can see the effects of the Kutta condition take hold, thus decelerating the flow to enforce the continuity condition

[4]. There are physical rules to the system.

When a system, such as an airfoil, has physical rules, it means that the system is constrained. From all the infinite numerical combinations of pressure that the sensors could measure, there is only a small set that obeys the Navier-Stokes equations. If we place large numbers of pressure sensors across the airfoil, the chances are that most of the sensors are measuring redundant dimensionality once we have enough data to train a model. The system is sparse, and we can leverage that in a special way.

Manohar et al. [126] provided an in-depth demonstration of how to exploit 'known patterns'[22]. Explicitly said, they advocated for using a tailored set of basis functions to represent the flow. The alternative would be to compress the dimensions using prescribed basis functions, such as in a Fourier series or Wavelet Transform, which may not be optimally sparse. Instead they advocate using a PCA to find the modes ranked in order of the variance they describe in the system. Essentially, these basis functions recognise that the Navier-Stokes equations constrain the system, encoding these rules into the highest ranked modes of the decomposition.

When the underlying physical system is dimensionally low, it is possible to truncate the number of modes used to represent the full reconstruction without causing much of an error [193]. We can see that, for the airfoil demonstration case, by the sixth component, the variance explained by additional components is close to zero (see Figure 7.12). It turns out that to predict to what degree each of these modes is present, only a small number of sensors is necessary. But which sensors? Choosing the sensors is a tricky problem because combinatorial searches are, in most cases, not computationally possible [126]. We also do not want to choose sensors that provide the same information.

[22] The following three paragraphs briefly describe the concepts from Manohar et al. [126]

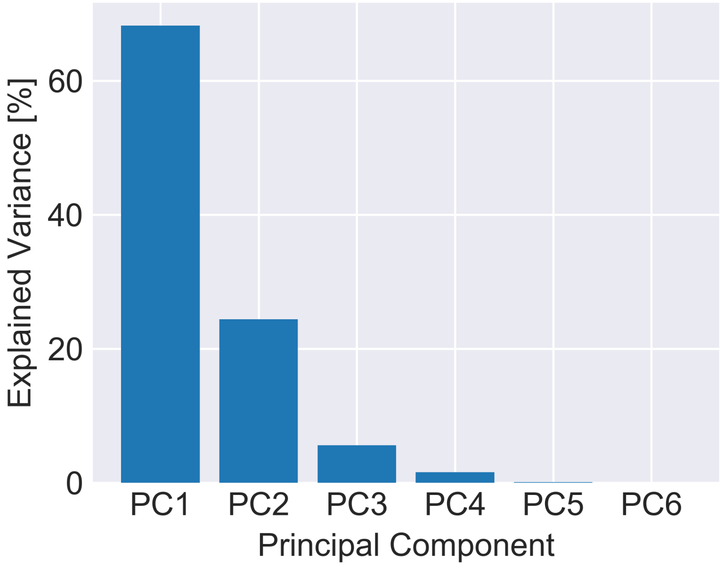


**Figure 7.12:** The principle components of airfoil pressure

Here, it is possible to use the QR decomposition, with pivoting, to rank-order the sensors. The QR decomposition with pivot has a nice property in that it selects the sensor with the highest predictive power as the first pivot, essentially, the first selected sensor that can be used most efficiently for predicting the low order representation of the airfoil pressure. The QR decomposition then removes the information provided by this sensor from the remaining sensors during the Householder step. This means that the next pivot will be the second most effective, given that we already have the information from the first sensor. As we continue, we can efficiently rank-order the sensors. However, this method doesn't account for every single combination; this is what is known as a greedy search.

Once we have selected the sensors, we need to use this data to reconstruct the full state. The first step is to find the mode coefficients that result in the what is being measured by the sensors (Figure 7.13). These coefficients are then multiplied out over the entirety of the mode shapes to provide the full reconstruction (Figure 7.14).

Let us now look at this method in action on an airfoil. Figure 7.15 shows a toy version of the airfoil problem, simply using airfoil pressures generated with XFoil [53]. With three sensors and two PCA components, it was possible to get 98% accuracy on the validation set. Naturally, real data will make the problem more difficult, but this example should provoke ideas of other possible applications in other areas of wind energy.

The same formulation can also estimate an entire wind field from a sparse set of anemometers [8]. Meteorological masts and LIDAR's are too expensive to install enough in each wind field to get a fine spatial resolution across an entire wind farm. Knowing the exact inflow

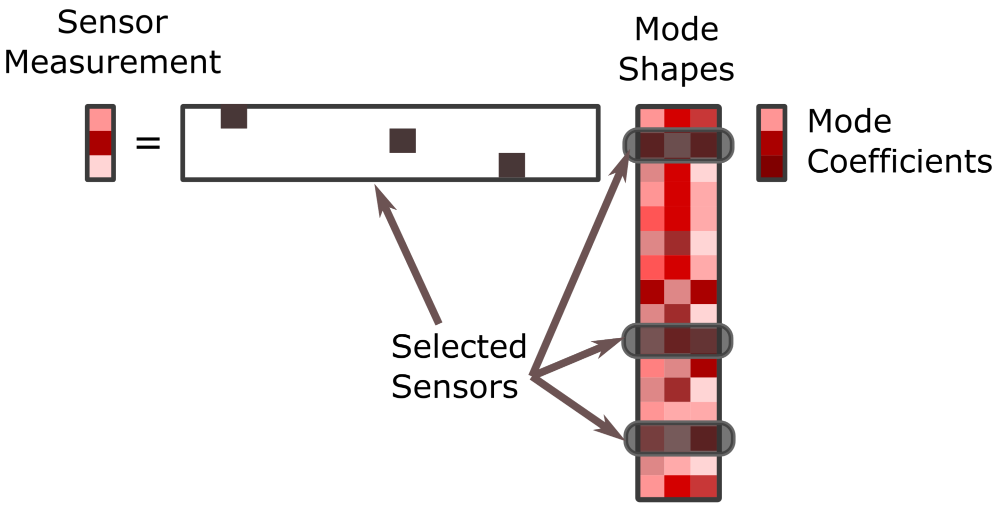


**Figure 7.13:** The principle components of airfoil pressure with a small number of selected sensors (adapted from Manohar et al. [126])

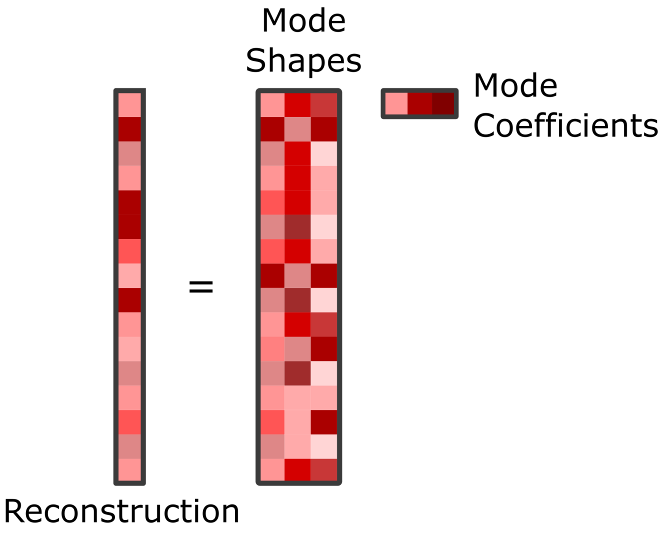


**Figure 7.14:** Reconstruction of the airfoil using three sensors

conditions experienced by each wind turbine at each time would help to identify machines that are performing badly, or that are experiencing higher than expected loads. Annoni et al. [8] achieved reasonable accuracy, but they required 50 sensor points.

In the description of this method so far, we have concentrated on the Manohar et al. [126] method. But PCA modes are simply one reduced order representation that we can leverage for this problem. Other methods, such as local linear embedding, also show promise. Figure 7.16 shows an example of a local linear embedding being used as an auto-encoder. First tests show that such an embedding can represent such a system with suitable accuracy, the trick will now be creating a sparse prediction of the embedded coordinates. This demonstration simply shows that

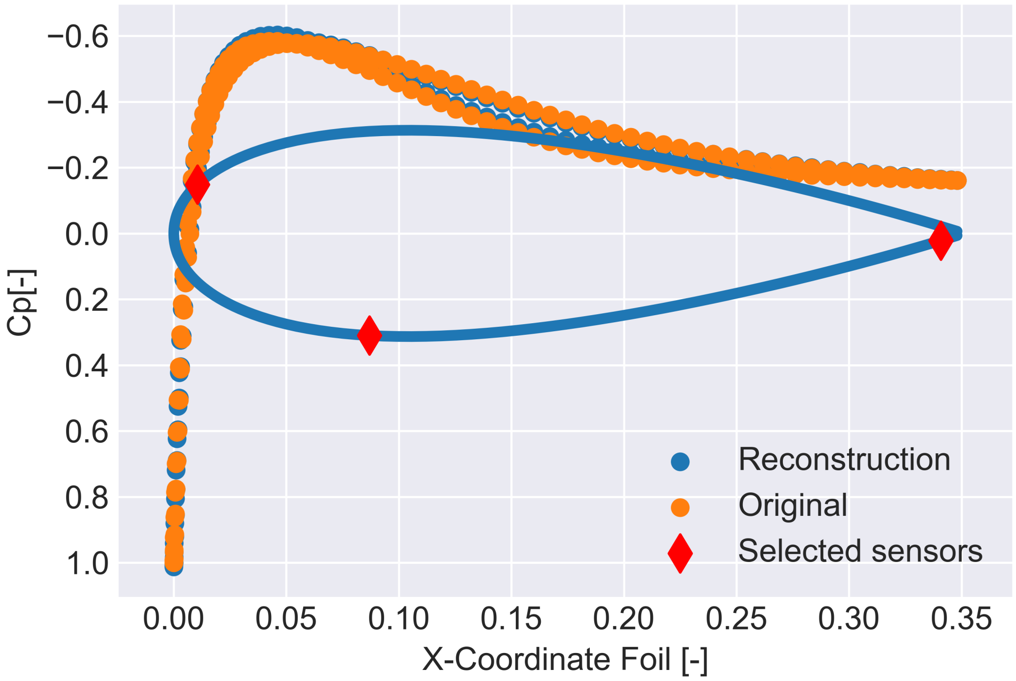


**Figure 7.15:** Reconstruction of the airfoil using three sensors

dimensionality reduction can take multiple forms, which can be useful for different datasets. Having a selection of methods helps in applying sparse sensing across a wide range of fields in a wind-energy context, such as control, structural dynamics and even manufacturing. The wind energy operating environment is incredibly hostile to sensors; Strategy 4 helps us better use the sensors we *can* place on a wind turbine.

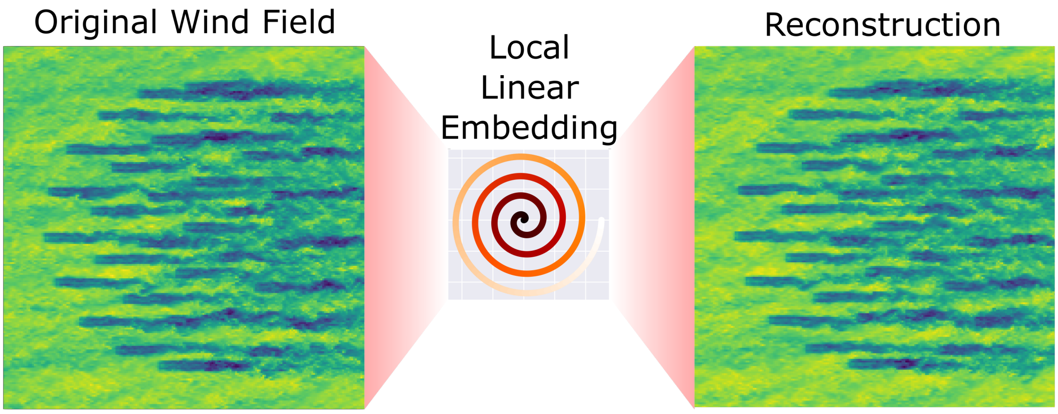


**Figure 7.16:** Reconstruction of a wind field from the validation set

## 7.5 Strategy 5 - Efficiency through convergence criteria

The IEC61400 and DNV GL guidelines provide the basic rules for conducting load case simulations. For extreme loads, the main estimation approaches are extrapolation or Monte Carlo [72]. Monte Carlo approaches are very inefficient at testing the tails of datasets, and take a very long time to converge. Furthermore, a high-dimensional system like a wind turbine, basic samplers such as the Metropolis-Hastings Markov chain Monte Carlo, will struggle to even sample the central moments efficiently [144]. Extrapolation out to extreme events where there is no sampled data is also questionable. Graf et al. [72] suggested using a combination of sampling techniques in stratified regions to solve this problem, although the work is only preliminary.

To demonstrate the issue, we can look at another sampling-based problem - ice fall risk assessments, which tend to be tricky. Ice formation and shedding tend to occur during inclement weather conditions, making it difficult to measure when and how the ice sheds from the wind turbines. The most common approach is to use simple ballistics with randomly-sampled ice chunks.

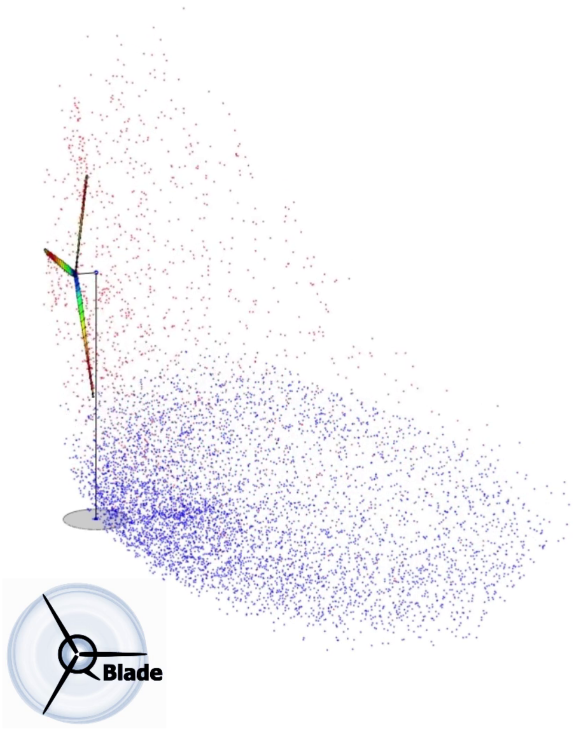


**Figure 7.17:** Ice throw simulation in QBlade [117]

The risk assessments have to determine the risk contours for $10^{-4}$ - $10^{-7}$ chance of someone being hit per square metre per year [99]. These are very much on the tails of the distribution. If we use a Monte Carlo sampling approach, we would need a huge number of samples to converge these risk thresholds with reasonable confidence. Instead, it is much more efficient to use uniform distributions for all inputs and operating conditions, and then weight the individual results based on their likelihood at the end of the calculation (see Figure 7.20) [107, 48, 116]. This has a second advantage; when we conduct bootstrapping to estimate the convergence of the tail contours, we are conducting the sampling on uniform distributions, meaning that an 'm from n' bootstrap is not unreasonable. Using this approach, we could show that it requires only 60,000-70,000 particles in a standard simulation to define the safety boundaries within a 20 m tolerance (see Figure 7.19) [117][23]. In this context, given the input data uncertainties[24], adding 20m is not an unreasonable compromise (Figure 7.18).

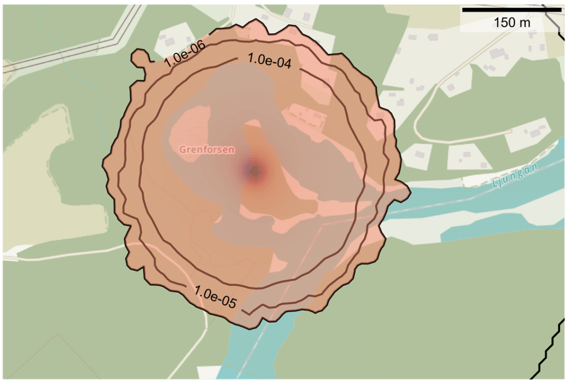


**Figure 7.18:** Risk contours overlaid onto a map [117]

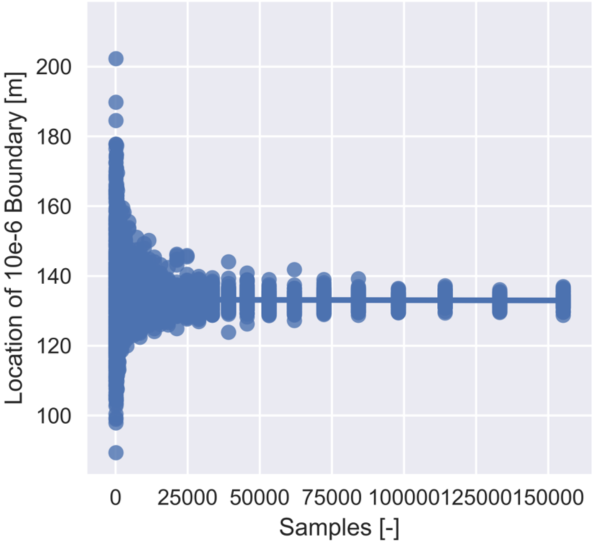


**Figure 7.19:** Convergence of the $10^{-6}$ probability boundary [117]

[23] Associated publication

[24] In "Uncertainties and choices in ice risk assessments: How to get the results you want", Drapalik elucidates just how wide the uncertainties are [52]

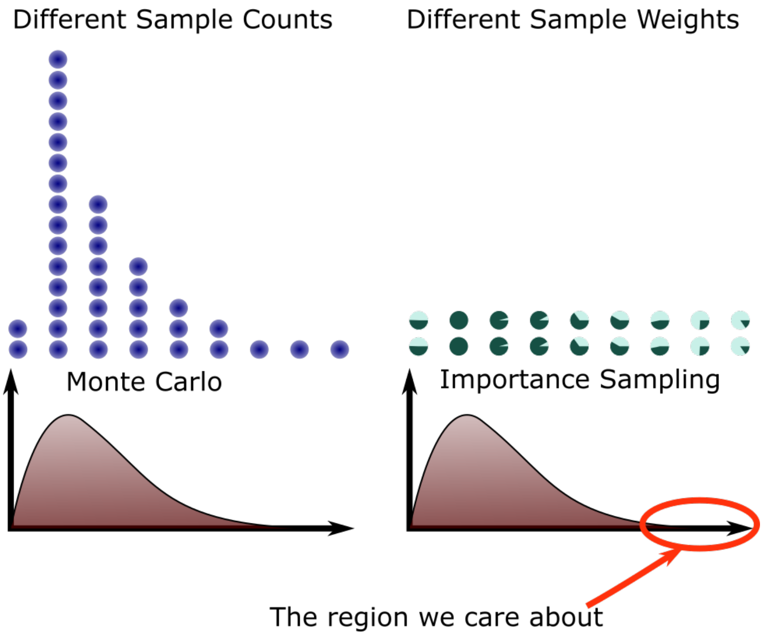


**Figure 7.20:** Monte Carlo versus importance (or uniform) sampling

Ice fall is not the only environmental phenomenon that needs to be treated stochastically. Simulating the structural response of Earth quakes is also approached statistically. From all of the possible earth quakes, we can derive range of statistics. From these we derive a frequency spectrum as a baseline and add sampled noise (Figure 7.21). The spectrum is converted into a time series and then placed in the aeroelastic simulation of QBlade. We can also see here, through bootstrapping, that we must ensure an adequate number of sampled earthquakes, to ensure that we properly representing the statistics of the possible earthquakes (Figure 7.22).

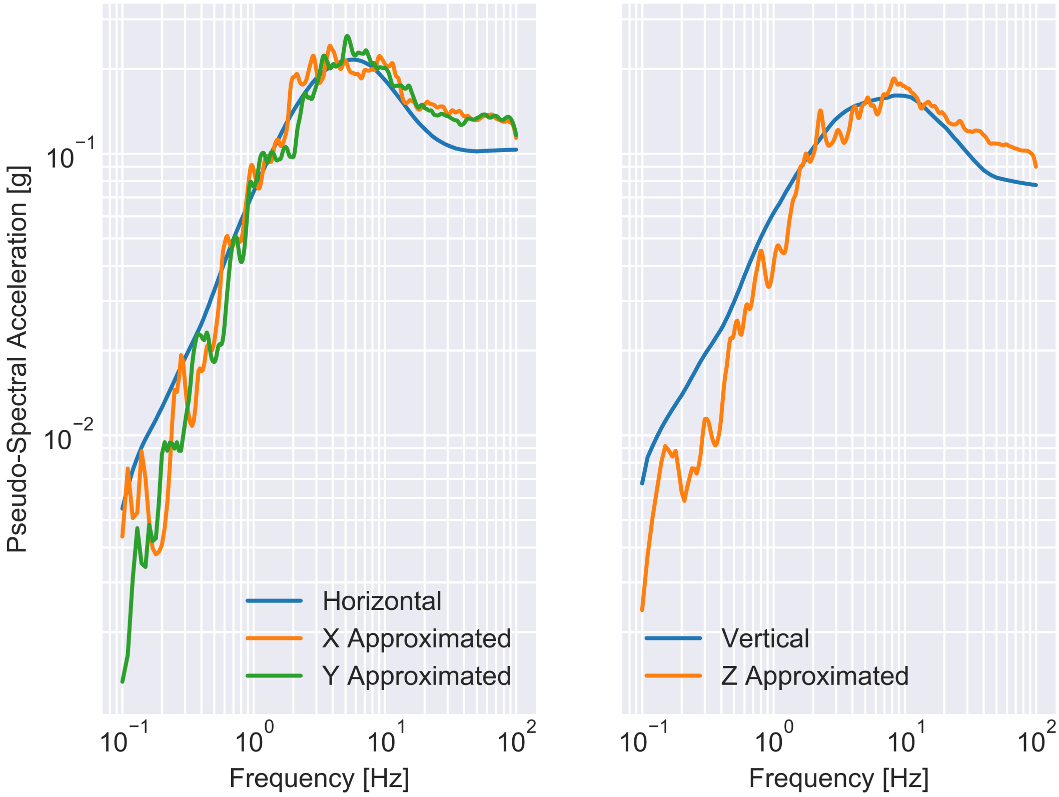


**Figure 7.21:** Creating a random earthquake

The same knowledge can be used in load-case simulations. By recognising exactly what sampling algorithms we are using in load-case simulations, we can also start to understand how sensitive our results are to our dataset. This can lead to smarter, more intelligent sampling approaches. Furthermore, Bos [19] pointed out the important concept that an exceedingly rare event, like a Mexican hat gust, does not necessarily lead to the most dangerous or damaging scenario. We can also understand how to properly implement our safety factors when using sampling methods (see Figure 7.23). This is critically important if we want to use more expensive simulation tools, such as FVLLT or CFD, where we want to minimise the number of cases being computed.

We should also be careful when our models use an expected value, but the reality is a distribution. One example are the material properties of composite materials, another is the cycle-to-cycle variability of aerodynamic responses during airfoil separation - a topic discussed at length in this dissertation. We should be careful in claiming convergence when we forced stochastic parameters to be constant without suitable justification. In aerodynamics [114, 66][25] and composites [35, 36, 170], there are examples of modifying engineering models to include a stochastic perturbation. [186] even go as far to recommend taking a probabilistic approach to the whole design process.

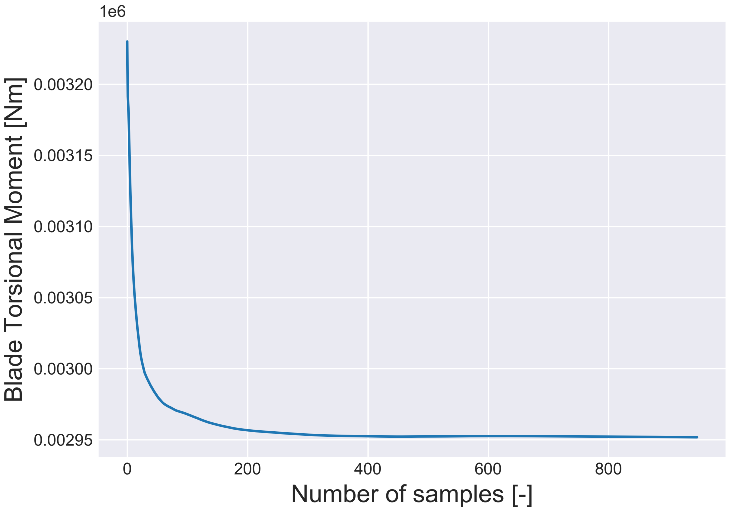


**Figure 7.22:** Convergence of the mean root torsional moment with a growing selection of earthquakes

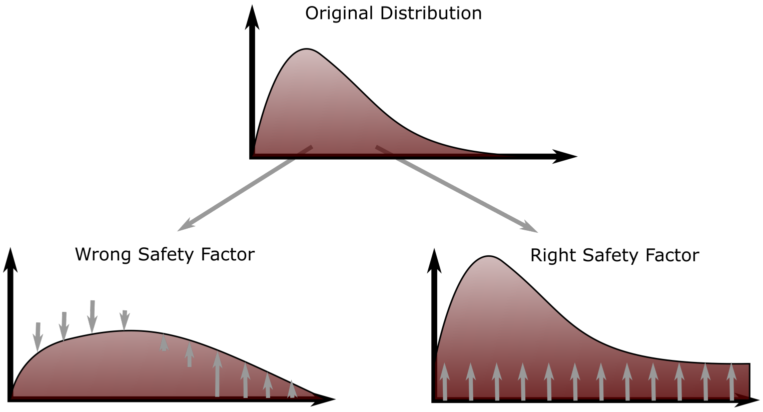


**Figure 7.23:** Understanding how modifying safety factors can falsely modify distributions

[25] Associated publication

## 7.6 Strategy 6 - Machine Learn a small part of the system

Wind turbines operate in hostile environments under a complex set of constraints. The controllers require a range of inputs that are more complicated than a sensor alone can provide. A good example of this is ice accumulation on wind turbine blades. As ice accumulates, the extra mass and the modified aerodynamics present an efficiency and loads problem [44]. The wind turbine needs to have an accurate estimation of the icing for two reasons; is a shut down required?; and/or should the anti- or de-icing systems be activated? Whether or not a wind turbines blade's are iced can be determined through a number of methods including:

1. Changes in modal frequency measured using strain gages [198];
2. Time series extrapolation from a nacelle-mounted anemometer pair [30]; and
3. Deviation from the power curve [44].

All of these methods can be compressed down to different versions of a classifier: icing? yes/no.

The modal frequency approach uses sparse sensing on a reduced-dimensionality representation of the blades (i.e. blade modes) [198]. The icing sensors have to account for icing onset and dissipation occuring at different times, resulting in some offsets [30]. The deviation from the power curve has to take in noisy data, and separate out real changes in operating conditions from the base-level noise. In short this method represents online change point detection of noisy data [44]; quite similar to the vortex generator analysis discussed in the introduction. The approaches here take different input features but all produce in a classifier at the end.

For the icing case, the machine-learned 'sensor' provided input to the supervisory controller, providing a better feature for the control system to work with. It was unnecessary to learn an entire controller with machine learning to start yielding benefits.

The icing sensor is a specific example of anomaly detection, which asks, is the operational data normal or is something weird [213, 212]? Due to the manufacturing tolerances and different operating conditions for wind turbines, it is often difficult to hard-code thresholds, for example, if the first flap frequency drops to 0.8 Hz, then there is a fault. Each machine has its own profile of normal behaviour. The data quality is an issue, but there are examples of research making headway, as seen in the review paper by Helbing and Ritter [80]. A machine-learning method would incorporate data, creating and tuning the model from scratch or perhaps transfer learning from a base example.

In some scenarios, it would be useful to operate the wind turbine in an unusual configuration to meet the demands of the grid. 'Curtailing' is the act of operating a wind turbine or farm under its capability to guarantee the power that was sold ahead of time. To do this efficiently, we need accurate knowledge of the offset between the current power generation and the maximum achievable power. This is difficult in a wind farm where the wakes of each wind turbine will affect the wind speed experienced by adjacent wind turbines. Essentially, we need a way of including the behaviour of the wind farm in the model. The behaviour can be estimated through physics-based engineering models, but the engineering models can be inadequate [70]. Recent work has used a recurrent neural network to produce a better estimation of the percentage reserves [71]. In this case, the wind turbine controller would then have an online estimate of the current performance compared to theoretically available. Again, notice here that we are gaining from a machine learned model with a relatively modest task.

This strategy is all about finding small parts of the models or monitoring systems that can be replaced with a machine-learned model for better performance. The more we can do with small data, small computational efforts and short development times, the better.

## 7.7 Strategy 7 - Machine learn the whole task

So far, I have been discussed where machine learning can make small improvements to parts of simulations or processes. The idea was to demonstrate that wind energy can benefit from a range of small implementations of modest tasks rather than an omniscient AI that will remove humans from the design and operation of wind turbines completely. Sometimes, there are tasks where, there are only two options; intense human input or machine learning. Two examples of this are rotor-blade inspections and automated maintenance reporting.

Rotor-blade inspections usually require a trained rope access crew to manoeuvre around the blades to conduct an assessment. For obvious reasons, the wind turbine can not be operated during this time. The combined cost of the crew and the shut down means that inspections are a costly task. However, a major part of the assessment is visual, answering questions such as:

1. How bad is the leading edge erosion?;
2. Have vortex generator panels lost wings or fallen off completely?;
3. Is there lightning damage on the blades?;
4. Are the trailing edges delaminated; and
5. Is the leading edge tape delaminated?

There are some questions that require a knock test or a more physical inspection, such as:

1. How deep does the crack in the leading edge bond line go?; and
2. Has the skin of the blade separated from the spar caps?

It seem difficult to create a machine-learning method to cover all of these functions, but if the visual component can be performed with operators on the ground, it would still save time and money.

With the continued development of convolutional neural networks, a range of tasks that were previously very difficult have become possible. A key task is the labelling of images for their content; that is, this picture contains a vortex generator or this picture has leading-edge erosion [181]. There have now been multiple implementations of mounting cameras to drones to create inspection images which are then post-processed by a neural network. The post-processed data can then be quickly checked by a structural engineer to recommend repairs or on blade inspection. This is a great example of where machine learning replacing cost intensive work.

Another example is the interpretation of the reports filled out by maintenance teams. Each time a maintenance crew does some work on a wind turbine, they fill out a report describing the work done. The input into these reports is not standardised and the crews often use abbreviations or synonyms (Generator becomes: 'gen', 'gen.', 'genie', 'gener' and so on). This means that a simple word search is labour-intensive to program because, for every rule a human can think to encode, the data will contain many more exceptions. However, a data-driven approach can automatically encode such rules [174]. Furthermore, dimensionality-reduction techniques can identify when alternate spellings are indeed the same word. Having an engineer or technician react to summary data, rather than the individual reports, is obviously economically sensible.

# Chapter 8 | Conclusion

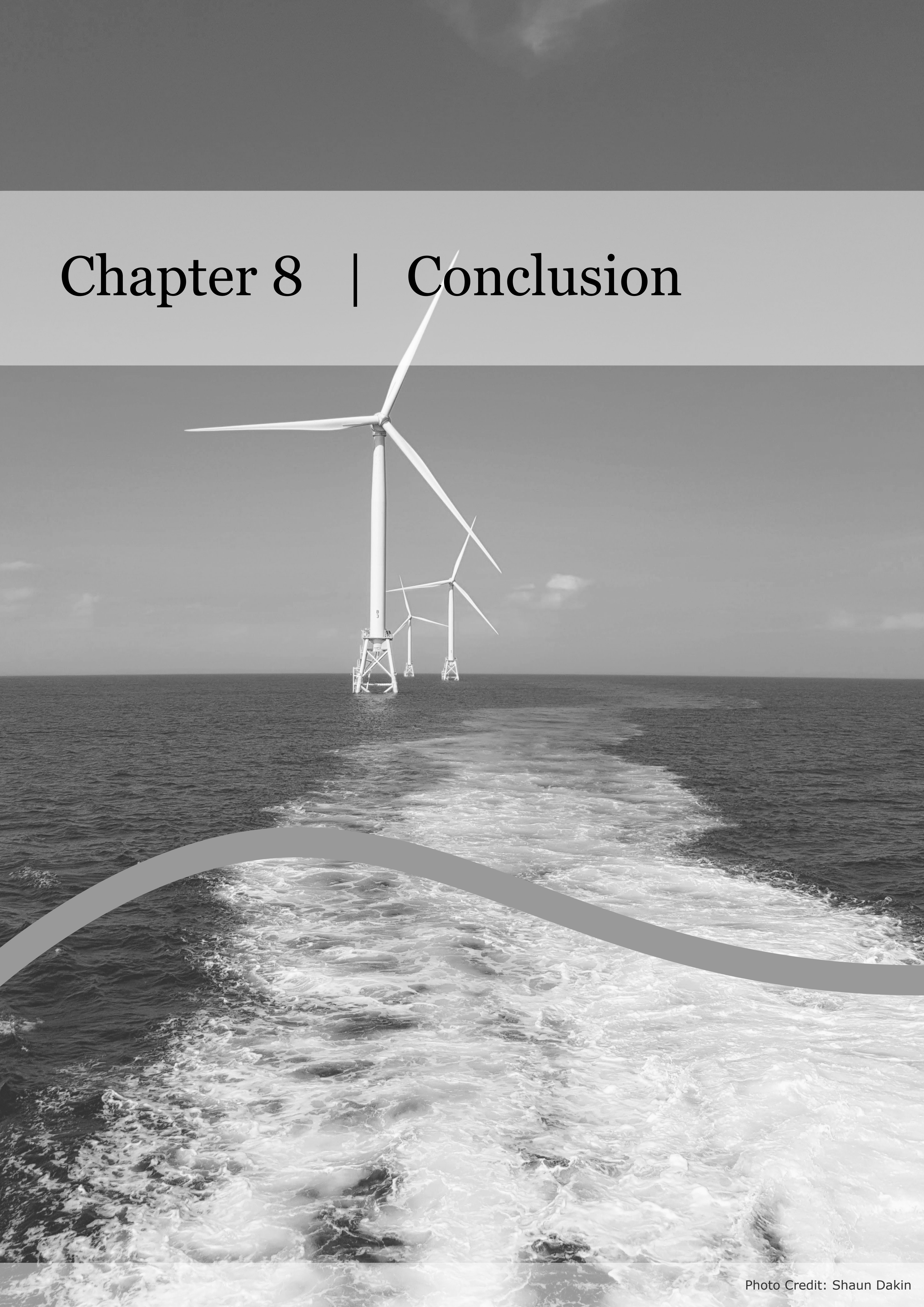


Photo Credit: Shaun Dakin

# Chapter 8 Conclusion

In this study, we have repeatedly revisited a number of themes. The numerical models we use in wind energy are critical for the continuing success of the wind industry. We can see that the engineering-model approach has been, and continues to be, a highly effective way of designing and operating wind turbines. However, these models tend to have some strong assumptions baked in. High-fidelity models continue to make progress from the blade tips to the foundations, but many of the models are quite computationally expensive. It has been shown in this dissertation that the new data science approaches will help us not only understand our data, but also to bring this data into our models. In the discussion, I outlined a number of recommendations for how to integrate the knowledge of data science and machine learning into our engineering approaches. Some of these examples are already being used in industry, others are works in progress.

This study sought to create solutions that:

1. Improve simulation tools to match the expanding needs of wind energy;

2. Provide better engineering models and analysis methods that meet modern challenges; and

3. Demonstrate better machine learning usage to help mine and model data.

All three of these solutions were achieved (Figure 8.1).

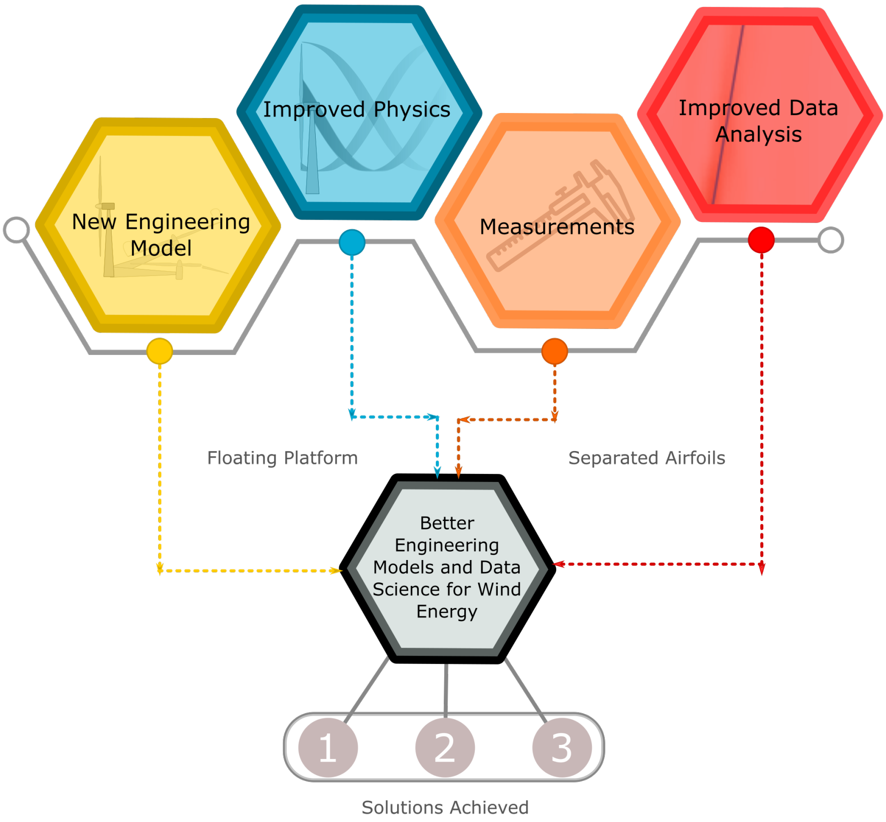


**Figure 8.1:** The Impact

These achievements will most obviously impact the load simulations of wind turbines, particularly for complicated cases like floating platform wind turbines. The instantaneous aerodynamics damping will prove a useful design feature and analysis feature for floating platform wind turbines. The experimental data provides a good insight into the complicated flows of stand-still wind turbine blades and provides a baseline for model development and simulation. The new methods for dissecting airfoil stall should change the way we interpret testing data and build low-order models. However, I believe that along with the specific results of the this study, my seven strategies will prove a vital breakthrough for machine learning in wind energy. At the end of this work, I hope that the reader shares my optimism for the future of machine learning and data science in wind energy without forgetting the noble engineering models that will continue to be the backbone of the clean energy revolution.

# Bibliography

*(excluding bibliography of publications I-IV)*

[1] Hywind - leading floating offshore wind solution. URL https://www.equinor.com/en/what-we-do/hywind-where-the-wind-takes-us.html.

[2] Abdallah, I., Natarajan, a., & Sørensen, J. Impact of uncertainty in airfoil characteristics on wind turbine extreme loads. *Renewable Energy*, 75:283–300, mar 2015. ISSN 09601481. doi:10.1016/j.renene.2014.10.009. URL http://linkinghub.elsevier.com/retrieve/pii/S096014811400634X.

[3] AI Impacts, . Recent trend in the cost of computing, 2018. URL https://aiimpacts.org/recent-trend-in-the-cost-of-computing/.

[4] Anderson, J. D. *Fundamentals of Aerodynamics*. Aeronautical and Aerospace Engineering Series. McGraw-Hill, 3 edition, 2001. ISBN 9780072373356. URL https://books.google.de/books?id=CaBTAAAAMAAJ.

[5] Andrea, A. Nath, C. FAST and ADAMS Incl. AeroDyn Module Certification, 2005.

[6] Andrei Ilas, Pablo Ralon, A. R. & Taylor, M. Renewable Power Generation Costs in 2017. Technical report, International Renewable Energy Agency (IRENA), Abu Dhabi, 2017.

[7] Anne Olhoff, . Emissions Gap Report 2018 Key Messages. Technical Report December, UN Environment, 2018. URL http://wedocs.unep.org/bitstream/handle/20.500.11822/26896/EGR-KEYMESSAGES{_}2018.pdf?sequence=1{&}isAllowed=y.

[8] Annoni, J., Taylor, T., Bay, C., Johnson, K., Pao, L., Fleming, P., & Dykes, K. Sparse-Sensor Placement for Wind Farm Control. *Journal of Physics: Conf. Series*, 1037, 2018. doi:10.1088/1742-6596/1037/3/032019. URL http://iopscience.iop.org/article/10.1088/1742-6596/1037/3/032019/pdf.

[9] a.S. Witteveen, J., Sarkar, S., & Bijl, H. Modeling physical uncertainties in dynamic stall induced fluid–structure interaction of turbine blades using arbitrary polynomial chaos. *Computers & Structures*, 85(11-14):866–878, jun 2007. ISSN 00457949. doi:10.1016/j.compstruc.2007.01.004. URL http://linkinghub.elsevier.com/retrieve/pii/S0045794907000168.

[10] Bacharoudis, K. C. & Philippidis, T. P. A probabilistic approach for strength and stability evaluation of wind turbine rotor blades in ultimate loading. *Structural Safety*, 40:31–38, 2013. ISSN 01674730. doi:10.1016/j.strusafe.2012.09.006. URL http://dx.doi.org/10.1016/j.strusafe.2012.09.006.

[11] Backaert, S., Chatelain, P., & Winckelmans, G. Vortex Particle-Mesh with Immersed Lifting Lines for Aerospace and Wind Engineering. *Procedia IUTAM*, 18:1–7, 2015. ISSN 22109838. doi:10.1016/j.piutam.2015.11.001. URL http://dx.doi.org/10.1016/j.piutam.2015.11.001.

[12] Bak, C., Madsen, H. A., Paulsen, U. S., Gaunaa, M., Fuglsang, P., Romblad, J., Olesen, N. A., Enevoldsen, P., Laursen, J., & Jensen, L. DAN-AERO MW: Detailed aerodynamic measurements on a full scale MW wind turbine. *European Wind Energy Conference and Exhibition (EWEC)*, (April 2015):1–10, 2010.

[13] Baldacchino, D., Manolesos, M., Ferreira, C., Gonz{\'{a}}lez Salcedo, {\'{A}}., Aparicio, M., Chaviaropoulos, T., Diakakis, K., Florentie, L., Ramos Garc{\'{i}}a, N., Papadakis, G., Sorensen, N. N., Timmer, N., Troldborg, N., Voutsinas, S., & van Zuijlen, A. Experimental benchmark and code validation for airfoils equipped with passive vortex generators. *Journal of Physics: Conference Series - The Science of Making Torque from Wind*, 2016. ISSN 1742-6588. doi:10.1088/1742-6596/753/2/022002.

[14] Bay, C. & Sjöholm, M. *Remote Sensing of Atmospheric Conditions for Wind Energy Applications*. MDPI, Basel, 2019. ISBN 978-3-03897-943-2. doi:10.3390/books978-3-03897-943-2. URL http://www.mdpi.com/books/pdfview/book/1308.

[15] Bender, E., Andersen, B., & Yagle, P. Vortex Generator Modeling for Navier-Stokes Codes. *ASME*, FEDSM9vols, 1999.

[16] Berg, J. C., Resor, B., Paquette, J., & White, J. SMART Wind Turbine Rotor: Design and Field Test. Technical Report January, Sandia, Albuquerque, New Mexico, 2014. URL http://energy.gov/sites/prod/files/smart{_}wind{_}turbine{_}design.pdf.

[17] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer International Publishing, Redmond, 1st edition, 2006.

[18] Boersma, S., Doekemeijer, B. M., Gebraad, P. M., Fleming, P. A., Annoni, J., Scholbrock, A. K., Frederik, J. A., & Van Wingerden, J. W. A tutorial on control-oriented modeling and control of wind farms. *Proceedings of the American Control Conference*, pages 1–18, 2017. ISSN 07431619. doi:10.23919/ACC.2017.7962923.

[19] Bos, R. *Extreme gusts and their role in wind turbine design*. Phd thesis, Tu Delft, 2017. doi:10.4233/UUID. URL https://doi.org/10.4233/uuid:d6097e3a-1cdd-4845-a71c-90f469d28b7a.

[20] Bottasso, C. L., Campagnolo, F., & Petrović, V. Wind tunnel testing of scaled wind turbine models: Beyond aerodynamics. *Journal of Wind Engineering and Industrial Aerodynamics*, 127:11–28, apr 2014. ISSN 01676105. doi:10.1016/j.jweia.2014.01.009. URL http://linkinghub.elsevier.com/retrieve/pii/S0167610514000269.

[21] Bottasso, C., Campagnolo, F., Croce, A., & Maffenini, L. Development of a wind tunnel model for supporting research on aero-servo-elasticity and control of wind turbines. In *13th International Conference on Wind Engineering*, 2011. URL http://folk.ntnu.no/karpa/ICWE13/ICWE13Proceedings/start/papers/052{_}8page{_}development{_}of{_}a{_}wind{_}tunnel{_}model{_}for{_}supporting{_}research{_}on{_}aeroelasticity{_}and{_}control{_}of{_}wind{_}turbines.pdf.

[22] Bowles, P. O., Corke, T. C., Coleman, D. G., Thomas, F. O., & Wasikowski, M. Improved Understanding of Aerodynamic Damping Through the Hilbert Transform. *AIAA Journal*, 52(11):2384–2394, nov 2014. ISSN 0001-1452. doi:10.2514/1.J052630. URL http://arc.aiaa.org/doi/abs/10.2514/1.J052630.

[23] Box, G. E. P. Science and statistics. *Journal of the American Statistical Association*, 71 (356):791–799, 1976. ISSN 1537274X. doi:10.1080/01621459.1976.10480949.

[24] Branlard, E. *and aeroelasticity using vortex-based methods DTU Wind Energy PhD Thesis 2015*, volume 0052. 2015. ISBN 9788793278448.

[25] Brunton, S. L., Proctor, J. L., & Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. ISSN 0027-8424. doi:10.1073/pnas.1517384113.

[26] Bughin, J., Hazan, E., Ramaswamy, S., Chui, M., Allas, T., Dahlstrom, P., Henke, N., & Trench, M. Artificial Intelligence: the Next Digital Frontier? Technical report, 2017. URL {%}60.

[27] Buhl, M. OSU Wind-Tunnel Test Data, 2012. URL https://wind.nrel.gov/airfoils/OSU{_}data/data/.

[28] Burton, T., Jenkins, N., Sharpe, D., & Bossanyi, E. *Wind Energy Handbook*. Wiley, 2011. ISBN 9781119993926. URL http://books.google.de/books?id=dip2LwCRCscC.

[29] Butterfield, C. & Nelsen, E. Aerodynamic testing of a rotating wind turbine blade. Technical report, Solar Energy Research Institute, Golden, Colorado, 1990. URL http://www.nrel.gov/docs/legosti/old/3490.pdf.

[30] Byrkjedal, Ø. IceLoss – 10 years with Hands-on Experiences from Icing on Wind Turbines, Kjeller VindTeknikk. In *Wind Turbine Icing & Ice Prevention Forum*, Berlin, 2018. BIS Group.

[31] Cairns, D. S., Nelson, J., & Riddle, T. Wind Turbine Composite Blade Manufacturing: The Need for Understanding Defect Origins, Prevalence, Implications and Reliability. Technical report, Sandia, Albuquerque, New Mexico, 2011.

[32] Canet, H., Bortolotti, P., & Bottasso, C. L. Gravo-aeroelastic scaling of very large wind turbines to wind tunnel size. *Journal of Physics: Conference Series*, 1037(4):0–9, 2018. ISSN 17426596. doi:10.1088/1742-6596/1037/4/042006.

[33] Castro, O., Lennie, M., Branner, K., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. Comparing fatigue life estimations of composite wind turbine blades using different fatigue analysis tools. *ICCM20 - 20th International Conference on Composite Materials*, (July):19–24, 2015.

[34] Castro, O., Lennie, M., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. The Use of a New Fatigue Tool (ALBdeS) to Analyse the Effects of Vortex Generators on Wind Turbines, 2015. URL http://dx.doi.org/10.1115/GT2015-43198.

[35] Castro, O., Branner, K., & Dimitrov, N. Assessment and propagation of mechanical property uncertainties in fatigue life prediction of composite laminates. *Journal of Composite Materials*, page 002199831876562, 2018. ISSN 0021-9983. doi:10.1177/0021998318765626. URL http://journals.sagepub.com/doi/10.1177/0021998318765626.

[36] Castro-Ardila, O.-G. *Fatigue strength of composite wind turbine blade structures*. Phd thesis, DTU, 2018.

[37] Cave, D. It Was Supposed to Be Australia's Climate Change Election. What Happened?, 2019. URL https://www.nytimes.com/2019/05/19/world/australia/election-climate-change.html.

[38] Chi, X. & Fang, Z. W. Review of association rule mining algorithm in data mining. *2011 IEEE 3rd International Conference on Communication Software and Networks, ICCSN 2011*, pages 512–516, 2011. doi:10.1109/ICCSN.2011.6014622.

[39] Chollet, F. & Others, . Keras. \url{https://keras.io}, 2015.

[40] contributors, W. Machine learning, 2019. URL `https://en.wikipedia.org/w/index.php?title=Machine{_}learning{&}oldid=907523609`.

[41] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35 (1):53–65, 2018. ISSN 10535888. doi:10.1109/MSP.2017.2765202.

[42] Danielle Muoio, . An ex-Tesla exec reveals how the company is transforming itself into a data powerhouse, jun 2017. URL `https://www.businessinsider.de/tesla-chris-lattner-explains-how-car-data-is-used-2017-6?r=US{&}IR=T`.

[43] Dask Development Team, . Dask: Library for dynamic task scheduling, 2016. URL `https://dask.org`.

[44] Davis, N. N., Byrkjedal, Ø., Andrea N. Hahmann, N.-E. C., & Zagar, M. Ice detection on wind turbines using the observed power curve. *Wind Energy*, 19(22 July 2015):999–1010, 2015. ISSN 1099-1824. doi:10.1002/we.1878. URL `10.1002/we.1878`.

[45] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[46] Di Lorenzo, E., Manzato, S., Peeters, B., Ruffini, V., Berring, P., Haselbach, P. U., Branner, K., & Luczak, M. M. Modal Analysis of Wind Turbine Blades with Different Test Setup Configurations. In Mains, M. L. & Dilworth, B. J., editors, *Topics in Modal Analysis & Testing, Volume 8*, pages 143–152, Cham, 2019. Springer International Publishing. ISBN 978-3-030-12684-1.

[47] DNV GL, . Guideline for the Certification of Wind Turbines. Technical report, Germanischer Lloyd Industrial Services, Hamburg, Germany, 2010.

[48] Dominin, S., Lennie, M., & Paschereit, C. O. Development of Ice Throw Model for Wind Turbine Simulation Software QBlade (Retracted to be resubmitted for Scitech 2019). In *AIAA Scitech 2018*, pages 1–13, Orlando, USA.

[49] Dose, B., Rahimi, H., Herráez, I., Stoevesandt, B., & Peinke, J. Fluid-structure coupled computations of the NREL 5MW wind turbine blade during standstill. *Journal of Physics: Conference Series*, 753(2), 2016. ISSN 17426596. doi:10.1088/1742-6596/753/2/022034.

[50] Dowell, H. *A modern course in aeroelasticity*. Solid mechanics and its applications. Kluwer Academic, 1995. ISBN 9780792327882. URL `http://books.google.de/books?id=6v5TAAAAMAAJ`.

[51] Dowler, J. L. & Schmitz, S. *A Solution Based Stall Delay Model For Horizontal Axis Wind Turbines*. PhD thesis, Pennsylvannie State University, 2013.

[52] Drapalik, M. Uncertainties and choices in ice risk assessments How to get the results you want. In *Winterwind International Wind Energy Conference*, Umea, 2019. URL https://winterwind.se/wp-content/uploads/2019/02/02{_}03{_}Drapalik{_}Uncertainties{_}and{_}choices{_}in{_}ice{_}risk{_}assessments{_}-{_}How{_}to{_}get{_}the{_}results{_}you{_}want{_}Pub{_}v2.pdf.

[53] Drela, M. XFOIL - An analysis and design system for low reynolds number airfoils, 1989.

[54] Driscoll, F., Jonkman, J., Robertson, A., Sirnivas, S., Skaare, B., & Nielsen, F. G. Validation of a FAST Model of the Statoil-hywind Demo Floating Wind Turbine. *Energy Procedia*, 94(January):3–19, 2016. ISSN 18766102. doi:10.1016/j.egypro.2016.09.181. URL http://dx.doi.org/10.1016/j.egypro.2016.09.181.

[55] Du, Z. & Selig, M. S. A 3-D stall-delay model for horizontal axis wind turbine performance prediction. *Proceedings of the 1998 ASME wind energy symposium, Reno, NV*, pages 9–19, 1998.

[56] Dudek, J. C. Modeling Vortex Generators in a Navier-Stokes Code. *AIAA Journal*, 49(4): 748–759, 2011. ISSN 0001-1452. doi:10.2514/1.j050683.

[57] ELGAMMI, M. & SANT, T. Predictions of the cycle-to-cycle aerodynamic loads on a yawed wind turbine blade under stalled conditions using a 3D empirical stochastic model. *Journal of Physics: Conference Series*, 753:022035, 2016. ISSN 1742-6588. doi:10.1088/1742-6596/753/2/022035. URL http://stacks.iop.org/1742-6596/753/i=2/a=022035?key=crossref.caedc619428f1465adef31bca9515609.

[58] Elite Data Science, . WTF is the Bias-Variance Tradeoff? (Infographic), jan 2019. URL https://elitedatascience.com/bias-variance-tradeoff.

[59] Ellis, D., Nieto, O., McFee, B., Liang, D., McVicar, M., Raffel, C., & Battenberg, E. librosa: Audio and Music Signal Analysis in Python. *Proceedings of the 14th Python in Science Conference*, (Scipy):18–24, 2015. ISSN 18770428. doi:10.25080/majora-7b98e3ed-003.

[60] Elsken, T., Metzen, J. H., & Hutter, F. Neural Architecture Search: A Survey. *Journal of Machine Learning Research*, 20:1–21, 2018. URL http://arxiv.org/abs/1808.05377.

[61] Engel, J., Resnick, C., Roberts, A., Dieleman, S., & Eck, D. Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. 2009.

[62] Equinor, . Floating offshore wind project in South Korea, 2019. URL https://www.equinor.com/en/news/2019-07-11-floating-offshore-wind-project-in-south-korea.html.

[63] Fleming, P., Lee, S., Churchfield, M., Scholbrock, A., Michalakes, J., Johnson, K., Moriarty, P., Gebraad, P., & van Wingerden, J. The SOWFA super-controller: A high-fidelity tool for evaluating wind plant control approaches. Technical Report January, National Renewable Energy Laboratory, TU Delft, 2013. URL http://www.nrel.gov/docs/fy13osti/57175.pdf.

[64] Fleming, P., Churchfield, M., Scholbrock, A., Clifton, A., Schreck, S., Johnson, K., Wright, A., Gebraad, P., Annoni, J., Naughton, B., Berg, J., Herges, T., White, J., Mikkelsen, T., Sjöholm, M., & Angelou, N. Detailed field test of yaw-based wake steering. *Journal of Physics: Conference Series*, 753(5), 2016. ISSN 17426596. doi:10.1088/1742-6596/753/5/052003.

[65] Fleming, P. A., Peiffer, A., & Schlipf, D. Wind Turbine Controller to Mitigate Structural Loads on a Floating Wind Turbine Platform. *Journal of Offshore Mechanics and Arctic Engineering*, 141(6):061901, 2019. ISSN 0892-7219. doi:10.1115/1.4042938.

[66] Fluck, M. & Crawford, C. A fast stochastic solution method for the Blade Element Momentum equations for long-term load assessment. *Wind Energy*, (March):1–14, 2017. ISSN 10954244. doi:10.1002/we.2148. URL http://doi.wiley.com/10.1002/we.2148.

[67] Gancarski, P. 2D Unsteady Aerodynamics Open Data Set, 2018. URL https://windbench.net/uaero.

[68] Gatys, L. A., Ecker, A. S., & Bethge, M. A Neural Algorithm of Artistic Style. *Journal of Computer Vision*, pages 3–7, 2015. URL http://arxiv.org/abs/1508.06576.

[69] Gaunaa, M., Heinz, J., & Skrzypiński, W. Toward an Engineering Model for the Aerodynamic Forces Acting on Wind Turbine Blades in Quasisteady Standstill and Blade Installation Situations. *Journal of Physics: Conference Series*, 753:022007, 2016. ISSN 1742-6588. doi:10.1088/1742-6596/753/2/022007. URL http://stacks.iop.org/1742-6596/753/i=2/a=022007?key=crossref.74c47a0445c8a5a73ea2e5946de6137e.

[70] Göçmen, T. & Giebel, G. Data-driven Wake Modelling for Reduced Uncertainties in short-term Possible Power Estimation. *Journal of Physics: Conference Series*, 1037(7), 2018. ISSN 17426596. doi:10.1088/1742-6596/1037/7/072002.

[71] Göçmen, T., Meseguer, A., & Yikon, J. Recurrent Neural Networks for Improved Dynamic Delta Control of Turbine Curtailment. 9(8):2016, 2016.

[72] Graf, P., Dykes, K., Damiani, R., Jonkman, J., & Veers, P. Adaptive stratified importance sampling: hybridization of extrapolation and importance sampling Monte Carlo methods for estimation of wind turbine extreme loads. *Wind Energy Science*, 3(2):475–487, 2018. doi:10.5194/wes-3-475-2018.

[73] Gupta, S. & Leishman, J. G. Dynamic stall modelling of the S809 aerofoil and comparison with experiments. *Wind Energy*, 9(6):521–547, nov 2006. ISSN 10954244. doi:10.1002/we.200. URL `http://doi.wiley.com/10.1002/we.200`.

[74] Hand, M. M., Simms, D. A., Fingersh, L. J., Jager, D. W., Cotrell, J. R., Schreck, S., & Larwood, S. M. Unsteady Aerodynamics Experiment Phase VI : Wind Tunnel Test Configurations and Available Data Campaigns Unsteady Aerodynamics Experiment Phase VI : Wind Tunnel Test Configurations and Available Data Campaigns. (December), 2001.

[75] Hansen, K. S., Barthelmie, R., Jensen, L. E., & Sommer, A. The impact of turbulence intensity and atmospheric stability on power deficits due to wind turbine wakes at Horns Rev wind farm. *Wind Energy*, (15 2015):183–196, 2012. doi:10.1002/we.

[76] Hansen, M. H. Improved modal dynamics of wind turbines to avoid stall-induced vibrations. *Wind Energy*, 6(2):179–195, 2003. ISSN 10954244. doi:10.1002/we.79.

[77] Hansen, M. O. L. *Aerodynamics of Wind Turbines*. Earthscan LLC, London Sterling, VA, second edi edition, 2008. ISBN 9781849770408.

[78] Hansen, M., Gaunaa, M., & Madsen, H. A. *A Beddoes-Leishman type dynamic stall model in state-space and indicial formulations*. 2004. URL `http://orbit.dtu.dk/fedora/objects/orbit:90298/datastreams/file{_}7711084/content`.

[79] He, K. & Sun, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9. IEEE, 2016. doi:10.1109/cvpr.2016.90. URL `http://dx.doi.org/10.1109/CVPR.2016.90`.

[80] Helbing, G. & Ritter, M. Deep Learning for fault detection in wind turbines, 2018. ISSN 18790690.

[81] Holierhoek, J. G. *Aeroelastic design of wind turbine blades*. Woodhead Publishing Limited, 2013. ISBN 9780857094261. doi:10.1533/9780857097286.1.150. URL `http://dx.doi.org/10.1533/9780857097286.1.150`.

[82] Holierhoek, J. G., Vaal, J. B. D., Zuijlen, A. H. V., & Bijl, H. Comparing different dynamic stall models. (April 2012):139–158, 2013. doi:10.1002/we.

[83] Howard, J. & Others, . FASTAI, 2019.

[84] Huanga, X., Moghadam, S. M. A., Meysonnat, P. S., Meinke, M., & Schroeder, W. Numerical Analysis of the Effect of Flaps on the Tip Vortex of a Wind Turbine Blade. *International Journal of Heat and Fluid Flow*, pages 1679–1682, 2019.

[85] Hwangbo, H., Ding, Y., Eisele, O., Weinzierl, G., Lang, U., & Pechlivanoglou, G. Quantifying the effect of vortex generator installation on wind power production: An academia-industry case study. *Renewable Energy*, 2017. ISSN 18790682. doi:10.1016/j.renene.2017.07.009.

[86] International Energy Agency, . Electricity generation by fuel, 2019. URL `https://www.iea.org//statistics/?country=WORLD{&}year=2016{&}category=Electricity{&}indicator=ElecGenByFuel{&}mode=chart{&}dataTable=ELECTRICITYANDHEAT`.

[87] Isaacs, R. Airfoil Theory for Flows of Variable Velocity. *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, 12(1):113–117, 1945. doi:10.2514/8.11202. URL `http://arc.aiaa.org/doi/pdf/10.2514/8.11202`.

[88] J Mayo, G. Airfoild in a sinusoidal motion in a pulsating stream. Technical report, NACA, Washington, 1947. URL `https://ntrs.nasa.gov/search.jsp?R=19930082116`.

[89] Jeong, M.-S., Kim, S.-W., Lee, I., Yoo, S.-J., & Park, K. The impact of yaw error on aeroelastic characteristics of a horizontal axis wind turbine blade. *Renewable Energy*, 60:256–268, dec 2013. ISSN 09601481. doi:10.1016/j.renene.2013.05.014. URL `http://linkinghub.elsevier.com/retrieve/pii/S0960148113002590`.

[90] Jha, P., Churchfield, M., Moriarty, P., & Schmitz, S. Accuracy of State-of-the-Art Actuator-Line Modeling for Wind Turbine Wakes. 2013. doi:10.2514/6.2013-608.

[91] Jonkman, J. M. Fast Users Guide. URL `http://wind.nrel.gov/designcodes/simulators/fast/FAST.pdf`.

[92] Jonkman, J. M. Dynamics Modeling and Loads Analysis of an Offshore.pdf. (November), 2007.

[93] Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P.-l., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, R., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick,

M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snelham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E., & Yoon, D. H. In-Datacenter Performance Analysis of a Tensor Processing Unit. pages 1–17, 2017. URL `http://arxiv.org/abs/1704.04760`.

[94] Kaiser, E., Kutz, J. N., & Brunton, S. L. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2219), 2018. ISSN 1471-2946. doi:10.1098/rspa.2018.0335.

[95] Kellner, T. Extreme Measures: At 107 Meters, The World's Largest Wind Turbine Blade Is Longer Than A Football Field. Here's What It Looks Like, 2019. URL `https://www.ge.com/reports/extreme-measures-107-meters-worlds-largest-wind-turbine-blade-longer-football-field-heres-looks-like/`.

[96] Kost, C., Shammugam, S., Jülch, V., Nguyen, H.-t., & Schlegl, T. Levelized Cost of Electricity- Renewable Energy Technologies. (March), 2018.

[97] Kragh, K. A. & Hansen, M. H. Load alleviation of wind turbines by yaw misalignment. (April 2013):971–982, 2014. doi:10.1002/we.

[98] Kramak, B. Data Quality - The Top 5 Issues When Assessing Wind Plant Performance, 2016. URL `https://aws-dewi.ul.com/data-quality-the-top-5-issues-when-assessing-wind-plant-performance/`.

[99] Krenn, A., Stökl, A., Weber, N., Barup, S., Weidl, T., Hoffmann, A., Bredesen, R. E., Lannic, M., Müller, S., Stoffels, N., Hahm, T., Storck, F., & Lautenschlager, F. IEA Wind TCP Task 19 : International Recommendations for Ice Fall and Ice Throw Risk Assessments.

[100] Kröger, L., Frederik, J., Van Wingerden, J. W., Peinke, J., & Hölling, M. Generation of user defined turbulent inflow conditions by an active grid for validation experiments. *Journal of Physics: Conference Series*, 1037(5), 2018. ISSN 17426596. doi:10.1088/1742-6596/1037/5/052002.

[101] Lam, S. K., Pitrou, A., & Seibert, S. Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC - LLVM '15*, pages 1–6, 2015. doi:10.1145/2833157.2833162. URL `http://dl.acm.org/citation.cfm?doid=2833157.2833162`.

[102] Leble, V. & Barakos, G. Forced pitch motion of wind turbines. *Journal of Physics: Conference Series*, 753:022042, 2016. ISSN 1742-6588. doi:10.1088/1742-6596/753/2/022042. URL http://stacks.iop.org/1742-6596/753/i=2/a=022042?key=crossref.958beaecb1498a8032f12f1d84468c29.

[103] Lee, C. H. & Newman, J. N. Computation of wave effects using the panel method. Numerical Models in Fluid-Structure Interaction. *Preprint, WIT Press, Southampton, UK*, (1):1–41, 2004.

[104] Lee, H. G. & Park, J. S. Optimization of resonance-type fatigue testing for a full-scale wind turbine blade. *Wind Energy*, 19(2):371–380, 2016. ISSN 10991824. doi:10.1002/we.1837.

[105] Leishman, J. G. & Beddoes, T. S. A Semi-Empirical Model for Dynamic Stall. *Journal of the American Helicopter Society*, 34(3):3–17, 1989.

[106] Leishman, J. G. *Principles of helicopter aerodynamics*. Cambridge University Press, Cambridge, 2nd edition, 2006.

[107] Lennie, M. Anti-Icing for Wind turbines Conference Open Source Ice Fall Risk Analysis : Academia to Industry.

[108] Lennie, M., Pechlivanoglou, G., Marten, D., Nayeri, C., & Paschereit, O. A Review of Wind Turbine Polar Data and its Effect on Fatigue Loads Simulation Accuracy. In *ASME. Turbo Expo: Power for Land, Sea, and Air, Volume 9: Oil and Gas Applications; Supercritical CO2 Power Cycles; Wind Energy ()*, Dusseldorf, 2015. doi:10.1115/GT2015-43249.

[109] Lennie, M., Steenbuck, J., Noack, B. R., & Paschereit, C. O. Cartographing dynamic stall with machine learning. *Wind Energy Science Discussions*, 2019:1–31, 2019. doi:10.5194/wes-2019-36. URL https://www.wind-energ-sci-discuss.net/wes-2019-36/.

[110] Lennie, M. *Development of the QFEM Solver : The Development of Modal Analysis Code for Wind Turbine Blades in QBLADE*. PhD thesis, KTH Stockholm and Technical University of Berlin, 2013.

[111] Lennie, M., Pechlivanoglou, G., Marten, D., Nayeri, C. N., & Paschereit, C. O. GT2015-43249 : A review of wind turbine polar data and it's effect on fatigue loads simulation accuracy. *Proceedings of ASME Turbo Expo 2015: Turbine Technical Conference and Exposition GT2015*, (JUNE), 2015. doi:10.1115/GT2015-43249.

[112] Lennie, M., Bach, A., Pechlivanoglou, G., Nayeri, C., & Paschereit, C. O. The Unsteady Aerodynamic Response of an Airfoil with Microtabs and it's Implications for Aerodynamic

Damping. *34th Wind Energy Symposium*, (January):1–12, 2016. doi:10.2514/6.2016-1006. URL http://arc.aiaa.org/doi/10.2514/6.2016-1006.

[113] Lennie, M., Marten, D., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. Modern methods for investigating the stability of a pitching floating platform wind turbine. *Journal of Physics: Conference Series*, 753:082012, 2016. ISSN 1742-6588. doi:10.1088/1742-6596/753/8/082012. URL http://stacks.iop.org/1742-6596/753/i=8/a=082012?key=crossref.82fe28b0acc888f6d82a7d7133fc26b7.

[114] Lennie, M., Wendler, J., Pechlivanoglou, G., Nayeri, C., Paschereit, C. O., & Greenblatt, D. Development of a Partially Stochastic Unsteady Aerodynamics Model. *35th Wind Energy Symposium*, pages 1–14, 2017. doi:10.2514/6.2017-2002. URL http://arc.aiaa.org/doi/10.2514/6.2017-2002.

[115] Lennie, M., Marten, D., Pechlivanoglou, G., Paschereit, C. O., & Dominin, S. Simulating Wind Turbine Ice Throw: QBlade and Statistical Analysis. page V009T48A009, 2018. doi:10.1115/gt2018-76485.

[116] Lennie, M., Dominin, S., Marten, D., Pechlivanoglou, G., & Paschereit, C. O. Development of Ice Throw Model for Wind Turbine Simulation Software QBlade. In *AIAA Scitech 2019 Forum*, number January, pages 1–13, 2019. doi:10.2514/6.2019-1800.

[117] Lennie, M., Dominin, S., Marten, D., Pechlivanoglou, G., & Paschereit, C. O. Winterwind 2019 Importance sampling for ice throw in QBlade. In *Winterwind International Wind Energy Conference*, Umea, 2019. URL https://winterwind.se/presentations-from-winterwind-2019/.

[118] Leroy, V., Gilloteaux, J. C., Lynch, M., Babarit, A., & Ferrant, P. Impact of aerodynamic modeling on seakeeping performance of a floating horizontal axis wind turbine. *Wind Energy*, (December 2018):1–15, 2019. ISSN 10991824. doi:10.1002/we.2337.

[119] Li, D. S., Li, R. N., Wei, L. J., Wang, X. Y., Qiang, Y., & Li, Y. R. Comparison of the pressure distribution of a wind turbine blade based on field experiment and CFD. *IOP Conference Series: Materials Science and Engineering*, 52(5):052004, dec 2013. ISSN 1757-8981. doi:10.1088/1757-899X/52/5/052004. URL http://stacks.iop.org/1757-899X/52/i=5/a=052004?key=crossref.a67bac85fefb0c7b4843e2c031b313fb.

[120] Mackay, D. J. C. *Sustainable Energy - without the hot air. Version 3.5.2. November 3, 2008.* Cambridge, 1st edition, 2008. ISBN 978-0954452933. URL https://www.withouthotair.com/.

[121] Madsen, H., Fuglsang, P., Romblad, J., Enevoldsen, P., Laursen, J., Jensen, L., Bak, C., Paulsen, U. S., Gaunna, M., Sorensen, N. N., & Others, . *The DAN-AERO MW Experiments*. 2010. ISBN 9788755038097. doi:doi:10.2514/6.2010-645.

[122] Madsen, H. A., Özçakmak, Ö. S., Bak, C., Troldborg, N., Sørensen, N. N., & Sørensen, J. N. Transition characteristics measured on a 2MW 80m diameter wind turbine rotor in comparison with transition data from wind tunnel measurements. In *AIAA Scitech 2019 Forum*. AIAA, 2019. doi:10.2514/6.2019-0801. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2019-0801`.

[123] Madsen, P. H. Introduction to the IEC 61400-1 standard.

[124] Malcolm, D. J. WindPACT Turbine Rotor Design Study WindPACT Turbine Rotor Design Study. (April), 2006.

[125] Mann, J. The Spatial Structure of Neutral Atmospheric Surface-Layer Turbulence. *Journal of Fluid Mechanics*, 273:141–168, 1994. ISSN 14697645. doi:10.1017/S0022112094001886.

[126] Manohar, K., Brunton, B. W., Kutz, J. N., & Brunton, S. L. Data-Driven Sparse Sensor Placement for Reconstruction. (june), 2017. ISSN 1066033X. doi:10.1109/MCS.2018.2810460. URL `http://arxiv.org/abs/1701.07569{%}0Ahttp://dx.doi.org/10.1109/MCS.2018.2810460`.

[127] Manohar, K., Brunton, B. W., Kutz, J. N., & Brunton, S. L. Data-Driven Sparse Sensor Placement for Reconstruction: Demonstrating the Benefits of Exploiting Known Patterns. *IEEE Control Systems*, 38(3):63–86, 2018. ISSN 1066033X. doi:10.1109/MCS.2018.2810460.

[128] Manolesos, M., Sørensen, N. N., Troldborg, N., Florentie, L., Papadakis, G., & Voutsinas, S. Computing the flow past Vortex Generators: Comparison between RANS Simulations and Experiments. *Journal of Physics: Conference Series*, 753:022014, 2016. ISSN 1742-6588. doi:10.1088/1742-6596/753/2/022014. URL `http://stacks.iop.org/1742-6596/753/i=2/a=022014?key=crossref.ee12c1d237ffafdcdd45cfadc5523cf8`.

[129] Manolesos, M., Papadakis, G., & Voutsinas, S. Revisiting the assumptions and implementation details of the BAY model for vortex generator flows. *Renewable Energy*, 146:1249–1261, 2019. ISSN 09601481. doi:10.1016/j.renene.2019.07.063. URL `https://linkinghub.elsevier.com/retrieve/pii/S0960148119310833`.

[130] Manolesos, M. & Voutsinas, S. G. Study of a stall cell using stereo particle image velocimetry. *Physics of Fluids*, 26(4), 2014. ISSN 10897666. doi:10.1063/1.4869726.

[131] Manolesos, M., Papadakis, G., & Voutsinas, S. G. Experimental and computational analysis of stall cells on rectangular wings. (May 2013):939–955, 2014. doi:10.1002/we.

[132] Manolesos, M., Papadakis, G., & Voutsinas, S. G. An experimental and numerical investigation on the formation of stall-cells on airfoils. *Journal of Physics: Conference Series*, 555(1):012068, 2014. ISSN 1742-6588. doi:10.1088/1742-6596/555/1/012068. URL `http://stacks.iop.org/1742-6596/555/i=1/a=012068`.

[133] Mart\'\in~Abadi, , Ashish~Agarwal, , Paul~Barham, , Eugene~Brevdo, , Zhifeng~Chen, , Craig~Citro, , Greg~S.~Corrado, , Andy~Davis, , Jeffrey~Dean, , Matthieu~Devin, , Sanjay~Ghemawat, , Ian~Goodfellow, , Andrew~Harp, , Geoffrey~Irving, , Michael~Isard, , Jia, Y., Rafal~Jozefowicz, , Lukasz~Kaiser, , Manjunath~Kudlur, , Josh~Levenberg, , Dan~Mané, , Rajat~Monga, , Sherry~Moore, , Derek~Murray, , Chris~Olah, , Mike~Schuster, , Jonathon~Shlens, , Benoit~Steiner, , Ilya~Sutskever, , Kunal~Talwar, , Paul~Tucker, , Vincent~Vanhoucke, , Vijay~Vasudevan, , Fernanda~Viégas, , Oriol~Vinyals, , Pete~Warden, , Martin~Wattenberg, , Martin~Wicke, , Yuan~Yu, , & Xiaoqiang~Zheng, . {TensorFlow}: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL `http://tensorflow.org/`.

[134] Marten, D., Lennie, M., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. Implementation, Optimization, and Validation of a Nonlinear Lifting Line-Free Vortex Wake Module Within the Wind Turbine Simulation Code QBlade. *Journal of Engineering for Gas Turbines and Power*, 138(7):072601, 2015. ISSN 0742-4795. doi:10.1115/1.4031872. URL `http://www.scopus.com/inward/record.url?eid=2-s2.0-84949256983{&}partnerID=tZOtx3y1`.

[135] Marten, D., Lennie, M., Pechlivanoglou, G., Paschereit, C. O., Dy, N. V., Paraschivoiu, I., & Saeed, F. Validation and comparison of a newly developed aeroelastic design code for VAWT. In *AIAA SciTech Forum*, 2017. doi:10.2514/6.2017-0452.

[136] Marten, D., Lennie, M., Pechlivanoglou, G., Paschereit, C. O., Bianchini, A., Ferrara, G., & Ferrari, L. Benchmark of a Novel Aero-Elastic Simulation Code for Small Scale VAWT Analysis. *Journal of Engineering for Gas Turbines and Power*, 141(4):041014, 2018. ISSN 0742-4795. doi:10.1115/1.4041519.

[137] Marten, D., Lennie, M., Pechlivanoglou, G., Paschereit, C. O., Bianchini, A., Ferrara, G., & Ferrari, L. Benchmark of a Novel Aero-Elastic Simulation Code for Small Scale VAWT Analysis. *Journal of Engineering for Gas Turbines and Power*, 141(4):041014, 2018. ISSN 0742-4795. doi:10.1115/1.4041519.

[138] Masson-Delmotte, V., Zhai, P., Pörtner, H.-O., Roberts, D., Skea, J., Shukla, P., Pirani, A., Moufouma-Okia, W., Péan, C., Pidcock, R., Connors, S., Matthews, J., Chen, Y.,

Zhou, X., Gomis, M., Lonnoy, E., Maycock, T., Tignor, M., & Waterfield, T. IPCC, 2018: Summary for Policymakers. In Press, I., editor, *Global Warming of 1.5°C. An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change,*, pages 1–21. IPCC, 1st edition, 2018. URL `https://www.ipcc.ch/site/assets/uploads/sites/2/2019/05/SR15{_}SPM{_}version{_}report{_}HR.pdf`.

[139] Matha, D., Schlipf, M., Cordle, A., Pereira, R., & Jonkman, J. Challenges in Simulation of Aerodynamics , Hydrodynamics , and Mooring-Line Dynamics of Floating Offshore Wind Turbines. In *Offshore and Polar Engineering Conference*, volume 8, pages 421–428, Maui, Hawaii, 2011. URL `http://www.nrel.gov/docs/fy12osti/50544.pdf`.

[140] Mishnaevsky, L., Branner, K., Petersen, H. N., Beauson, J., McGugan, M., & Sørensen, B. F. Materials for wind turbine blades: An overview. *Materials*, 10(11):1–24, 2017. ISSN 19961944. doi:10.3390/ma10111285.

[141] Møller, M., Domagalski, P., & Sætran, L. R. Comparing Abnormalities in Onshore and Offshore Vertical Wind Profiles. *Wind Energy Science Discussions*, (July):1–33, 2019. URL `https://www.wind-energ-sci-discuss.net/wes-2019-40/wes-2019-40.pdf`.

[142] Moriarty, & Hansen, . AeroDyn Theory Manual. *Renewable Energy*, 15(January): 500–36313, 2005. ISSN 00664189. URL `http://www.nrel.gov/docs/fy05osti/36881.pdf`.

[143] Morison, J., O'Brien, M. P., Johnson, J., & Schaaf, S. The Force Exerted by Surface Waves on Piles. *Journal of Petroleum Technology*, 2(05):149–154, 1950. ISSN 0149-2136. doi:10.2118/950149-g.

[144] Murray, I. Slides: Advanced MCMC methods, 2006. URL `http://mlg.eng.cam.ac.uk/tutorials/06/im.pdf`.

[145] Nair, A. G., Yeh, C.-A., Kaiser, E., Noack, B. R., Brunton, S. L., & Taira, K. Cluster-based feedback control of turbulent post-stall separated flows. *Journal of Physics Fluid Dynamics*, (M):1–32, 2018. URL `http://arxiv.org/abs/1809.07220`.

[146] NASA, . Climate Time Machine, 2019. URL `https://climate.nasa.gov/interactives/climate-time-machine`.

[147] Nelson, J., Cairns, D., Riddle, T., & Workman, J. Composite Wind Turbine Blade Effects of Defects: Part B- Progressive Damage Modeling of Fiberglass/Epoxy Laminates with

Manufacturing Induced Flaws. *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference<BR>20th AIAA/ASME/AHS Adaptive Structures Conference<BR>14th AIAA*, apr 2012. doi:10.2514/6.2012-1421. URL `http://arc.aiaa.org/doi/abs/10.2514/6.2012-1421`.

[148] Nelson, J. W., Riddle, T. W., & Cairns, D. S. Characterization and Mechanical Testing of Manufacturing Defects Common to Composite Wind Turbine Blades. *Wind Energy Science Discussions*, (April):1–17, 2017. ISSN 2366-7621. doi:10.5194/wes-2017-13. URL `http://www.wind-energ-sci-discuss.net/wes-2017-13/`.

[149] Nelson, J. W., Riddle, T. W., & Cairns, D. S. Progressive Damage Modeling of Fiberglass/Epoxy Composites with Manufacturing Induced Waves Common to Wind Turbine Blades. *Wind Energy Science Discussions*, (April):1–17, 2017. ISSN 2366-7621. doi:10.5194/wes-2017-15. URL `http://www.wind-energ-sci-discuss.net/wes-2017-15/`.

[150] Nelson, J. W., Riddle, T. W., & Cairns, D. S. Characterization and Mechanical Testing of Manufacturing Defects Common to Composite Wind Turbine Blades. *Wind Energy Science Discussions*, pages 1–17, 2017. doi:10.5194/wes-2017-13.

[151] Newman, J. N. Progress in wave load computations on offshore structures. *Omae*, 2004. URL `http://www.wamit.com/Publications/newman{_}omae04.pdf`.

[152] Ng, A., Coates, A., Saxe, A., Maas, A., Manning, C., Ngiam, J., Socher, R., & Le, K. Machine Learning and AI via Brain simulations, 2013. ISSN 0163-4437.

[153] Nielsen, P. H., Berring, P., Pavese, C., & Branner, K. Rotor blade full-scale fatigue testing technology and research Department of Wind Energy E Report 2013. Technical report, DTU Wind Energy, Roskilde, 2013.

[154] O'Dea, M. L. & Guessous, L. Simulation of Wind Turbine Flow Using the Actuator Line Method in NEK5000. page V07BT09A052, 2016. doi:10.1115/imece2015-53671.

[155] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer, A. Automatic Differentiation in {PyTorch}. In *NIPS Autodiff Workshop*, 2017.

[156] Pechlivanoglou, G., Fischer, J., Eisele, O., Vey, S., Nayeri, C., & Paschereit, C. Development of a Medium Scale Research Hawt for Inflow and Aerodynamic Research in the TU Berlin Wind Tunnel. In *Dewek*, Bremen, 2015.

[157] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau,

D., Brucher, M., Perrot, M., & Duchesnay, É. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2012. ISSN 15324435. doi:10.1007/s13398-014-0173-7.2. URL http://dl.acm.org/citation.cfm?id=2078195{%}5Cnhttp://arxiv.org/abs/1201.0490.

[158] Pereira, R., Schepers, G., & Pavel, M. D. Validation of the Beddoes Leishman dynamic stall model for horizontal axis wind turbines using MEXICO?data. *Wind Energy*, 16(2): 207–219, 2013. ISSN 1099-1824. doi:10.1002/we.541. URL http://dx.doi.org/10.1002/we.541.

[159] Perez-becker, S., Saverin, J., Marten, D., Pechlivanoglou, G., Paschereit, C. O., & Hermann, F. Investigations on the Fatigue Load Reduction Potential of Advanced Control Strategies for Multi-MW Wind Turbines Using a Free Vortex Wake Model. In *Proceedings of ASME Turbo Expo 2018 Turbomachinery Technical Conference and Exposition*, pages 1–11, 2018.

[160] Petersen, J. T., Madsen, H. A., & Rasmussen, F. Dynamic Stall and Aerodynamic Damping. 121(August 1999), 1999.

[161] Pires, O., Munduate, X., Boorsma, K., Ceyhan Yilmaz, O., Aa Madsen, H., & Timmer, W. A. Experimental investigation of Surface Roughness effects and Transition on Wind Turbine performance. In *The Science of Making Torque from Wind (TORQUE 2018)*, volume 1037, Milan, 2018. Journal of Physics: Conference Series. doi:10.1088/1742-6596/1037/5/052018.

[162] Pires, O., Munduate, X., Boorsma, K., Yilmaz, O. C., Madsen, H. A., & W.A.Timmer, . Experimental investigation of Surface Roughness effects and Transition on Wind Turbine performance. *The Science of Making Torque from Wind (TORQUE 2018)*, 1037, 2018. doi:10.1088/1742-6596/1037/5/052018. URL https://iopscience.iop.org/article/10.1088/1742-6596/1037/5/052018/pdf.

[163] Pitt, D. M. & Peters, D. A. Rotor Dynamic Inflow Derivatives and Time Constants from Various Inflow Models. In *9th European Rotorcraft Forum*, pages 1–24. US Army Troop Support and Aviation Materiel Readiness Command, St. Louis Missouri, 1983. URL http://www.scopus.com/inward/record.url?eid=2-s2.0-0019681684{&}partnerID=tZOtx3y1.

[164] Quaeghebeur, E. & Zaaijer, M. B. How to improve your metocean datasets. (July):1–33, 2019.

[165] Ramos-García, N., Spietz, H. J., Sørensen, J. N., & Walther, J. H. Vortex simulations of wind turbines operating in atmospheric conditions using a prescribed velocity-

vorticity boundary layer model. *Wind Energy*, 21(11):1216–1231, 2018. ISSN 10991824. doi:10.1002/we.2225.

[166] Raschka, S. *Python Machine Learning*. Packt Publishing, 1st edition, 2015. ISBN 1783555130, 9781783555130.

[167] Reid, S. ML/DL for Non-Stationary Time Series Analysis in Financial Markets and Beyond with Stuart Reid, 2018. URL `https://twimlai.com/twiml-talk-203-ml-dl-for-non-stationary-time-series-analysis-in-financial-markets-and-beyond-with-stuart-reid/`.

[168] Reuss Ramsay, R., Hoffmann, M. J., & Gregorek, G. M. Effects of Grit Roughness and Pitch Oscillations on the NACA 4415 Airfoil. *Nrel/Tp-442-7817*, (December):152, 1995. doi:10.2172/266691.

[169] Richard, C. German regulator cuts wind feed-in tariff again, 2018. URL `https://www.windpowermonthly.com/article/1458353/german-regulator-cuts-wind-feed-in-tariff-again`.

[170] Riddle, T. W., Nelson, J. W., & Cairns, D. S. Probabilistic Design of Wind Turbine Blades with Treatment of Manufacturing Defects as Uncertainty Variables in a Framework. *Wind Energy Science Discussions*, 1(April):1–21, 2017. doi:10.5194/wes-2017-14.

[171] Riddle, T. W., Nelson, J. W., & Cairns, D. S. Effects of defects in composite wind turbine blades – Part 3: A framework for treating defects as uncertainty variables for blade analysis. *Wind Energy Science*, 3(1):107–120, 2018. ISSN 2366-7451. doi:10.5194/wes-3-107-2018. URL `https://www.wind-energ-sci.net/3/107/2018/`.

[172] Riziotis, V. A., Voutsinas, S. G., Politis, E. S., & Chaviaropoulos, P. K. Aeroelastic stability of wind turbines: The problem, the methods and the issues. *Wind Energy*, 7(4): 373–392, 2004. ISSN 10954244. doi:10.1002/we.133.

[173] Rumsey, C. L. & Nishino, T. Numerical study comparing RANS and LES approaches on a circulation control airfoil. *International Journal of Heat and Fluid Flow*, 32(5): 847–864, 2011. ISSN 0142727X. doi:10.1016/j.ijheatfluidflow.2011.06.011.

[174] Salo, E., Mcmillan, D., & Connor, R. Extracting value from free text maintenance records – an active learning approach. In *Wind Energy Science Conference*, Cork, 2019.

[175] Saverin, J., Marten, D., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. GT2016-56290 Coupling of an unsteady lifting line free vortex wake code to the aeroelastic HAWT simulation suite FAST. In *Proceedings of ASME Turbo Expo 2016*, Seoul, South Korea, 2016.

[176] Sayed, M., Lutz, T., Krämer, E., Shayegan, S., & Wüchner, R. Aeroelastic analysis of 10 MW wind turbine using CFD–CSD explicit FSI-coupling approach. *Journal of Fluids and Structures*, 87:354–377, 2019. ISSN 10958622. doi:10.1016/j.jfluidstructs.2019.03.023. URL https://doi.org/10.1016/j.jfluidstructs.2019.03.023.

[177] Schepers, J. G., Boorsma, K., Gomez-Iradi, S., Schaffarczyk, P., Madsen, H. A., Sørensen, N. N., Shen, W. Z., Lutz, T., Schulz, C., Herraez, I., & Schreck, S. Final report of IEA Task 29, Mexnext (Phase 2). *Researchgate.Net*, (December), 2014. URL http://www.researchgate.net/profile/Daniel{_}Micallef/publication/272183785{_}Final{_}report{_}of{_}IEA{_}Task{_}29{_}Mexnext{_}(Phase{_}1){_}Analysis{_}of{_}Mexico{_}wind{_}tunnel{_}measurements/links/54df3a550cf2953c22b16a4d.pdf.

[178] Schito, P. & Zasso, A. Wind turbine wakes: An actuator line approach. In *European Wind Energy Conference and Exhibition, EWEC 2013*, volume 2, 2013. ISBN 9781632663146.

[179] SethBling, . MarI/O - Machine Learning for Video Games, 2015. URL www.youtube.com/watch?v=qv6UVOQ0F44.

[180] Shalizi, C. Nonlinear Dimensionality Reduction I : Local Linear Embedding Why We Need Nonlinear Dimensionality Reduction. In *Data Mining 36-350*. Carnegie Mellon, 2009. doi:10.1126/science.290.5500.2323.

[181] Shihavuddin, A. S., Chen, X., Fedorov, V., Christensen, A. N., Riis, N. A. B., Branner, K., Dahl, A. B., & Paulsen, R. R. Wind turbine surface damage detection by deep learning aided drone inspection analysis. *Energies*, 12(4):1–15, 2019. ISSN 19961073. doi:10.3390/en12040676.

[182] Silver, N. *The Signal and the Noise: Why So Many Predictions Fail-but Some Don't*. Penguin Publishing Group, 1st edition, 2012. ISBN 9781101595954. URL https://books.google.de/books?id=SI-VqAT4{_}hYC.

[183] Skrzypinski, W. *Analysis and modeling of unsteady aerodynamics with application to wind turbine blade vibration at standstill conditions*. PhD thesis, DTU, 2012.

[184] Sørensen, J. N., Mikkelsen, R. F., Dan, S., Ivanell, S., Sarmast, S., & Andersen, S. J. Simulation of wind turbine wakes using the actuator line technique. *Philosophical Transactions, Royal Society*, 373, 2015. doi:http://dx.doi.org/10.1098/rsta.2014.0071.

[185] Sorensen, J. N. & Shen, W. Z. Numerical Modeling of Wind Turbine Wakes. *Journal of Fluids Engineering*, 124(2):393, 2002. ISSN 00982202. doi:10.1115/1.1471361.

[186] Sørensen, J. D. & Toft, H. S. Probabilistic Design of Wind Turbines. *Energies*, 3(2): 241–257, 2010. ISSN 19961073. doi:10.3390/en3020241.

[187] Stanley, K. O. & Miikkulainen, R. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*, 10(2):99–127, 2002. URL `http://mitpress.mit.edu/journals{%}5Cnhttp://mitpress.mit.edu/journals`.

[188] Stevens, R. J. & Meneveau, C. Flow Structure and Turbulence in Wind Farms. *Annual Review of Fluid Mechanics*, 49(1):311–339, 2016. ISSN 0066-4189. doi:10.1146/annurev-fluid-010816-060206.

[189] Stewart, G. & Muskulus, M. A Review and Comparison of Floating Offshore Wind Turbine Model Experiments. *Energy Procedia*, 94(1876):227–231, 2016. ISSN 18766102. doi:10.1016/j.egypro.2016.09.228. URL `http://dx.doi.org/10.1016/j.egypro.2016.09.228`.

[190] Strangfeld, C. *Active control of trailing vortices by means of long- and short-wavelength actuation*. Doctor thesis, Technical University of Berlin, 2015.

[191] Sutskever, I. Recent Advances in Deep Learning and AI from OpenAI, 2018. URL `https://www.youtube.com/watch?v=ElyFDUab30A{&}feature=youtu.be`.

[192] Tabor, G. R. & Baba-Ahmadi, M. H. Inlet conditions for large eddy simulation: A review, 2010. ISSN 00457930.

[193] Taira, K., Brunton, S. L., Dawson, S. T. M., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V., & Ukeiley, L. S. Modal Analysis of Fluid Flows: An Overview. *AIAA Journal*, 55(12):1–46, 2017. URL `http://arxiv.org/abs/1702.01453`.

[194] Tang, D., Dowell, E. H., Juang, J.-N., & Kholodar, D. System Identification and Proper Orthogonal Decomposition Method Applied to Unsteady Aerodynamics. *AIAA Journal*, 39(8):1569–1576, 2001. ISSN 0001-1452. doi:10.2514/2.1482. URL `http://doi.aiaa.org/10.2514/2.1482`.

[195] Theodorsen, T. General theory of aerodynamic instability and the mechanism of flutter, 1935. ISSN 1098-6596.

[196] Timmer, W. & Rooij, R. V. Summary of the Delft University wind turbine dedicated airfoils. In *AIAA-2003-0352*, 2003. URL `http://solarenergyengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1456892`.

[197] Timmer, W. & van Rooij, R. Some aspects of high angle-of-attack flow on airfoils for wind turbine application. *EWEC 2001, Copenhagen, Denmark*, pages 4–7, 2001. URL http://scholar.google.com/scholar?hl=en{&}btnG=Search{&}q=intitle:No+Title{#}0http://www.lr.tudelft.nl/fileadmin/Faculteit/LR/Organisatie/Afdelingen{_}en{_}Leerstoelen/Afdeling{_}AEWE/Wind{_}Energy/Research/Publications/Publications{_}2001/doc/TimmerEWEC2001.pdf.

[198] Tindl, A. Ice detection on rotor blades – Rotor Ice Control. URL https://www.fos4x.de/blog/Wissensdatenbank/ice-detection-on-rotor-blades/.

[199] Troldborg, N., Bak, C., Özçakmak, Ö. S., Sørensen, J. N., Sørensen, N. N., & Madsen, H. A. Correction: Transition characteristics measured on a 2MW 80m diameter wind turbine rotor in comparison with transition data from wind tunnel measurements. (January), 2019. doi:10.2514/6.2019-0801.c1.

[200] TwimlAI, . Trends in Deep Learning with Jeremy Howard, 2018. URL https://twimlai.com/twiml-talk-214-trends-in-deep-learning-with-jeremy-howard/.

[201] UpWind, . UpWind Design limits and solutions for very large wind turbines. Technical report, UpWind, 2011.

[202] van der Wall, B. G. *The Influence of Variable Flow Velocity on Unsteady Airfoil Behavior*. Masters thesis, DLR-Forschungsbericht, 1992. URL http://elib.dlr.de/20308/.

[203] van Kuik, G. a. M., Peinke, J., Nijssen, R., Lekou, D., Mann, J., Sørensen, J. N., Ferreira, C., van Wingerden, J. W., Schlipf, D., Gebraad, P., Polinder, H., Abrahamsen, a., van Bussel, G. J. W., Sørensen, J. D., Tavner, P., Bottasso, C. L., Muskulus, M., Matha, D., Lindeboom, H. J., Degraer, S., Kramer, O., Lehnhoff, S., Sonnenschein, M., Sørensen, P. E., Künneke, R. W., Morthorst, P. E., & Skytte, K. Long-term research challenges in wind energy – a research agenda by the European Academy of Wind Energy. *Wind Energy Science*, 1(1):1–39, 2016. ISSN 2366-7451. doi:10.5194/wes-1-1-2016. URL http://www.wind-energ-sci.net/1/1/2016/wes-1-1-2016.pdf{%}5Cnhttp://www.wind-energ-sci.net/1/1/2016/.

[204] Various Contributors, . Hyperparameter (machine learning), jun 2019. URL https://en.wikipedia.org/wiki/Hyperparameter{_}(machine{_}learning).

[205] Veers, P. S. Three-Dimensional Wind Simulation. Technical report, Sandia, 1988. URL https://prod.sandia.gov/techlib-noauth/access-control.cgi/1988/880152.pdf.

[206] Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W. M., Dudzik, A., Huang, A., Georgiev, P., Powell, R., Ewalds, T., Horgan, D., Kroiss, M., Danihelka, I., Agapiou, J., Oh, J., Dalibard, V., Choi, D., Sifre, L., Sulsky, Y., Vezhnevets, S., Molloy, J., Cai, T., Budden, D., Paine, T., Gulcehre, C., Wang, Z., Pfaff, T., Pohlen, T., Wu, Y., Yogatama, D., Cohen, J., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Apps, C., Kavukcuoglu, K., Hassabis, D., & Silver, D. AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. \url{https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/}, 2019.

[207] Wagner, H. Uber die Entstehung des dynamischen Auftriebes von Tragflugeln. 1925.

[208] Wang, Z., Chen, J., & Hoi, S. C. H. Deep Learning for Image Super-resolution: A Survey. pages 1–23, 2019. URL `http://arxiv.org/abs/1902.06068`.

[209] Wendler, J., Marten, D., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. GT2016-57184 An Unsteady Aerodynamics Model for Lifting Line Free Vortex Wake Simulations of HAWT and VAWT in QBlade. In *Proceedings of ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition*, Seoul, South Korea, 2016.

[210] Xu, B. F., Wang, T. G., Yuan, Y., & Cao, J. F. Unsteady aerodynamic analysis for offshore floating wind turbines under different wind conditions. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 373(2035), feb 2015. ISSN 1364-503X. doi:10.1098/rsta.2014.0080. URL `http://www.ncbi.nlm.nih.gov/pubmed/25583859`.

[211] Xu, M., Duan, L.-Y., Cai, J., Chia, L.-T., Xu, C., & Tian, Q. HMM-Based Audio Keyword Generation. Technical report, Nanyang Technological University, Singapore, 2004.

[212] Yuan, B., Wang, C., Jiang, F., Long, M., Yu, P. S., & Liu, Y. WaveletFCNN: A Deep Time Series Classification Model for Wind Turbine Blade Icing Detection. *CoRR*, abs/1902.0, 2019. URL `http://arxiv.org/abs/1902.05625`.

[213] Zhao, H., Liu, H., Hu, W., & Yan, X. Anomaly detection and fault analysis of wind turbine components based on deep learning network. *Renewable Energy*, 127:825–834, 2018. ISSN 18790682. doi:10.1016/j.renene.2018.05.024.

[214] Zhong, H., Du, P., Tang, F., & Wang, L. Lagrangian dynamic large-eddy simulation of wind turbine near wakes combined with an actuator line method. *Applied Energy*, 144: 224–233, 2015. ISSN 03062619. doi:10.1016/j.apenergy.2015.01.082.

# Associated Publications

**Lennie, M**., Steenbuck, J., Noack, B., & Paschereit, C. O. (2019). Dynamic stall means are meaningless. In Wind Energy Science Conference. Cork.

**Lennie, M**., Marten, D., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2016). Modern methods for investigating the stability of a pitching floating platform wind turbine. Journal of Physics: Conference Series, 753, 082012. https://doi.org/10.1088/1742-6596/753/8/082012

**Lennie, M**., Dominin, S., Marten, D., Pechlivanoglou, G., & Paschereit, C. O. (2019). Development of Ice Throw Model for Wind Turbine Simulation Software QBlade, (January), 1–13. https://doi.org/10.2514/6.2019-1800

**Lennie, M**., Bach, A., Pechlivanoglou, G., Nayeri, C., & Paschereit, C. O. (2016). The Unsteady Aerodynamic Response of an Airfoil with Microtabs and it's Implications for Aerodynamic Damping. 34th Wind Energy Symposium, (January), 1–12. https://doi.org/10.2514/6.2016-1006

Castro, O., **Lennie, M**., Branner, K., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2015). Comparing fatigue life estimations of composite wind turbine blades using different fatigue analysis tools. ICCM20 - 20th International Conference on Composite Materials, (July), 19–24.

**Lennie, M**. (2013). Development of the QFEM Solver : The Development of Modal Analysis Code for Wind Turbine Blades in QBLADE. KTH Stockholm - Technical University of Berlin.

**Lennie, M**., Wendler, J., Pechlivanoglou, G., Nayeri, C., Paschereit, C. O., & Greenblatt, D. (2017). Development of a Partially Stochastic Unsteady Aerodynamics Model. 35th Wind Energy Symposium, 1–14. https://doi.org/10.2514/6.2017-2002

Castro, O., **Lennie, M**., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2015). The Use of a New Fatigue Tool (ALBdeS) to Analyse the Effects of Vortex Generators on Wind Turbines. Retrieved from http://dx.doi.org/10.1115/GT2015-43198

**Lennie, M**., Pechlivanoglou, G., Marten, D., Nayeri, C. N., & Paschereit, C. O. (2015). GT2015-43249 : A review of wind turbine polar data and it's effect on fatigue loads simulation accuracy. Proceedings of ASME Turbo Expo 2015: Turbine Technical Conference and Exposition GT2015, (JUNE). https://doi.org/10.1115/GT2015-43249

Marten, D., **Lennie, M**., Pechlivanoglou, G., Paschereit, C. O., Dy, N. V., Paraschivoiu, I., & Saeed, F. (2017). Validation and comparison of a newly developed aeroelastic design code for VAWT. In AIAA SciTech Forum. https://doi.org/10.2514/6.2017-0452

Bach, A., **Lennie, M.**, Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2014). Finite micro-tab system for load control on a wind turbine. Journal of Physics: Conference Series, 524, 012082. https://doi.org/10.1088/1742-6596/524/1/012082

Marten, D., **Lennie, M.**, Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2015). Implementation, Optimization, and Validation of a Nonlinear Lifting Line-Free Vortex Wake Module Within the Wind Turbine Simulation Code QBlade. Journal of Engineering for Gas Turbines and Power, 138(7), 072601. https://doi.org/10.1115/1.4031872

**Lennie, M.**, Marten, D., Pechlivanoglou, G., Paschereit, C. O., & Dominin, S. (2018). Simulating Wind Turbine Ice Throw: QBlade and Statistical Analysis, V009T48A009. https://doi.org/10.1115/gt2018-76485

Marten, D., **Lennie, M.**, Pechlivanoglou, G., Paschereit, C. O., Bianchini, A., Ferrara, G., & Ferrari, L. (2018). Benchmark of a Novel Aero-Elastic Simulation Code for Small Scale VAWT Analysis. Journal of Engineering for Gas Turbines and Power, 141(4), 041014. https://doi.org/10.1115/1.4041519

Marten D, **Lennie, M.**, Pechlivanoglou G, et al. Benchmark of a Novel Aero-Elastic Simulation Code for Small Scale VAWT Analysis. ASME. Turbo Expo: Power for Land, Sea, and Air, Volume 9: Oil and Gas Applications; Supercritical CO2 Power Cycles; Wind Energy ():V009T48A006. doi:10.1115/GT2018-75922.

**Lennie, M.**, Marten D, Pechlivanoglou G, Nayeri C, Paschereit C. Development and Validation of a Modal Analysis Code for Wind Turbine Blades. ASME. Turbo Expo: Power for Land, Sea, and Air, Volume 3B: Oil and Gas Applications; Organic Rankine Cycle Power Systems; Supercritical CO2 Power Cycles; Wind Energy ():V03BT46A031. doi:10.1115/GT2014-27151.

H. Spiegelberg, **Lennie, M**. and G. Pechlivanoglou Passive load reduction in wind turbine blades with an adaptive camber airfoil DEWEK2015, 2015

Platzer, M. F., **Lennie, M.**, & Vogt, D. M. (2013). Analysis of the conversion of ocean wind power into hydrogen. In Proceedings of the World Renewable Energy Congress, Murdoch, Australia (Vol. 14).

**Lennie, M.**, Dominin, S., Marten, D., Pechlivanoglou, G., & Paschereit, C. O. (2019). Winterwind 2019 Importance sampling for ice throw in QBlade. In Winterwind International Wind Energy Conference. Umea. Retrieved from https://winterwind.se/presentations-from-winterwind-2019/

Marten, D., **Lennie, M.**, Pechlivanoglou, G., Paschereit, C. O., Bianchini, A., Ferrara, G., & Ferrari, L. (2018). Benchmark of a Novel Aero-Elastic Simulation Code for Small Scale VAWT Analysis. Journal of Engineering for Gas Turbines and Power, 141(4), 041014. https://doi.org/10.1115/1.4041519