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
[2]LIMSI, CNRS, Université Paris-Saclay, Bât 507, rue du Belvédère, Campus Universitaire, F-91403 Orsay, France

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

Abstract. Airfoil stall is bad for wind turbines. 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 it's low pressure core causing a lift spike and moment dump. 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/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 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.

Keywords. Wind Energy, Machine Learning, Unsteady Aerodynamics, Dynamic Time Warping, Clustering, Deep Learning, Data Science, Multi-Dimensional Scaling, Dynamic Stall, Dynamic Stall Vortex

1 Introduction

15   Beyond small angles of attack, airfoil boundary layers have to contend with strong adverse pressure gradients. When the boundary layer doesn't have enough momentum, a flow reversal occurs and eventually the flow separates from the surface of the airfoil (Abbott and von Doenhoff, 2012). 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 task. Even in the age of Computational Fluid Dynamics (CFD), attempts to simulate stall with Unsteady Reynolds Averaged Navier-

20   Stokes (URANS) have not yet yielded good quality results (Stangfeld et al., 2015; Rumsey, 2008; Rumsey and Nishino, 2011). Large Eddy Simulations (LES) show promise but are still too computationally expensive to be used as an ordinary design and analysis tool (Rumsey and Nishino, 2011). In practice, in the wind industry, semi-empirical 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, the key questions are: What information is lost?

25   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 understand the myriad of terms used within literature;

    1. A flow reversal in the boundary layer resulting in the stream-wise streamline no longer following the surface of the airfoil (Abbott and von Doenhoff, 2012). The region of flow reversal will usually have a neutral pressure.

    2. The presence of instabilities, such as shear layer instabilities or wake mode (vortex shedding) instabilities (Hudy and
30   Naguib, 2007). 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
35   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 von Doenhoff, 2012). The vertical size of the viscous region will be in 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 by-passed rapidly (Leishman, 2006). The separated region won't contribute to the lift implying a smooth roll off of
40   the lift, increase of drag, and a nose up moment. Even on a stationary airfoil, the boundaries of the separated region will be unsteady (Mulleners and Rütten, 2016) and will vary along the span.

    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 von Doenhoff, 2012).
45   Here the airfoil leading edge geometry is critical as a tight radius will cause a stronger adverse pressure gradient. Even though the stall occurs the 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 in 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 von Doenhoff, 2012) or a re-thickening of the airfoil e.g. wind turbine airfoils can have dents due to manufacturing (Madsen et al., 2019). This phenomenon is call a separation bubble. Bubbles are a sensitive phenomena 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 on 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 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 hasn't 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 to be variable between cycles of pitching, then it follows that the aeroelastic stability of the airfoil will also be variable between cycles.

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, 1981); Leishman (2002); Carr (1987) and more modern experimental works can be found in Granlund et al. (2014); Mulleners and Raffel (2013); Mulleners et al. (2012); Mulleners and Raffel (2012); Muller-Vahl et al. (2017); Müller-Vahl et al. (2015); Stangfeld et al. (2015); Balduzzi et al. (2019); Holst et al. (2019)). For the discussion here, it is sufficient to note that 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. There are complex instabilities that lead to stall being variable in both space and time.

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 it took measurements of 50 cycles of a pitching airfoil undergoing dynamic stall to ensure convergence of the lift. Only more recently have researchers have 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 Karman shedding pattern or a stable dead water zone (Mulleners and Rütten, 2016). The separation pattern fluctuates unreliably and when vorticity is present, the vortex convection speed is also variable.

Experimental data from Manolesos serves as a detailed reminder that stall happens in 3 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 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. (2018); Vey et al. (2014)). Wind turbines uniquely experience very high angles of attack; where the spatial patterns complicate further (Skrzypi et al., 2014; Skrzypinski, 2012; Gaunaa et al., 2016; Lennie et al., 2018). The picture that should be now 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? It should be clear given the discussion so far that simple averaging or even phase averaging will remove important data (Riches et al., 2018). The cycle to cycle variations and outliers are an important part of the dataset and shouldn't be smeared out. Manolesos (2014) suggested conditional averaging to produce better airfoil polar diagrams. Mulleners and Rütten (2016) also performed a kind of conditional averaging using the orbits of POD coordinates displayed onto recurrence plots. Furthermore, Holst et al. (2019) also suggests a binning approach, especially when considering very deep stall. Conditional averaging is an interesting approach, but the important question becomes, what condition is appropriate to split the data?

Fluid dynamics has always been a natural case for dimensionality reduction. In particular, there is abundant literature using Singular Value Decomposition (SVD) based methods such as POD/PCA (Taira et al., 2017), DMD (Schmid, 2010; Kutz et al., 2015; Brunton et al., 2015), 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/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, 2016), 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 in Christopher Olah's website (Olah, 2019). In this paper, we will show how Multi-Dimensional Scaling (MDS) (Borg and Groenen, 2007) and clustering (Maimon and Rokach, 2006) can be used as a reliable analysis technique for airfoil stall. Nair et al. (2018) 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. Both Loiseau et al. (2018) and Ehlert et al. (2019) have both demonstrated that Local Linear Embedding (LLE), a neighborhood type method, can create a sparse representation of the system.

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 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. (2018). The difference in approach comes from intended application.

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. 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. Furthermore, the concept of transfer learning exploits the fact that once a model has been trained for one task, it can be easily re-molded 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 and Others, 2019)). 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 challenge to extract this feature automatically, however 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).

2 Experimental Data

The analyses shown in the rest of this paper relies on two existing datasets. 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 set ups.

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 $75\,\text{kW}$ backward bladed radial blower. The test section is depicted in Figure 1 and is $610\,\text{mm}$ per $1004\,\text{mm}$. 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-side (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 $347\,\text{mm}$ and a span of $610\,\text{mm}$. 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".

The wind tunnel data covers a comprehensive collection of experiments with varying boundary conditions. It ranges from static baseline investigations over oscillating pitching and variation of free stream velocity (and a combination of both). In order to manipulate the boundary layer, blowing was added. One peculiarity of this data set 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. The water tank dimensions are $250\,\text{m}$ length, a width of $8.1\,\text{m}$, and an average depth of about $4.8\,\text{m}$. A carriage runs on rails, towing a rig (and the model) through the water with a maximum speed of $12.5\,\text{m\,s}^{-1}$. On it the complete measuring system is installed. The rig consists of to side plates with a length of $1.25\,\text{m}$, a height of $1\,\text{m}$, and a thickness of $0.035\,\text{m}$ prohibiting lateral flow around the model. In between the side plates, the model, with a span size up to $1\,\text{m}$ can be inserted at arbitrary angles of attacks. The model resembles a flat plate with an elliptical nose and blunt trailing edge. It has a span of $0.95\,\text{m}$, $0.5\,\text{m}$ chord a thickness of $0.03\,\text{m}$.
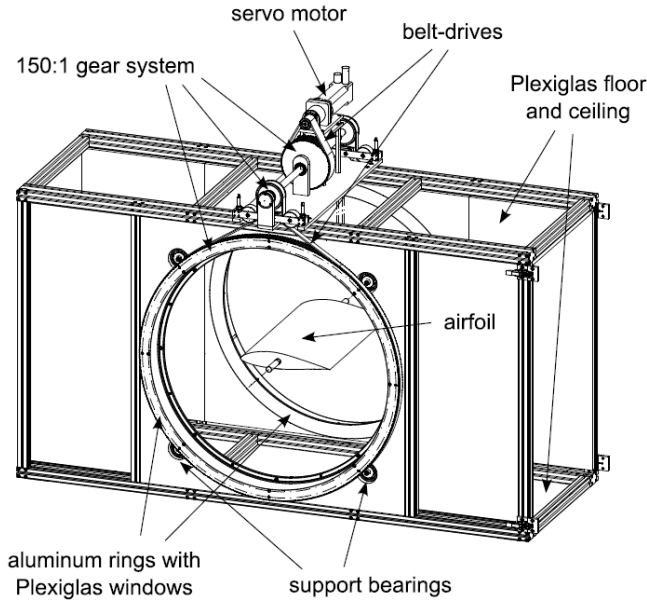
Figure 1. View of the test section showing the pitching mechanism and the approximate location of the airfoil model. From Müller-Vahl (2015).

The surface is covered in aluminum and 12 pressure ports are inserted at the specified locations in Figure 2. The airfoil model
180 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).
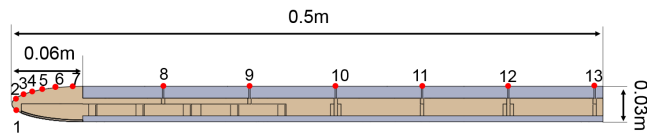
Figure 2. Cross section of the mounted flat plate. Red dots indicate position of pressure sensors. From Jentzsch et al. (2019).

3   Machine Learning Approaches

3.1   Dynamic time warping, clustering and multi-dimensional scaling

Before being able to compare repetitions of the experiments, we have to have a distance metric in order to numerically assess
185 similarity. Dynamic time warping is a distance measurement that allows for squashing and stretching of the time series in order to reach a best fit. The simplest interpretation is that Euclidean distance measures directly vertical between the time series and that conversely DTW the measurement is allowed to have a time component to find the lowest possible distance (Figure 5). In

practice, it's comparable to taking a winding path through a grid where each box corresponds to a time step from the two paths being compared (see Figure 5). The general rule of thumb is that a small amount of warping is a good thing, a lot can end up
190 distorting reality. Therefore, DTW algorithms are usually implemented with either global or local constraints, these constraints have a bonus of increasing the computational efficiency.

A useful extension to the DTW algorithm creates a composite of multiple time series called a centroid (see Figure 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
195 across different cycles of the experiments (see Figure 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.
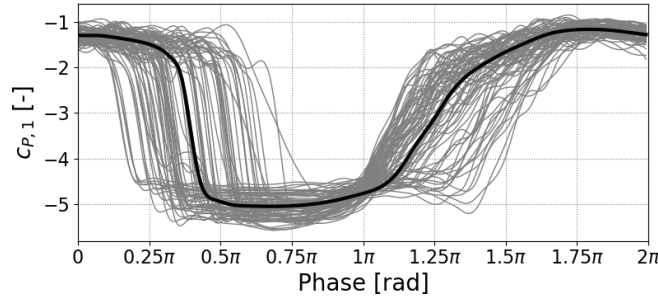
Figure 3. soft-DTW centroid for clustered time series with strong phase jitter.

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 that the algorithm is $O(N^2)$. Two steps
200 were taken to scale the process; firstly the data was down-sampled 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, 1-2 workers (nodes) would process a single experiment within a few minutes. A combination of parallelization and downsampling
205 was used in this study.

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 is, in addition to downsampling, filtered. Dynamic Time Warping is noise sensitive as the algorithm shifts and bends the time series
210 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 detailed in the related work from Steenbuck (2019).
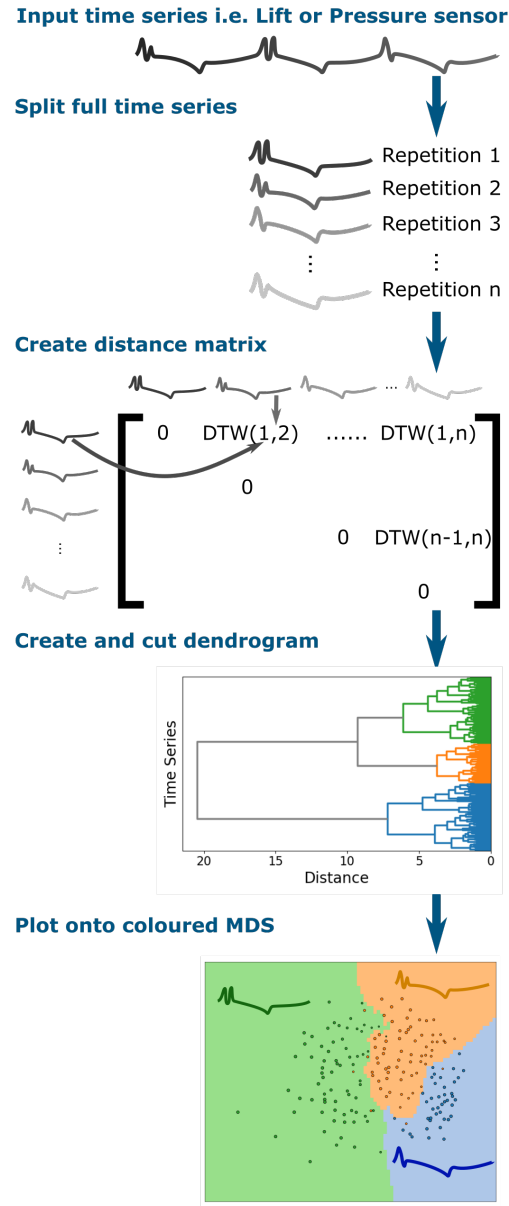
Figure 4. Time series clustering algorithm

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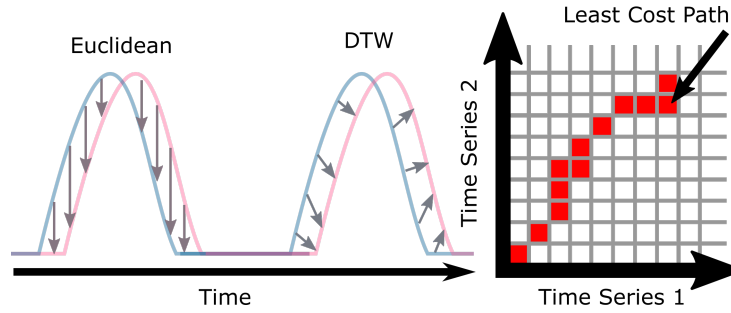
Figure 5. Euclidean distance vs DTW distance between two time series

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, the simplest of these parameters is the number of clusters. A

220   first estimation about a reasonable number can be made from inspecting the dendogram 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. The silhouette score is calculated by comparing the distance from a data element to its own cluster centre to the distance to the centre of the closest neighboring cluster. Large scores correspond to better clusters.
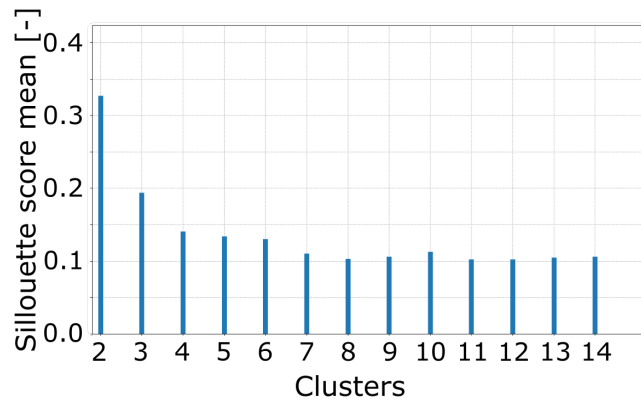
Figure 6. Mean of silhouette scores per cluster number.

      For this application hierarchical clustering turned out to produce groups that were physically meaningful and shared features.

225   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 Figure 4. 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 scipys (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

230   that other basic algorithms such as KMeans tended to perform poorly for this data.

Another way of presenting the data is to use Multidimensional Scaling (MDS) (Borg and Groenen, 2007). 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 reveals an underlying structure. We can then color each point and use a k nearest neighbors classifier to color the background as seen in Figure 4. The resulting point cloud (hopefully) inherents distinct clusters. The number of clusters encounterd 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 appears 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 Reynolds number. A broad overview of the algorithm used in this paper can be found in Figure 4.

An example of the cluster analysis is depicted in Figure 4. In here, the time series of each cluster are represented by their centroid which is of their cluster at the given time step. We can see that each of the centroids represents a slightly different behaviors 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

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 and then apply shape filters over the pictures. In the first layer of the network, the filters will be detecting edges, slow gradients and color changes (Zeiler and Fergus, 2013). With proceeding layer, the filters begin to look like natural features such as: a birds eye, a bicycle wheel or a door-frame. 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.

We have discussed here neural networks with a high number of layers. This is referred to as deep learning. Deep learning is a field that has recently become a reality due to the abundance of graphical processor units (GPU) and more recently tensor processing units (TPU). Platforms such as PyTorch or TensorFlow provide high level front ends in Python. The front ends just generate code to handle low level interactions and optimizations. Furthermore, it is common practice to publish well performing neural network architectures that are already pre-trained (transfer learning). 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 Figure 7).

Dynamic stall vortices have a strong low pressure core which causes a lift overshoot and moment dump. When dynamic stall vortex data is averaged over 50+ cycles, it tends to show dynamic 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
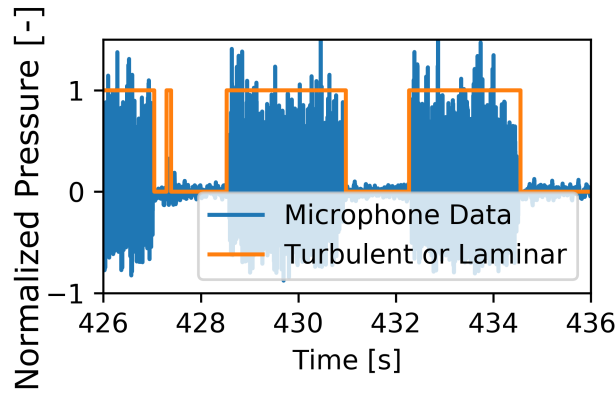
Figure 7. Identification of a boundary layer state using a recurrent neural network (data from Bak et al. (2010)) (see code example https://github.com/MatthewLennie/Aerodynamics)

the question: 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 using a Markov Chain Monte Carlo sampler to put sample puts onto the stall vortex. 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 wasn't trivial. However, this is a standard computer vision task very similar to a driver-less 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) [1]. 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 re-mould the convolutional layers to identify dynamic stall vortices's and give the convection speed and phase.

Pre-trained neural networks can be built and re-trained 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 and Others, 2019). The FASTAI architecture implements several current best practices as defaults such as; cyclical learning rates, drop-out, training data augmentation and data normalization. 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 was represented as a picture where the

---

[1]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.

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 Figure 8, the training data does not have the blue line identifying the vortex but is otherwise the same). Training data was 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 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 was 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, once the training edge reached an asymptote, the internal layers were unfrozen to mold the internal layers for a small number of epochs.

Initially 80% of the data was taken as 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 begun to increase. The convolutional layers were then unfrozen 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 [2]. The error statistics were still unsatisfactory and additional training did not improve the performance further. However, the current settings of the hyper-parameters settings and training procedure had seemed to extract the best model given the available data. The training procedure was repeated exactly the same a second time, with the same hyper-parameters and the same number of epochs, however this time the dataset wasn't split into test and training sets thus neglecting validation error. This may be perceived as a opening up the risk of over fitting, however the training procedure and hyper-parameters were already tested and the neural network didn't over-fit. Furthermore, usually additional data will usually 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 Figure 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 in the order of 30mins of computational time on a GPU. Readers are encouraged to view the source code at "https://github.com/MatthewLennie/VortexCNN". 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, we can on a statistical basis work backwards. 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

---

[2]Geometric augmentations would have been the next method to improve the model if the process hadn't worked well enough

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, the vortex will usually have an additional component normal to the airfoil.

320   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.

4   Results

4.1   Extracting vortex convection with a convolutional neural network

A number of test configurations with dynamic stall were chosen and pushed through the neural network to demonstrate some of the patterns. The first case is relatively complicated, as it features, an oscillating inflow velocity, pitching into the dynamic stall range and leading edge blowing. Four examples tests were compared with different phase differences between the angle of attack motion and the inflow velocity. Medina et al. (2018) made a very similar analysis and found that decelerating flow tended 330   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 11 shows that for cases where the inflow speed is in phase with the angle of attack, the shedding occurs later. However, when it does finally occur, the vortex will shed at a higher velocity (see Figure 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 335   of the vortex shedding and the convection speed, there is a weak correlation (0.3 Pearson metric) but not strong enough with the existing data to make conclusions about the relationship between the two.
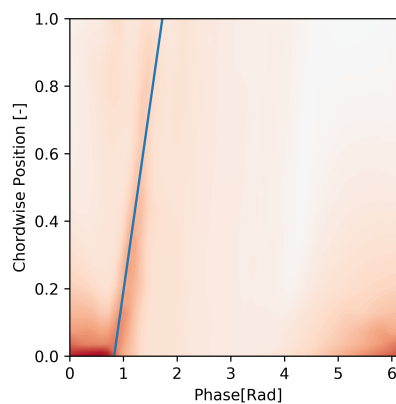
Figure 8. Example CNN Output, color intensity refers to suction pressure, blue line is regressed fit. Pressure is standardized therefore the colors represent Z-scores, no units or color bar are provided for this reason
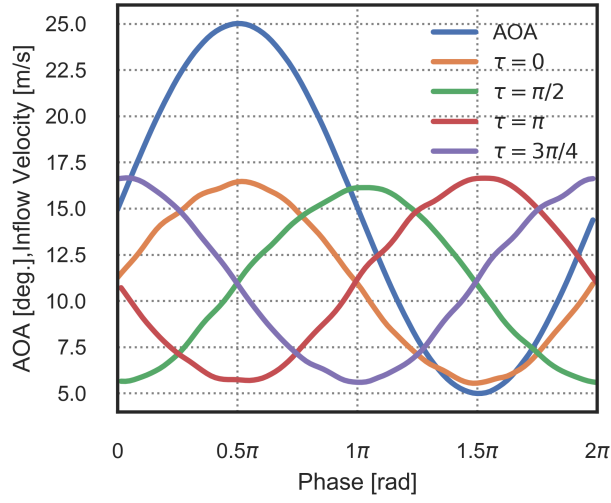
Figure 9. Inflow and angle of attack for Figure 10 and Figure 11
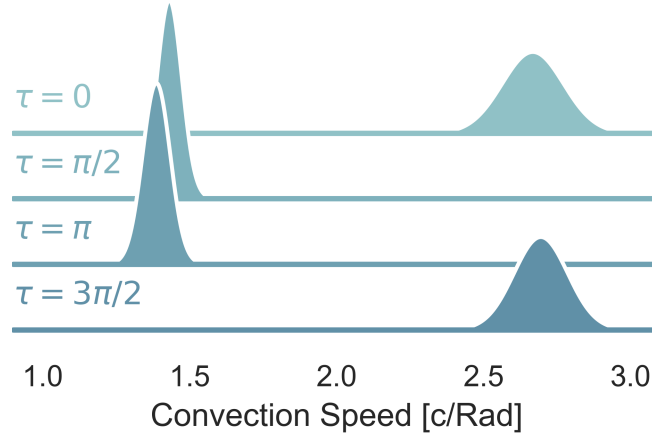
Figure 10. Probability distributions of the convection speed of dynamic stall with airfoil blowing different phases of harmonic inflow (τ). $\frac{U_{amp}}{U} = 0.5$, $k = 0.08$, $Re = 2.5 \cdot 10^5$ and $\alpha_0 = 15°$, $\alpha_{amp} = 10°$.

-
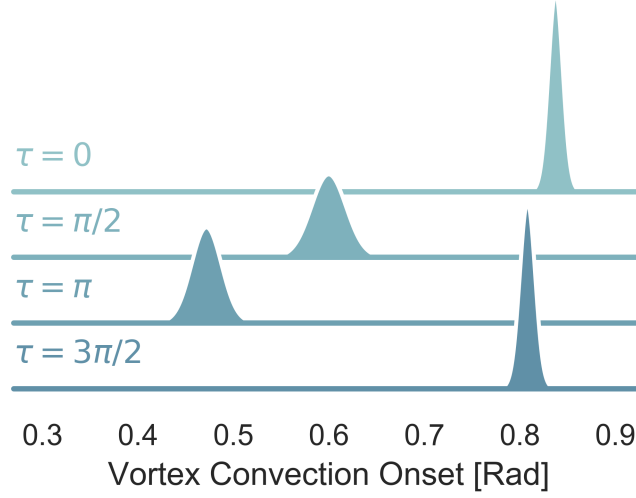
Figure 11. Probability distributions of the onset of dynamic stall with airfoil blowing and different phases of harmonic inflow (τ). $\frac{U_{amp}}{U} = 0.5$, $k = 0.08$, $Re = 2.5 \cdot 10^5$ and $\alpha_0 = 15°, \alpha_{amp} = 10°$.
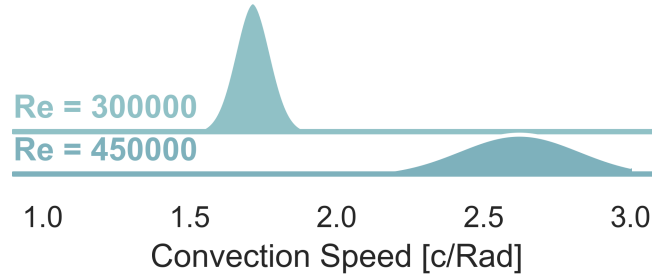
Figure 12. Probability distributions of the convection speed of dynamic stall with different Reynolds Number, $k = 0.09$, and $\alpha_0 = 18°, \alpha_1 = 7°$.

4.2 Dynamic Stall Clustering

In the following section, we have chosen a few examples purely for the purpose of demonstrating the methods. It would require a separate body of work to trawl over a larger selection of data in order to conduct a proper analysis. At high angles of attack ($\alpha_0 = 21.25°$ and $\alpha_{amp} = 8.25°$), we can observe the different kinds of stall behaviors that can occur. Figure 15 and Figure 16 show contrasting behaviors for the same angles of attack. In Figure 15, a quasi-periodic shedding appears. Without
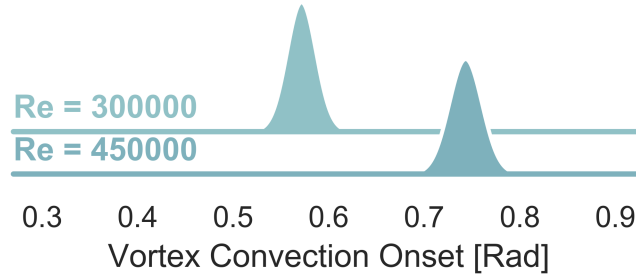
Figure 13. Probability distributions of the onset of dynamic stall with different Reynolds Number, $k = 0.09$, and $\alpha_0 = 18°, \alpha_1 = 7°$.
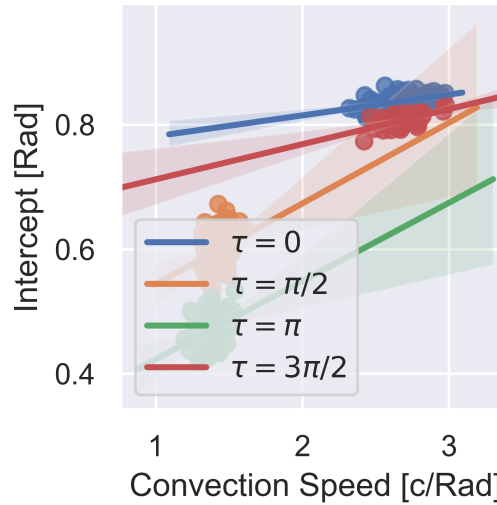
Figure 14. Relationship between the onset of shedding and convection speed for a range of blowing cases

flow visualization it is hard to determine the shedding type, but the pressure footprint show 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.
345 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.

 Now let us consider a second case with a different Reynolds number and reduced frequency but with the same angle of attack range (Figure 15). The airfoil moves into stall, releases one (cluster 2 - orange) or two coherent vortices's (cluster 1 -
350 blue cluster) and the resolves into weaker small scale shedding.

In Figure 17 and Figure 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 vorticity remain constant. The data does otherwise follow the general wisdom that the lift overshoot will increase with reduced frequency but not 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 non-existent secondary vorticity (read carefully, colors do not match). Interestingly the higher reduced frequency in Figure 18 seems to suppress the secondary vortex as Figure 17 shows strong secondary vorticity 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. Figure 20) make the differences clear and easy to interpret. In either of these cases, a phase averaged results would have been a poor representation of the dataset. This should help future modeling efforts as we should at least understand our data somewhat before trying to model it.
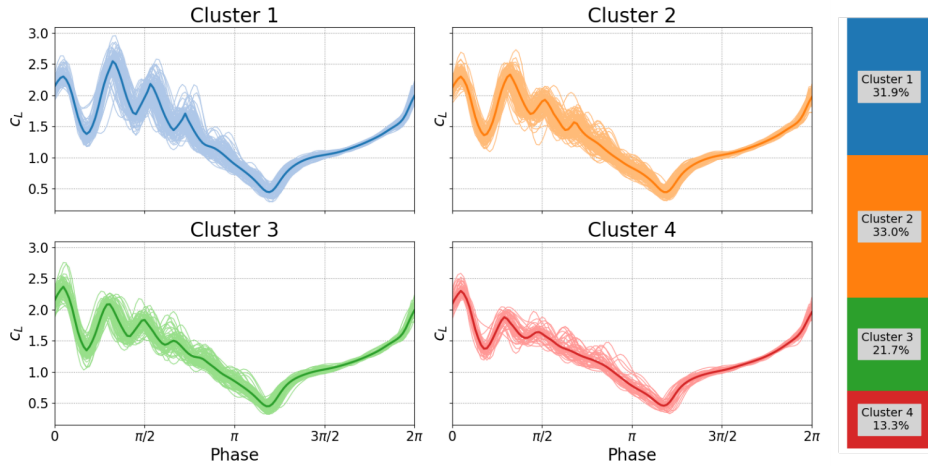
Figure 15. Deep stall investigations: cluster analysis for boundary conditions: $k = 0.0992$, $Re = 3.3 \cdot 10^5$ and $\alpha_0 = 21.25°, \alpha_{amp} = 8.25°$.

5 Convergence and outliers

The clustering and MDS can also be used together to identify outliers. 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
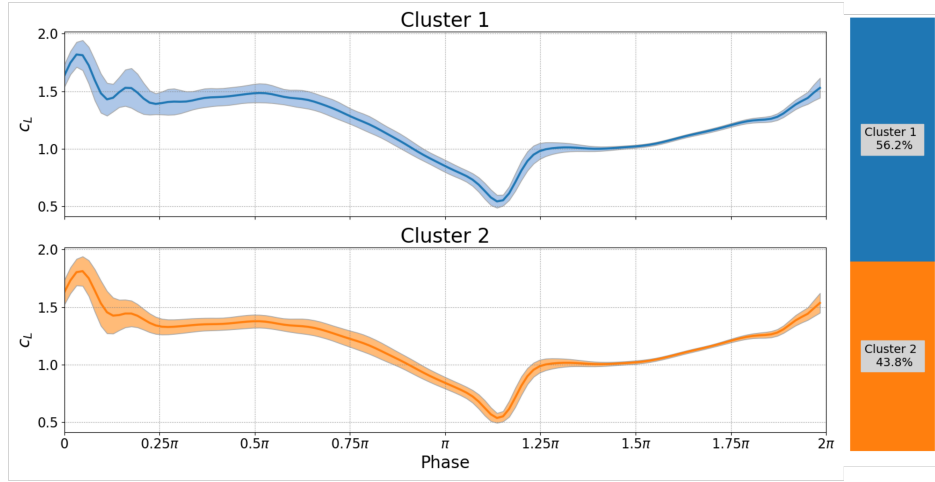
Figure 16. Deep stall investigations: cluster analysis for boundary conditions: $k = 0.0574$, $Re = 5.7 \cdot 10^5$ and $\alpha_0 = 21.25°, \alpha_{amp} = 8.25°$.
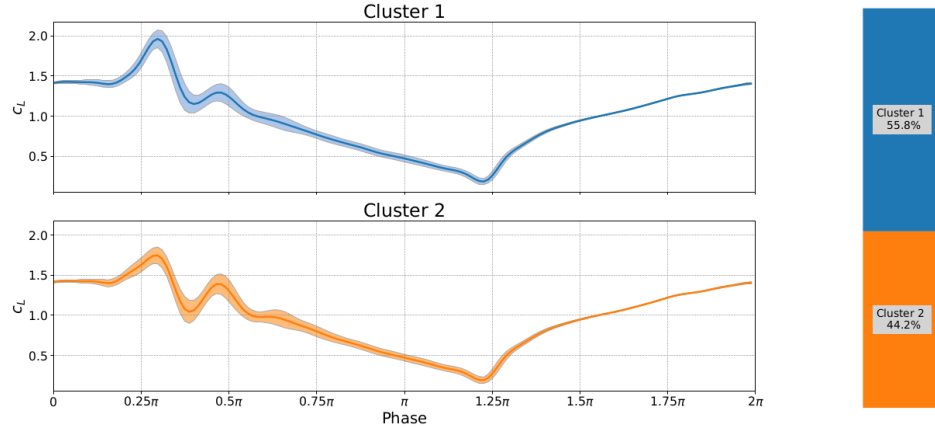
Figure 17. Deep stall investigations: cluster analysis for boundary conditions: $k = 0.0992$, $Re = 3 \cdot 10^5$ and $\alpha_0 = 18°, \alpha_{amp} = 7°$.

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 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 Figure 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.
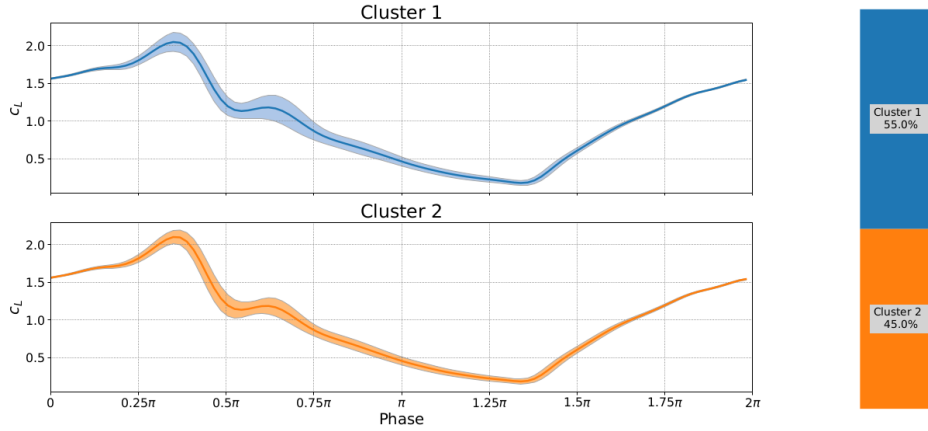
Figure 18. Deep stall investigations: cluster analysis for boundary conditions: $k = 0.1346$, $Re = 3 \cdot 10^5$ and $\alpha_0 = 18°, \alpha_{amp} = 7°$.
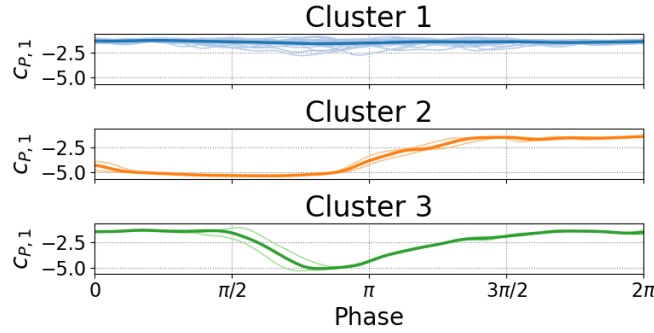
Figure 19. Clustered time series from surge experiment. Boundary conditions: $U_\infty(\phi) = 2.5\,\mathrm{m\,s}^{-1} + 0.7\,\mathrm{m\,s}^{-1}\sin(\phi)$, $\overline{Re} = 1.25 \cdot 10^6$, f=0.21 Hz,$\alpha = 10°$

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 re-sampling. The concept is quite simple; stick the data in a bucket, re-sample with replacement until you reach the size of the dataset, 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 the concept of confidence intervals. This provides us a quantitative statement such as "the existing data indicates 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).
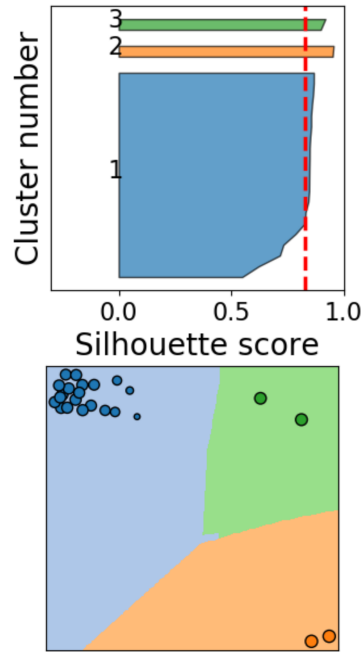
Figure 20. Above: Silhouette Samples per cluster. Below: MDS representation

In our case, we would like to see how the uncertainty of our population estimates decrease 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 Figure 21 and Figure 22). One will note that the variance and inter-quartile 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 less data. In practice this 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 inter-quartile range will serve better in cases of stall. All of the population estimates are presented here as percentile based estimates such as median and inter-quartile 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 (Dominin et al.).

A number of test cases were chosen with varying degrees of separation. In deep stall cases, as seen in Figure 22, the error of the standard deviation drops below 2% after 60 repetitions. The light stall case in Figure 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 then apply normalization to speed up convergence. Of course
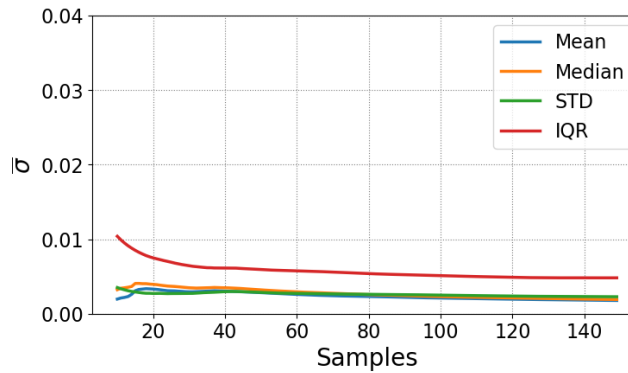
Figure 21. Convergence of the population estimates for a light stall case as the number of tests increase
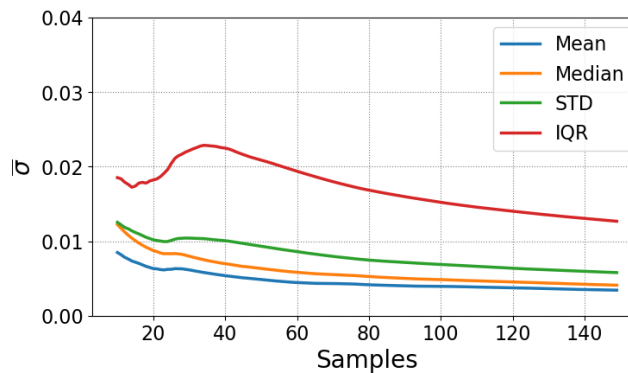
Figure 22. Convergence of the population estimates for a deep stall case as the number of tests increase

different levels of confidence would require more or less repetitions, however, for general purpose the following principles can be made:

1. For deep stall use <60-100 Cycles.

410   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 context of the limited example given here. In most of the examined cases, the variability and thus the rate of convergence was 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
415 is always best practice to conduct the bootstrapping for each new test configuration. The full results supporting this advice can be found in the master thesis of Steenbuck (2019).

6   Conclusions

This paper has attempted to bridge the gap between unsteady aerodynamics and the field of data science. Stall is a complex phenomena which varies in both time and space and the data has shown strong variations between cycles of the experiments. The combination of clustering, dynamic time warping and multi-dimensional scaling allow 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 with-in 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 vorticity. The neural network was able 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. 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., 2018). At higher angles of attack the wake structure and shedding frequencies would shift again (Lennie et al., 2018). Separated flows are incredibly complex and varied across different operating regimes.

The marriage of data science and aerodynamics presented in this study has 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. The empirical models[3], 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. Essentially it becomes too difficult for a human creator to write down a complex enough model that is well behaved over all operating regimes.

Machine learning provides another path to improving aerodynamic models, as it provides the tools and techniques to fit highly performing complex models while simultaneously handling the problem of over-fitting. Such an approach would perform much the same roll as the current models but would be machine learned. Using the concept of transfer learning it would be possible to train the model in stages. 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 this 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.

---

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

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 near endless supply of cheap data from the standard unsteady aerodynamics models. We would now have confidence that over nearly all operating conditions the model wouldn't diverge too far from reality. The model can then be re-moulded 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.

However, these are recommendations future investigation, the results of this study already provide a number of recommendations about stall and data science.

1. Means aren't sufficient at describing 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, machine learning is probably the better option.

5. Dynamic stall vortices's 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   Code Availability

A distillation of the codes used in this paper are available on https://github.com/MatthewLennie/ The data used for the convection plots is also in the repository. An example file is provided for the time series clustering with MDS plot.

8   Author Contribution

Matthew Lennie prepared the manuscript with the help of all co-authors. The computer vision model was constructed by Matthew Lennie. Johannes Steenbuck constructed the clustering code with assistance and supervision from Matthew Lennie. Bernd R. Noack provided code review and technical advice. Christian Oliver Paschereit provided assistance with the paper review.

9 Acknowledgments

The wind tunnel data was 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 was measured by Marvin Jentzsch and Hajo Schmidt at the Technische Universität Berlin. The time series microphone data used in the recurrent neural network example was provided by Helge Madsen under the DanAero project. The DanAero projects were funded partly by the Danish Energy Authorities, (EFP2007. Journal nr.: 33033-0074 and EUDP 2009-II. Journal nr 64009-0258) and partly by eigenfunding from the project partners Vestas, Siemens, LM, Dong Energy and DTU. These datasets required extensive measurement effort and on going 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.

10 Competing Interests

The authors declare that they have no conflict of interest.

References

Abbott, I. H. and von Doenhoff, A. E.: Theory of Wing Sections: Including a Summary of Airfoil Data, Dover Books on Aeronautical Engineering, Dover Publications, https://books.google.de/books?id=lWe8AQAAQBAJ, 2012.

Andersen, P. B., Gaunaa, M., Bak, C., and Hansen, M. H.: A Dynamic Stall Model for Airfoils with Deformable Trailing Edges, Journal of Physics: Conference Series, 75, 012 028, https://doi.org/10.1088/1742-6596/75/1/012028, http://stacks.iop.org/1742-6596/75/i=1/a=012028?key=crossref.1942e106c9f3122067370dc5265395b2, 2007.

Bak, C., Madsen, H. A., Fuglsang, P., and Rasmussen, F.: Double stall, vol. 1043, http://orbit.dtu.dk/fedora/objects/orbit:90308/datastreams/file{_}7731788/content, 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, European Wind Energy Conference and Exhibition (EWEC), pp. 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, Journal of Engineering for Gas Turbines and Power, 141, 1–10, https://doi.org/10.1115/1.4041150, 2019.

Borg, I. and Groenen, P. J. F.: Modern Multidimensional Scaling: Theory and Applications, Springer Series in Statistics, Springer New York, https://books.google.de/books?id=nRal-OSpknUC, 2007.

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 Journal, 52, 2384–2394, https://doi.org/10.2514/1.J052630, http://arc.aiaa.org/doi/abs/10.2514/1.J052630, 2014.

Brownlee, J.: A Gentle Introduction to Transfer Learning for Deep Learning, https://machinelearningmastery.com/transfer-learning-for-deep-learning/, 2017.

Brunton, S. L. and Noack, B. R.: Closed-Loop Turbulence Control: Progress and Challenges, Applied Mechanics Reviews, 67, 050 801, 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, Journal of Computational Dynamics, 2, 165–191, https://doi.org/10.3934/jcd.2015002, http://aimsciences.org//article/id/40180361-df75-4ef0-9b53-5b0613f8575f, 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, Experiments in Fluids, 55, https://doi.org/10.1007/s00348-014-1837-y, 2014.

Carr, L. W.: Progress in Analysis and Prediction of Dynamic Stall, Journal of Aircraft, 25, 6–17, https://doi.org/10.2514/3.45534, http://arc.aiaa.org/doi/abs/10.2514/3.45534{%}5Cnhttp://doi.aiaa.org/10.2514/3.45534, 1987.

Chernick, M. R.: Bootstrap Methods: A Guide for Practitioners and Researchers, Wiley Series in Probability and Statistics, Wiley, https://books.google.de/books?id=jx5K6IyjhhIC, 2008.

Cuturi, M. and Blondel, M.: Soft-DTW: a Differentiable Loss Function for Time-Series, http://arxiv.org/abs/1703.01541, 2017.

Dask Development Team: Dask: Library for dynamic task scheduling, https://dask.org, 2016.

Dominin, S., Lennie, M., and 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, pp. 1–13, Orlando, USA.

Ehlert, A., Nayeri, C. N., Morzynski, M., and Noack, B. R.: Locally linear embedding for transient cylinder wakes, Journal of Fluid Mechanics, [Under Con, http://arxiv.org/abs/1906.07822, 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, Journal of Physics: Conference Series, 753, 022 007, https://doi.org/10.1088/1742-6596/753/2/022007, http://stacks.iop.org/1742-6596/753/i=2/a=022007?key=crossref.74c47a0445c8a5a73ea2e5946de6137e, 2016.

Granlund, K., Monnier, B., Ol, M., and Williams, D.: Airfoil longitudinal gust response in separated vs. attached flows, Physics of Fluids, 26, 027 103, https://doi.org/10.1063/1.4864338, http://scitation.aip.org/content/aip/journal/pof2/26/2/10.1063/1.4864338, 2014.

Greenblatt, D.: Unsteady Low-Speed Wind Tunnels, AIAA Journal, 54, 1817–1830, https://doi.org/10.2514/1.J054590, http://arc.aiaa.org/doi/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, June, AIAA, https://doi.org/doi: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), pp. 1–9, IEEE, https://doi.org/10.1109/cvpr.2016.90, http://dx.doi.org/10.1109/CVPR.2016.90, 2016.

Holierhoek, J. G., Vaal, J. B. D., Zuijlen, A. H. V., and Bijl, H.: Comparing different dynamic stall models, pp. 139–158, https://doi.org/10.1002/we, 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, Journal of Engineering for Gas Turbines and 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, Journal of Fluid Mechanics, 798, 717–750, https://doi.org/10.1017/jfm.2016.345, 2016.

Howard, J. and Others: FASTAI, 2019.

Hudy, L. M. and Naguib, A.: Stochastic estimation of a separated-flow field using wall-pressure-array measurements, Physics of 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, San Diego, California, https://doi.org/10.2514/6.2019-2164, https://arc.aiaa.org/doi/abs/10.2514/6.2019-2164, 2019.

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

Kirk, T. M. and Yarusevych, S.: Vortex shedding within laminar separation bubbles forming over an airfoil, Experiments in Fluids, 0, 0, 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., pp. 1097–1105, Curran Associates, Inc., http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf, 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, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 233, 2794–2812, https://doi.org/10.1177/0954406218797972, https://doi.org/10.1177/0954406218797972, 2019.

Kutz, J. N., Fu, X., and Brunton, S. L.: Multi-Resolution Dynamic Mode Decomposition, Journal Applied Dynamical Systems, 15, pp. 713–735, https://arxiv.org/abs/1506.00564v1, 2015.

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, Cambridge University Press, Cambridge, 2nd edn., 2006.

565 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, pp. 1–12, https://doi.org/10.2514/6.2016-1006, http://arc.aiaa.org/doi/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, 35th Wind Energy Symposium, pp. 1–14, https://doi.org/10.2514/6.2017-2002, http://arc.aiaa.org/doi/10.2514/6.

570 2017-2002, 2017.

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, Journal of Engineering for Gas Turbines and Power, 140, 112 603–112 603, https://doi.org/10.1115/1.4039818, 2018.

Loiseau, J. C., Noack, B. R., and Brunton, S. L.: Sparse reduced-order modelling: Sensor-based dynamics to full-state estimation, Journal of

575 Fluid Mechanics, 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 2MW 80m diameter wind turbine rotor in comparison with transition data from wind tunnel measurements, AIAA, https://doi.org/10.2514/6.2019-0801, https://arc.aiaa.org/doi/abs/10.2514/6.2019-0801, 2019.

Maimon, O. and Rokach, L.: Data Mining and Knowledge Discovery Handbook, Springer US, https://books.google.de/books?id=

580 S-XvEQWABeUC, 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, Journal of Physics: Conference Series, 555, 012 068, https://doi.org/10.1088/1742-6596/555/1/012068, http://stacks.iop.org/1742-6596/

585 555/i=1/a=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, http://oai.dtic.mil/oai/oai?verb=getRecord{&}metadataPrefix=html{&}identifier=ADA098191, 1981.

590 McCroskey, W. J.: Unsteady Airfoils, Annual Review of Fluid Mechanics, 14, 285–311, https://doi.org/10.1146/annurev.fl.14.010182.001441, http://www.annualreviews.org/doi/abs/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 Journal, 56, 1–7, https://doi.org/10.2514/1.J056408, https://arc.aiaa.org/doi/10.2514/1.J056408, 2018.

595 Morel, M., Achard, C., Kulpa, R., and Dubuisson, S.: Time-series averaging using constrained dynamic time warping with tolerance, Pattern Recognition, 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, Experiments in Fluids, 52, 779–793, https://doi.org/10.1007/s00348-011-1118-y, 2012.

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

Mulleners, K. and Rütten, M.: Analysis of intermittent trailing edge vortex shedding using recurrence plots, 18th International Symposium
on the Applications of Laser and Imaging Techniques to Fluid Mechanics, p. 2016, 2016.

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

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

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

Muller-Vahl, H. F., Pechlivanoglou, G., Nayeri, C. N., Paschereit, C. O., and Greenblatt, D.: Matched pitch rate extensions to dynamic stall
on rotor blades, Renewable Energy, 105, 505–519, https://doi.org/10.1016/j.renene.2016.12.070, http://dx.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, Journal of Physics Fluid Dynamics, pp. 1–32, http://arxiv.org/abs/1809.07220, 2018.

Olah, C.: Visualizing MNIST: An Exploration of Dimensionality Reduction, https://colah.github.io/posts/2014-10-Visualizing-MNIST/,
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: The Science of Making Torque from Wind (TORQUE 2018), vol.
1037, Journal of Physics: Conference Series, Milan, https://doi.org/10.1088/1742-6596/1037/5/052018, 2018.

Ratanamahatana, C. and Keogh, E.: Everything you know about dynamic time warping is wrong, Third Workshop on Mining Temporal and
Sequential Data, pp. 22–25, https://doi.org/10.1097/01.CCM.0000279204.24648.44, http://spoken-number-recognition.googlecode.com/
svn/trunk/docs/Dynamictimewarping/DTW{_}myths.pdf, 2004.

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

Rumsey, C. L.: Successes and Challenges for Flow Control Simulations (Invited), Control, pp. 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, International Journal
of Heat and Fluid Flow, 32, 847–864, https://doi.org/10.1016/j.ijheatfluidflow.2011.06.011, 2011.

Schlichting, H. and Gersten, K.: Boundary-Layer Theory, Springer Berlin Heidelberg, Heidelberg, 9th editio edn., https://books.google.ie/
books?id=bOUyDQAAQBAJ, 2016.

Schmid, P.: Dynamic mode decomposition of numerical and experimental data, Journal of Fluid Mechanics, 656, 5–28,
https://doi.org/10.1017/S0022112010001217, https://hal-polytechnique.archives-ouvertes.fr/hal-01020654, 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, Journal of Fluid Mechanics, 792, pp. 798–828,
https://doi.org/10.1017/jfm.2016.103, 2015.

Skrzypi, W., 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, 2014.

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

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, pp. 1–19, https://doi.org/10.2514/6.2015-3071, http://arc.aiaa.org/doi/10.2514/6.2015-3071, 2015.

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

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 Journal, 55, 1–46, http://arxiv.org/abs/1702.01453, 2017.

Tavenard, R.: tslearn, https://tslearn.readthedocs.io/en/latest/index.html, 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, Journal of Physics: Conference Series, 524, 012 011, https://doi.org/10.1088/1742-6596/524/1/012011, http://stacks.iop.org/1742-6596/524/i=1/a=012011?key=crossref.5d435c0459d5992cddcee230bd222a76, 2014.

Ward, J. W.: The Behaviour and Effects of Laminar Separation Bubbles on Aerofoils in Incompressible Flow, The Aeronautical Journal -N. E. Rowe Medal Papers, 67, 783–790, 10.1017/S0001924000061583, 1963.

Wendler, J., Marten, D., Pechlivanoglou, G., Nayeri, C. N., and Paschereit, C. O.: 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.