Wind Farm Fatigue Loads - Methodology and Implementation

W. P. Engels (ECN)
A. Marina (ECN)

August 2015
ECN-X--15-098
Acknowledgement

I would like to acknowledge the work of Koert Lindenburg. Without his efforts this project would have been very difficult to complete and he lay most of the groundwork for the results in this report. I would also like to acknowledge the work of Johan Peeringa for describing the modifications of SwiftWake in more detail.

The work in this project was conducted as part of the FLOW project “Wind Farm Wake Modelling, Fatigue Loads and Control”, proposal number P201102_004.

Abstract

In wind farms, wind turbines are often exposed to a wind that is disturbed by the wake of other wind turbines, rather than an undisturbed wind. This causes different loading than the design loads described in IEC standards 61400-1 and 61400-3 that assume undisturbed wind loading.

One of the goals of this project is to study how wakes affect the loads on a wind turbine in a farm and to be able to take that into account in the design of a wind farm. To do this, the following methodology is used:

- An extensive database is established with different loads for different wind and wake conditions at the wind turbine;
- For each wind turbine in the farm, the wake conditions depending on wind speed and wind direction are calculated;
- The loads are interpolated from the database for the conditions for a particular wind turbine for all wind speeds and directions. Next the relative loading of different parts is established.

This report examines how such a database can be established. It also examines whether such a database has a predictive value for other wind turbines.

The results show that the database can be used to effectively interpolate the results and requires some 50 000 load cases to accurately interpolate the results. The use of the database shortens the time to obtain an accurate load prediction for a given wind farm design considerably.

In this database, wake meandering and the dynamic, controlled behaviour of upwind wind turbines is not taken into account. Not taking wake meandering into account leads to a small underestimation of the loads. The effect on loading due to dynamic effects in the wake due to variations due to the upwind wind turbine behaviour is not clear and cannot be evaluated at the moment.

The predictive value of the blade loads between the wind turbine types was good, but for the tower loads the prediction is less accurate. The cause of this needs to be further investigated, also for further different wind turbine designs.
Contents

1 Introduction 6

2 Farm Fatigue Methodology 7
  2.1 Methodology overview 7
  2.2 Wake parameters 8

3 Establishing load set and loads 10
  3.1 Adaptations to software 10
  3.2 Basic sensitivity study 11
  3.3 Selection of wind farm conditions 23
  3.4 Implementation of the database for Farmflow 23

4 Validating assumptions 24
  4.1 Effect of wake meandering 24
  4.2 Predicting results for other wind turbines 28

5 Conclusion 29

References 30

Appendices

A. Modification of simulation software 31
B. ART5 wind turbine design 35
C. Implementation of LoadModule as DLL 38
D. SwiftWake 46
1

Introduction

The wind conditions in a farm are often quite different than the conditions the wind turbine is designed for. That can mean that the loads in the wind farm may be higher than the design loads and lead to extra maintenance and wind turbine stand still due to failures.

To improve the performance of wind farms, it would be good to understand how the wind turbines are affected by the wakes, both in terms of power performance as well as the loads on various parts of the wind turbine.

To be able to use that knowledge in the design process, these loads must be obtained quickly and accurately. This report describes a method that can be used to evaluate the loads in the wind farm quickly and accurately.

To establish loads, the following methodology is used:
- An extensive database is established with different loads for different wind and wake conditions at the wind turbine
- For each wind turbine in the farm, the wake conditions depending on wind speed and wind direction are calculated
- The loads are interpolated from the database for the conditions for a particular wind turbine for all wind speeds and directions and the relative loading of different parts is established.

The following steps that are taken:
- establish the most relevant parameters for the database;
- establish the sensitivity of the loads to these parameters;
- verify the basic assumptions.

Furthermore it is examined whether one database can be used to predict loads of another wind turbine design.
2

Farm Fatigue Methodology

2.1 Methodology overview

The goal of the project is to quickly evaluate the loads and yields of a windfarm for a particular wind farm layout. This allows the wind farm designer to optimise the loads and yield of a wind farm.

The loads could be evaluated directly based on FarmFlow output. FarmFlow is an inhouse program that calculates the wind conditions, including wakes and turbulence parameters, assuming a steady wind speed and wind direction [Ecen2010]. For each wind turbine it calculates the steady state inflow conditions, including wake effects. These wake effects can then be taken into account in a load analysis using the aeroelastic simulation code Phatas [Lind2014] and evaluated over the whole farm. This approach is illustrated in figure 1.

For a 100 wind turbine wind farm, examining one wind farm layout for a wind rose at 5 degree intervals for 20 wind speeds would then result in 144 000 different wake situations that need to be evaluated. Although this can be achieved, the amount of computational time involved makes it unpractical for use in an optimisation of the wind farm layout.

Figure 1: Direct loads evaluation from the wake conditions
Another approach is to generate a database, containing results calculated for
generalised wake conditions, as illustrated in figure 2. The loads in the optimisation can
then be interpolated from the database. This interpolation will be in the order of 10 000
– 100 000 times faster than a calculation per wind turbine, per wind speed. It is
assumed that evaluating the loads for ca. 20 000 to 50 000 different wake conditions
will give a sufficiently accurate result from the interpolation.

2.2 Wake parameters

Figure 3 shows two wind
turbines, where the
second turbine is exactly
in line with the first wind
turbine and the wind
direction. In this case, the
wind speed at the second
wind turbine is
significantly reduced at
the second wind turbine.

It can also be seen that
behind the second wind
turbine the wake is a
combination of the wake of the second and the first wind turbine.

In total 6 parameters are identified that describe the wind environment at a wind
turbine in the wind farm. These are:

1) Wind speed + corresponding wave conditions
2) Ambient turbulence
3) Shear (in the wake)
4) The wind velocity deficit in the wake, also known as bulge_depth
5) Wake size: the further two wind turbines are apart, the more spread out the wake
will be. The parameter is called bulge_width.
6) Wake location: the center of the wake is often misaligned and the wake may cover
only part of the rotor. This parameter is called bulge_y_pos
Parameters that are lacking from this analysis are the distance between the wind turbines and the behaviour of the upwind turbine.

On the one hand, the distance between the wind turbines affects the wake meandering: the longer the distance between the wind turbines, the more the wake meanders. A well published wake meandering model is that of Risoe (e.g. [Lar2008], [Schmidt2011], [Mark2011], [Keck2014], [Mach2014] and [Keck2015]). The code used to generate the wind field here is SwiftWake (see Appendix D) and that does generate a wake meander path. However, that code assumes that the wake meandering is not related to the distance to the wind turbine in front of the wind turbine experiencing the wake.

The behaviour of the upwind wind turbine, on the other hand, results in a dynamic wake deficit in the wind speed. At below-rated wind speeds, this should reduce the amplitude of wind speed variations in the wake: if the wind speed increases in below rated conditions, the absolute induced wind speed caused by first wind turbine should increase as well, because it will try to maintain the optimal induction ratio. On the other hand, at above-rated wind speeds, the dynamic behaviour will amplify wind speed variations. In that case, if the wind speed increases, the induced wind speed of the first wind turbine will decrease in order to maintain the maximum power output, thus creating a larger wind speed variation in the wake at the second wind turbine.

For the moment, neither wake meandering nor the dynamic wake behaviour will be considered as part of farm fatigue. It will be checked whether wake meandering has a significant effect on fatigue loads. The dynamic wake behaviour cannot be included yet in the current models and is therefore not taken into account.
Establishing load set and loads

Normally a complete offshore loadset can easily consist of several thousand loadcases. However, in a wind farm, the possible circumstances at each wind turbine location can also vary heavily, with wind speed, turbulence and the location and geometry of the wake. It is not practical to calculate all loadcases for all possible circumstances for the wind. That means a selection of load cases must be made to limit the number of calculations that need to be done.

The approach is to first conduct a basic sensitivity study for each parameter, to establish which effects are the most pertinent and what the required resolution is. Then the values of the parameters are selected and the database established.

The wind turbine model that is used is the ART5 wind turbine model. This study model is a 126 m, 5 MW wind turbine, based on the NREL 5 MW reference model. A more detailed description of this wind turbine model is given in Appendix B.

3.1 Adaptations to software

For this study, we make use of the software packages Swift and Phatas. Swift generates wind fields, whereas Phatas is an aero-elastic simulation tool.

SwiftWake (Adams1996) was used to generate wind fields that reflect the shape of wake, based on the extra input parameters required to describe the wake, in particular the wind velocity deficit (called bulge_depth) and wake size (bulge_width). Wind speed, turbulence and wind shear were already used in the wake definitions.

Phatas was adapted to take wake location and wake meandering into account. These adaptations are further explained in Appendix A.
3.2 Basic sensitivity study

For each parameter for the wake, the sensitivity of the loads are examined to establish how sensitive the outcome is with respect to each particular variable and to establish what resolution will be needed in the final reference database.

3.2.1 Effect of wake velocity deficit

The first effect that is examined is the velocity deficit in the wake itself. This is simply the reduction of the wind speed in the wake. Figure 4 shows the for-aft tower bottom and flapwise blade root loads for varying relative wake velocity deficit ($bulge_{depth}$) and varying ambient wind speeds. The top figures show the fatigue loads for different wind speeds and velocity deficits. For each wind speed, the results of six different wind-wave realisations were averaged. Table 1 shows the exact input parameters that are used.

The second row of figures shows the average of the extreme, minimum and maximum loads encountered during the calculations at each wind speed and velocity deficit.

The last row of figures shows the numerical average of the fatigue loads over all wind speeds. This average is not intended as a design equivalent load, but only to show the trend of the loads depending on the wake velocity deficit.

At some high relative wake velocity deficits (0.7 and 0.8), the load calculations did not converge to a solution. This resulted in some calculations not being taken into account for the loads.

Based on these results, it can be concluded that:

1. The sensitivity of the fatigue loads on both the blade root bending moment and tower bottom bending moment increases as the wake velocity deficit increases.
2. At low windspeeds and low wake velocity deficits, the blade root loads are affected less strongly. Gravity may have a stronger effect here than the wake deficit.
3. Around rated conditions, the effect of wake velocity deficit is also reduced.
4. The difference between minimum and maximum of the loads increase with higher wake deficits and the maximum load shifts to higher ambient wind speeds.
5. For the reference database, a quadratic interpolation between the loads for varying wake velocity deficit would be ideal, but a linear approximation based on 3 or 4 points would capture the trend sufficiently.
Table 1: Table of input values for wake velocity deficit analysis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turbulence series</td>
<td>0.1</td>
</tr>
<tr>
<td>Shear series</td>
<td>0.12</td>
</tr>
<tr>
<td>Bulge depth</td>
<td>0.1, 0.3, 0.5, 0.7, 0.8</td>
</tr>
<tr>
<td>Bulge width</td>
<td>1.3</td>
</tr>
<tr>
<td>Bulge y pos</td>
<td>0.5</td>
</tr>
<tr>
<td>Yaw error series</td>
<td>8.0</td>
</tr>
</tbody>
</table>
Figure 4: The effect of the wake velocity deficit on for-aft tower bottom and flapwise blade root bending moments. Shown are the fatigue loads vs. wind speed for different bulge depths, averaged fatigue load vs. bulge_depth and maximum (solid line), average (dashed line) and minimum (dotted line) loads vs. wind speed for different bulge depths. The bulge depth is relative, e.g. a bulge depth 0.8 means that wake deficit is 80% of the ambient wind speed and the remaining wind speed in the center of the bulge is 20% of ambient. Fatigue loads are calculated with a fatigue exponent of 10 for the blade root and 4 for the tower bottom moment.
3.2.2 Effect of ambient turbulence

Figure 5 shows the loads versus ambient turbulence. The figures show similar trends as the wake velocity deficit. The figures show that the fatigue loads increase generally linearly with turbulence intensity and that the increase is significant. Unlike for the wake velocity deficit, there do not seem to be any regions of operation where the turbulence increase has little effect. Table 2 shows the parameters that are used for the plots in figure 5.

A linear approximation on the basis of two or three turbulence datapoints should suffice for the reference database.

Table 2: Table of input values for ambient turbulence analysis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turbulence series</td>
<td>0.05, 0.1, 0.15, 0.2, 0.25, 0.3</td>
</tr>
<tr>
<td>Shear series</td>
<td>0.12</td>
</tr>
<tr>
<td>Bulge depth</td>
<td>0.4</td>
</tr>
<tr>
<td>Bulge width</td>
<td>1.3</td>
</tr>
<tr>
<td>Bulge y_pos</td>
<td>0.5</td>
</tr>
<tr>
<td>Yaw error series</td>
<td>8.0</td>
</tr>
</tbody>
</table>
Figure 5: The effect of ambient turbulence on for-aft tower bottom and flapwise blade root bending moments. Shown are the fatigue loads vs. wind speed for different bulge depths, averaged fatigue load vs. turbulence and maximum (solid line), average (dashed line) and minimum (dotted line) loads vs. wind speed for different ambient turbulences. Fatigue loads are calculated with a fatigue exponent of 10 for the blade root and 4 for the tower bottom moment.

Fatigue loads per wind speed

Extreme loads per wind speed

Wind speed average fatigue loads
3.2.3 Effect of wake size

Figure 6 shows the loads as the wake size varies. The figures show that the loads are not affected as much by the wake size as they were for turbulence intensity or wake velocity deficit. The figures also show that the blade root fatigue loads decrease slightly with increased wake size, whereas the tower loads increase slightly with increased wake size. For the blade root moments, the effects are more noticeable around rated wind speeds, whereas for the tower, the loads are affected more at low and high wind speeds. Table 3 shows the parameters that are used for the plots in figure 6. The parameter for the wake size is called ‘Bulge width’.

Both sensitivities decrease as the wake size increases.

A linear approximation on the basis of two or three wake-size datapoints should suffice for the reference database.

<table>
<thead>
<tr>
<th>Table 3: Table of input values for wake size analysis.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turbulence_series</td>
</tr>
<tr>
<td>Shear series</td>
</tr>
<tr>
<td>Bulge depth</td>
</tr>
<tr>
<td>Bulge width</td>
</tr>
<tr>
<td>Bulge y pos</td>
</tr>
<tr>
<td>Yaw_error_series</td>
</tr>
</tbody>
</table>
**Figure 6**: The effect of wake size on for-aft tower bottom and flapwise blade root bending moments. Shown are the fatigue loads vs. wind speed for different bulge depths, averaged fatigue load vs. wake size and maximum (solid line), average (dashed line) and minimum (dotted line) loads vs. wind speed for different wake sizes. Fatigue loads are calculated with a fatigue exponent of 10 for the blade root and 4 for the tower bottom moment.
3.2.4 Effect of wake location

The effect of the wake location on the loads is quite different from the turbulence, wake size and wake velocity deficit. Figure 7 illustrates different wake locations. Figure 8 shows the effect of the wake location on loads for varying wake locations. Table 4 shows the parameters that are used for the plots in figure 8.

It can be seen that the blade loads in particular vary strongly non-linearly with the wake location. The wake location has therefore been examined in greater detail. Figure 9 shows the results of a detailed analysis for wake locations varying from -1.0 to -0.1.

The blade loads peak when the wake is near the blade tip, whereas the tower loads are at their highest when the wake is exactly aligned with the wind turbine. For the final database at least 7 points are needed to cover large misalignment, the start of overlap, the highest load and the midpoint.

Figure 7: The wake location is relative to the rotor diameter, e.g. if the wake location is 0.5, the centre of the wake is placed exactly on the tip of the rotor.

<table>
<thead>
<tr>
<th>Table 4: Table of input values for wake location analysis.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>Turbulence_series</td>
</tr>
<tr>
<td>Shear_series</td>
</tr>
<tr>
<td>Bulge_depth</td>
</tr>
<tr>
<td>Bulge_width</td>
</tr>
<tr>
<td>Bulge_y_pos</td>
</tr>
<tr>
<td>Yaw_error_series</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5: Table of input values for detailed wake location analysis.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>Turbulence_series</td>
</tr>
<tr>
<td>Shear_series</td>
</tr>
<tr>
<td>Bulge_depth</td>
</tr>
<tr>
<td>Bulge_width</td>
</tr>
<tr>
<td>Bulge_y_pos</td>
</tr>
<tr>
<td>Yaw_error_series</td>
</tr>
</tbody>
</table>
**Figure 8**: The effect of wake location on for-aft tower bottom and flapwise blade root bending moments. Shown are the fatigue loads vs. wind speed for different bulge depths, averaged fatigue load vs. bulge_depth and maximum (solid line), average (dashed line) and minimum (dotted line) loads vs. wind speed for different bulge depths. Fatigue loads are calculated with a fatigue exponent of 10 for the blade root and 4 for the tower bottom moment.
Figure 9: The effect of wake location in more detail on for-aft tower bottom and flapwise blade root bending moments. Shown are the fatigue loads vs. wind speed for different wake locations, averaged fatigue load vs. wake location and maximum (solid line), average (dashed line) and minimum (dotted line) loads vs. wind speed for different wake locations. Fatigue loads are calculated with a fatigue exponent of 10 for the blade root and 4 for the tower bottom moment.

Fatigue loads per wind speed

Extreme loads per wind speed

Wind speed average fatigue loads
3.2.5 Effect of wind shear

The effect of wind shear is shown in the picture 10 below. Wind shear does also have a slight effect on the effective turbulence, but it is minimal. Only if there is extreme wind shear, is there a significant effect on the loading. The tower is nearly completely unaffected by wind shear. Table 6 shows the parameters that are used for the plots in figure 10.

The difference between the loads is very small. To allow interpolation, at least two values must be chosen that cover the range of wind shear that is encountered in the farm.

<table>
<thead>
<tr>
<th>Table 6: Table of input values for detailed wake location analysis.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turbulence series</td>
</tr>
<tr>
<td>Shear series</td>
</tr>
<tr>
<td>Bulge depth</td>
</tr>
<tr>
<td>Bulge width</td>
</tr>
<tr>
<td>Bulge y pos</td>
</tr>
<tr>
<td>Yaw error series</td>
</tr>
</tbody>
</table>
Figure 10: The effect of shear in the wake on for-aft tower bottom and flapwise blade root bending moments. Shown are the fatigue loads vs. wind speed for different shear values, averaged fatigue load vs. shear and maximum (solid line), average (dashed line) and minimum (dotted line) loads vs. wind speed for different shear values. Fatigue loads are calculated with a fatigue exponent of 10 for the blade root and 4 for the tower bottom moment.
3.3 Selection of wind farm conditions

Considering the results of the sensitivity studies, a selection is made for the parameters for which the database is calculated. This selection is chosen such that an accurate interpolation is possible for different wakes in the wind farm. The main considerations are that any non-linearities are sufficiently covered such that linear interpolation gives an accurate result and that the scope of the parameters covers the wakes calculated with FarmFlow for typical wind farms.

The final selection is shown in table 7. The wind speeds that are analysed are [4 6 8 10 12 14 16 18 20 22 25] m/s. For 6 different seeds, this adds up to a total number of 49 896 load cases.

<table>
<thead>
<tr>
<th>Turbulence series</th>
<th>0.05, 0.11, 0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shear series</td>
<td>0.05, 0.15, 0.25</td>
</tr>
<tr>
<td>Bulge depth</td>
<td>0.0, 0.3, 0.5, 0.8</td>
</tr>
<tr>
<td>Bulge width</td>
<td>1.2, 2.0, 4.0</td>
</tr>
<tr>
<td>Bulge_y_pos</td>
<td>-1.8, -1.2, -0.6, 0.0, 0.6, 1.2, 1.8</td>
</tr>
<tr>
<td>Yaw_error_series</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 7: Table of input values for detailed wake location analysis.

3.4 Implementation of the database for Farmflow

The calculation of the required load cases on ECN’s 120 core Intel-Xeon cluster, while using Focus6 required ca. 2 days of runtime. Once the database has been established, the loads can be evaluated quite quickly.

Appendix B describes the implementation of the database in greater detail. As the database needs to be loaded only once, the evaluation of the loads for as many as 100 000 different wake situations should take a few minutes at most on a common pc. This compares very favourably to 4 days of cluster run time, if it was necessary to run a Phatas calculation per wake situation.

This makes using the database in an optimisation of the wind farm layout, quite feasible.
4 Validating assumptions

4.1 Effect of wake meandering

Wake meandering results in the wake moving across the rotor disc, according to some horizontal and vertical path. This means that the wind turbine will experience different wake locations and additional variations in rotor-effective wind speeds.

Meandering is a result of lateral and/or vertical components in the wind and/or an instantaneous misalignment of the overall wind direction. The effect is that the wake moves away from the median wake trajectory in a process that more or less resembles a random walk [Lar2008].

It has been shown above, that both the turbulence and the wake location have a significant effect on the fatigue loading of both the tower and the blades. It seems likely therefore that wake meandering will also have a significant effect on the loads.

To study the effect of wake meandering as well as the implementation (see section 2.1 and appendix A) that was used, several data sets are examined:

- A case with no wind velocity deficit and no meandering at different wake locations;
- A case with a wind velocity deficit but no meandering at different wake locations;
- A case with no wind velocity deficit, but where ‘meandering’ is modelled (see Appendix A.);
- The complete case with wind velocity deficit and meandering.

To establish a fair comparison, the meander path is prescribed rather than derived from the average lateral wind speed components in the wind field. Only one wind speed was
examined (10 m/s). With 6 different wind seeds. Also, a fairly high induced wind velocity is chosen to see the effect most clearly.

Table 8: Table of input values for detailed meandering analysis.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turbulence series</td>
<td>0.2</td>
</tr>
<tr>
<td>Shear series</td>
<td>0.12</td>
</tr>
<tr>
<td>Bulge depth</td>
<td>0.4</td>
</tr>
<tr>
<td>Bulge width</td>
<td>1.3</td>
</tr>
<tr>
<td>Bulge_y_pos</td>
<td>0.0, 0.5, 1.0, 2.0</td>
</tr>
<tr>
<td>Yaw error series</td>
<td>-8.0, 0.0, 8.0</td>
</tr>
</tbody>
</table>

Figure 11 (a) shows results for one set of simulations plotted at different wake locations (bulge_y_pos). The meander path was defined here as a sinusoid path, with an amplitude of 32 m (1/4 rotor diameter) and a frequency of 30 s. Assuming the wind turbines are 8 rotor diameters apart and that there is no change in the lateral component of the wind speed over time, a lateral component of 5% of the wind speed would result in a ¼ diameter shift. If meandering is viewed in terms of instantaneous misalignment rather than a lateral component in the wind, this shift corresponds to an instantaneous misalignment of 1.8 degrees.

The period of this frequency should be seen as the time it takes for the wake displacement to lose its coherence. At 10 m/s, 30 seconds corresponds to 300 m in space. This order of magnitude seems reasonable when compared to measurements in the field [Kum2015].

Figure 11 (b) to (d) show the loads as the period and the amplitude of the meandering is varied. Longer periods are examined as the loads should to some extent converge towards a case where no meandering is taken into account.

When examining the effect of meandering, in figure 11 a few things are clear:

- There is some difference between when meandering is simulated without a velocity deficit in the wind. The cause of this is the somewhat reduced coherence in the sampled wind, because the wind is shifted relative to the wind turbine. This reduced coherence acts effectively as an increased turbulence. It should be noted that this is purely a simulation artefact due to the way meandering was modelled.

- This can also be observed in the data with a velocity deficit at the 2D location of the wake: for small amplitudes of the prescribed sinusoidal meandering (half D or less, i.e. 64 m), the wake should not have any overlap with the rotor. The increase in loads at this location at these amplitudes is again a simulation artefact. At high amplitudes, there will be some overlap of the wake with the rotor and the loads will increase more.

- The change in loading with and without a wake and the change in loading for the various meander periods and frequencies is considerably larger than the simulation artefact.

- The fatigue loads on the blades are affected less than the tower loads (ca. 3% vs. 5% increase in loads in figure 11(a)) when meandering is included.
The effect on the loads is less than that of the velocity deficit in the wake, turbulence or wake location as long as frequency and amplitude of the wake meandering is limited.

Based on these results, it can be concluded that not taking the wake meandering effect into account in the evaluation of the loads will give some inaccuracies and somewhat underestimate the loading.
Figure 11: The effect of meandering in the wake on fatigue for-aft tower bottom and flapwise blade root bending loads. Fatigue loads are calculated with a fatigue exponent of 10 for the blade root and 4 for the tower bottom moment and shown as function of wake location. (a) shows the loads with and without meandering, whereas (b) – (d) show the loads for various amplitudes and periods.

Fatigue loads at 10 m/s

(a) Loads with and without meandering (32 m, 30 s period)

(b) Loads at various meander periods, amplitude 32 m

(c) Loads at various meander periods, amplitude 64 m

(d) Loads at various meander periods, amplitude 128 m
4.2 Predicting results for other wind turbines

The idea behind using the database is that it can also be used in early design phases of a wind farm when the wind turbine selection is not final. To use it as such, the loads must be shown to have good predictive values for a different wind turbine design.

For this study the Nordex N80 is used. The data presented below, is only in relative loads, i.e. the load in the wake relative to the fatigue load at the same mean, undisturbed wind speeds for the normal design load cases. Figure 12 shows the relative blade root and blade tower loads of the ART wind turbine relative to the Nordex N80 wind turbine.

The predictive value of the loads should be considered 'good' if the relative loads on the N80 have increased as much for the same load cases as for the ART5 wind turbine, i.e. if the loads for the ART5 wind turbine increase by a factor 2 for a particular wake condition, the loads for the N80 should have increased by similar amount. That means if the relative loads of the N80 and ART5 are plotted against each other for each wake conditions, they should, ideally form a line through 0.0 with a slope 1.0.

The figure on the left shows that the database does indeed have good predictive value for the blade loads (slope of 0.999 and -0.03 vertical offset). However, the figure on the right shows that the tower loads are not predicted as well. The slope is 1.2 rather than 1, but more importantly the offset is quite off. The relative loads for the N80 go as low as 0.25. That means the loads calculated for the reference cases are up to 4 times higher than the cases including the wake.

The exact cause of this difference has yet to be determined.

Figure 12: Comparison of relative loads for similar wake conditions, for the blade (left figure) and the tower (right figure)
5
Conclusion

Based on the data, presented in chapter 3 and 4, we can conclude that the database can be used effectively to evaluate the loads in a wind farm. Ambient turbulence and the velocity deficit in the wake influence the loads very strongly. The shear on the other hand, hardly influences the loads at all.

Two main uncertainties remain in the calculation of the loads; these are dynamic wake meandering and dynamic effects in the wake due to variations in the wind and due to the upwind wind turbine behaviour. Both these effects are currently further examined as part of the TKI project “Simulation of Wind Farm Aerodynamics” (project nr. P201406-003-WMC).

If wake meandering were taken into account, the predicted loads are likely to increase. The current method of modelling would likely lead to a slight over-prediction of the blade loads. This over-prediction is small relative to the higher loads due to meandering and very small when compared to the effects of turbulence and the velocity deficit. The effect of the dynamic wake due to the behaviour of an upwind turbine and the development of that dynamic wake over time cannot yet be taken into account accurately. It should be examined if the effect of meandering and upwind wind turbine behaviour can be included as, for instance, additional turbulence. This would limit the overall size of the database.

The predictive value of relative loads between wind turbine types also needs to be further investigated. The agreement between the blade loads was excellent, but was poor for the tower loads. The cause of this difference is not yet clear. Furthermore, it is difficult to conclude the predictive value for other turbines on the basis of one comparison. More wind turbines should therefore be analysed.

Despite the lack of a dynamic wake model that includes upwind wind turbine behaviour and a more physical accurate wake meander model, we conclude that the use of the database is very effective at predicting the relative loads for different configurations of the same wind farm.
References


Appendix A. Modification of simulation software

In order to analyse the effect of wakes on wind turbines, ECN had to adapt its simulation software. Logically, the wake affects the wind input at the wind turbine, so the biggest changes are to be expected in the wind generation software.

The wind generation program SwiftWake (Adams1996) is used to create a wind file based on output from FarmFlow. Farmflow is able to calculate a steady state wake description, at a particular wind turbine in the wind farm, depending on the wind direction and average wind speed. This work flow is illustrated in figure 13.

However, the intention is to also generate a database, containing the loads in different wind conditions. To be able to conveniently analyse a large set of different wake situations a specific load case preparation program was developed (LcprepWake). This program generates a description of the wake in terms of the parameters. A second program, (Make Wake) then translates these parameters to the same file format as the FarmFlow wake output. This workflow is illustrated in figure 14.

---

**Figure 13: Farmflow calculation to loading workflow**

Farmflow

- Static wake description per turbine, wind direction and wind speed

Swift wake

- Turbulent wind field
- Wake meander path

Phatas

- Loading

**Figure 14: Lcprep to loading workflow**

Lcprep

- Loadcases with different wake situations

Make wake

- Static wake description

Swift wake

- Turbulent wind field
- Wake meander path

Phatas

- Loading
A.1. Adaptation of Swift

Swift was originally designed to create wind fields for the load cases in the IEC61400 standard with turbulent wind fields. As this standard requires the analysis of a single wind turbine, it was not equipped with additional inputs for wake meandering analysis.

In normal case, Swift therefor generates a wind field that does not take this into account. The wind field is a circular wind field, with data at different azimuth angles. The diameter of this wind field generally corresponds to the diameter of the wind turbine. This is illustrated in figure 13(a).

Ideally, the wake only affects the wind input, as illustrated in figure 13(b). Here the wake has an offset relative to the centre of the wind turbine and it meanders within the wind field and the wind field interacts with the rotor as normal.

Unfortunately, the ideal solution was not feasible for this investigation, as the required modifications to the code of Swift were too extensive for this project. Currently a different project is working on generating new input for Phatas, that not only enables the ideal solution, but also allows time variations in the wake.

For the purpose of optimisation, one ought to have a rough idea of the effect of wake meandering and therefor a simpler method was employed.

The approach that was actually used is illustrated in figure 13(c). The figure shows the wake in the Swift wind field at a specific offset. The meandering in turn is taken into account by shifting samples in the wind field relative to the rotor. This is equivalent to shifting the wind field relative to the rotor. This does, however, reduce the coherence of the wind as experienced by the rotor. This could be considered equivalent to increasing the turbulence in the wind.
Figure 15: Wind field and wake location versus rotor location

(a) Default situation: no wake, wind field and wind turbine are aligned

(b) Ideal situation: wind field and wind turbine are aligned, wake affects wind field only

(c) Approximation: Wake is at fixed location within the wind field. The wind is shifted relative to the wind turbine.
Appendix B. ART5 wind turbine design

This appendix contains a detailed description of the 5MW Aerodynamic Reference Wind Turbine (ART5) design, which is used as generic wind turbine model for this study. The ART5 is based on the Upwind 5MW, with modifications to the rotor design and controller.

B.1. UpWind 5MW

The 5MW turbine as documented by NREL (Jonk2009) is used as basis of the ART5. This is a three bladed collective pitch to vane controlled turbine for installation in an offshore environment. For the design of this turbine, there was a heavy emphasis on the usage of data from the REpower 5MW turbine and the DOWEC project (Kooi203). A summary of the main turbine parameters is given in the table below.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rating</td>
<td>5 MW</td>
</tr>
<tr>
<td>Wind regime</td>
<td>IEC 61400-3 (Offshore) Class 1B/Class 6 winds</td>
</tr>
<tr>
<td>Cut in / out wind speed</td>
<td>4 m/s / 25 m/s</td>
</tr>
<tr>
<td>Rotor orientation</td>
<td>Upwind</td>
</tr>
<tr>
<td>Control</td>
<td>Variable speed, collective pitch</td>
</tr>
<tr>
<td>Rotor diameter / Hub diameter</td>
<td>126 m / 3 m</td>
</tr>
<tr>
<td>Hub height</td>
<td>90 m</td>
</tr>
<tr>
<td>Maximum rotor / Generator speed</td>
<td>12.1 rpm / 1173.7 rpm</td>
</tr>
<tr>
<td>Generator efficiency</td>
<td>94.3%</td>
</tr>
<tr>
<td>Maximum tip speed</td>
<td>80 m/s</td>
</tr>
<tr>
<td>Overhang / Shaft tilt / Precone</td>
<td>5 m / 5° / -2.5°</td>
</tr>
<tr>
<td>Rotor/Nacelle/Tower mass</td>
<td>110,000 kg / 240,000 kg / 347,460 kg</td>
</tr>
<tr>
<td>Water depth</td>
<td>20 m</td>
</tr>
</tbody>
</table>

B.2. Blade redesign

The 61.5 m long blade of the UpWind reference model was derived from the 62.6 m long blade designed for the DOWEC project. Thereto the last 1.1 m was cut off without a proper redesign.
A closer look at the planform of the DOWEC blade showed a rather large chord compared to common blade designs, as well as an unconventional induction along the blade. Hence an aerodynamic redesign of the blade was performed.

The redesign is based on the DOWEC blade without shortening it and thus lowering the power density to approximately 380 W/m². This is in line with the current trend of larger rotor diameters for the same rated power. A slightly smaller chord was employed in the mid- and outboard sections. One of the outboard profiles was changed to improve the matching between the profiles, whilst maintaining aerodynamic performance. The twist was lowered, resulting in higher angles of attack but still maintaining a safe margin from stall.

The maximum power coefficient remains the same at $C_p=0.50$ for $\lambda=7.7$. Here it is noted that the rotor characteristics are calculated without taking into account the effects of rotor tilt, wind shear, pre-bend and turbine flexibility. Figure 16 illustrates the differences between the DOWEC and ART-5 blade designs.

The relative thickness was kept constant along the radius. This implies a slightly lower absolute thickness since the chord is reduced. The largest reduction approximately amounts to 19% at $r=51.5$ m. It is however chosen to leave the DOWEC structural properties (stiffness and mass distribution) unchanged. The redesign of the blade necessitates a new controller design compared to the NREL reference turbine controller, which is discussed in the next section.

### B.3. Controller design

Using the Advanced Control Tool (ACT) a pitch and torque control algorithm can be designed, ready for industrial application on multi megawatt variable speed wind turbines. Below rated wind speed the controller aims for optimal power production, and above rated for speed and power regulation. Industry standard stability margins are guaranteed and actuator protection (limitation) is included.

The ART5 is designed to deliver a nominal output of 5MW at 12.1 rpm. Losses are assumed to occur in the generator ($\eta=94.3\%$) as conversion from mechanical to
electrical power. As the controller supplies the mechanical power set point, the rated power is set to 5.3MW at 12.1rpm. Below rated, the optimal lambda ($\lambda=8.2$) curve is followed using variable speed torque control. Pitch is held constant at the optimal angle (1deg). The turbine starts at 3m/s, 6rpm rotor speed, and cuts out at 25m/s. Peak shaving is applied to reduce the thrust peak around rated. The pitch angle already starts to increase before rated (11rpm) to 4deg at rated. Other load reduction algorithms (IPC, tower damping) are not applied for this reference design. Apart from changes to the controller, a more realistic pitch actuator is selected, with the following constraints:
- maximum pitch acceleration of 12deg/s²
- maximum pitch speed of 6deg/s
- pitch angle range of 0 to 90 deg

The resulting controller regulates the rotor speed during normal operation within 10% of the nominal value. Above rated, the wind turbine delivers almost constant nominal power. The actuator limitation forces the actuator to operate within the given constraints.
Appendix C. Implementation of LoadModule as DLL

This appendix describes the implementation of the loads database as a dynamic link library (DLL) to be called by FarmFlow to give the loads on wind turbines in a wind farm. The following topics are covered:

- Overview of the programming
- Format of the binary database files
- Overview of the DLL
- Interpolation method
- Error codes

C.1. Overview of the programming

The DLL created for use in the program FarmFlow uses two databases to calculate the loads on wind turbines within a wind farm. The values from the database are calculated in the program Phatas, and the relative post processing and analysis is conducted to insert them into the databases. The two databases are created in a standardized manner as indicated in the section which follows. The main function of the DLL is to perform a multidimensional interpolation of the relevant database, perform error checking on the input data and return the appropriate database data to FarmFlow. The figure which follows outlines how the appropriate components interact with each other.

The DLL was written in Matlab due to the presence of predefined functions for interpolation of the data. The toolbox Matlab Coder is used to create a C/C++ Dynamic Library and all of the relevant files associated with a Dynamic Link Library that can be called through Windows programs.

Testing of the functionality of the DLL was conducted in two parts. For variation of the correct working of the DLL, a test bench program was written in Visual Studio 2013. For verification of the interpolation schemes, analysis was conducted for both 1 dimensional and 2 dimensional interpolation of all the relevant variables. It was assumed the results could be extended to n dimensions.
C.2. Format of the binary database files

For the creation of the database, various simulations were conducted in Phatas. Two databases were generated. The generated databases and the parameter variables are indicated below. Two databases are included as for high wind speeds, variation of the yaw angle is unlikely.

For each database, data from various signals and locations are available. For each signal and location combination, the relevant data is available for material exponents 2 through 14. For insertion of these values to relevant binary database files, the signals for each material exponent are averaged over all the seed values. The signal and locations data that are available are given in the following table. Up to 25 signal, location and material exponent are can be input to the DLL. The corresponding 25 outputs are returned to the Farm Flow program.
### Table 10: Reference Database:

<table>
<thead>
<tr>
<th>Wind Speed</th>
<th>Turbulence Series</th>
<th>Shear Series</th>
<th>Bulge Depth</th>
<th>Bulge Width</th>
<th>Bulge y Position</th>
<th>Yaw Error</th>
<th>Seed</th>
<th>Material Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.05</td>
<td>0.05</td>
<td>0</td>
<td>1.2</td>
<td>-1.8</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>0.11</td>
<td>0.15</td>
<td>0.3</td>
<td>2.0</td>
<td>-1.2</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>0.25</td>
<td>0.5</td>
<td>4.0</td>
<td>-0.6</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.8</td>
<td></td>
<td></td>
<td>0</td>
<td>3</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td>0.6</td>
<td></td>
<td>4</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td>1.2</td>
<td></td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td>1.8</td>
<td></td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 11: AWC Database:

<table>
<thead>
<tr>
<th>Wind Speed</th>
<th>Turbulence Series</th>
<th>Shear Series</th>
<th>Bulge Depth</th>
<th>Bulge Width</th>
<th>Bulge y Position</th>
<th>Yaw Error</th>
<th>Seed</th>
<th>Material Exponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.05</td>
<td>0.09</td>
<td>0</td>
<td>1.2</td>
<td>-1.8</td>
<td>-40</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>0.11</td>
<td>0.3</td>
<td>2.0</td>
<td>-1.2</td>
<td>-25</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.25</td>
<td>0.5</td>
<td>-0.6</td>
<td>-10</td>
<td>2</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>0.8</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td>0.6</td>
<td>10</td>
<td>4</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td>1.2</td>
<td>25</td>
<td>5</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.8</td>
<td>40</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Table 12: Signal table

<table>
<thead>
<tr>
<th>Signal</th>
<th>Location</th>
<th>Signal Information</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>Aerodynamic Power Extracted from the Air</td>
<td>W</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>Axial Aerodynamic force on the Rotor</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>Generator Power (After Subtracting losses)</td>
<td>W</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>Bending Moment in the Tower about the Inertial X Axis</td>
<td>N.m</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>Bending Moment in the Tower about the Inertial Y Axis</td>
<td>N.m</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>Torsional (yawing) moment in the tower</td>
<td>N.m</td>
</tr>
<tr>
<td>21</td>
<td>113</td>
<td>Bending Moment in the Tower about the Inertial X Axis</td>
<td>N.m</td>
</tr>
<tr>
<td>22</td>
<td>113</td>
<td>Bending Moment in the Tower about the Inertial Y Axis</td>
<td>N.m</td>
</tr>
<tr>
<td>51</td>
<td>0</td>
<td>Torque on the Drive Train Support</td>
<td>N.m</td>
</tr>
<tr>
<td>52</td>
<td>0</td>
<td>Tilting Moment on the drive train, about Y GL Axis</td>
<td>N.m</td>
</tr>
<tr>
<td>53</td>
<td>0</td>
<td>Yawing Moment on the Drive Train</td>
<td>N.m</td>
</tr>
<tr>
<td>54</td>
<td>0</td>
<td>Axial (compressive) force on the drive train</td>
<td>N</td>
</tr>
<tr>
<td>93</td>
<td>0</td>
<td>Demanded Torque for Generator</td>
<td>N.m</td>
</tr>
<tr>
<td>131</td>
<td>0</td>
<td>Blade 1 - Lead moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>132</td>
<td>0</td>
<td>Blade 1 - Torsional moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>-133</td>
<td>0</td>
<td>Blade 1 - Upwind moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>231</td>
<td>0</td>
<td>Blade 2 - Lead moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>232</td>
<td>0</td>
<td>Blade 2 - Torsional moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>-233</td>
<td>0</td>
<td>Blade 2 - Upwind moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>331</td>
<td>0</td>
<td>Blade 3 - Lead moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>332</td>
<td>0</td>
<td>Blade 3 - Torsional moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>-333</td>
<td>0</td>
<td>Blade 3 - Upwind moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>131</td>
<td>42</td>
<td>Blade 1 - Lead moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>132</td>
<td>42</td>
<td>Blade 1 - Torsional moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
<tr>
<td>-133</td>
<td>42</td>
<td>Blade 1 - Upwind moment in the blade (w.r.t. rotor plane reference system for straight blades)</td>
<td>N.m</td>
</tr>
</tbody>
</table>
In total, four binary files are created and required for correct operation of the database. These are described below.

**AWC Database Variables.bin/Reference Database Variables.bin**

This file contains the database variables in the following order (data type: 64 bit double), separated by negative 1000:

1. Wind Speed
2. Turbulence Series
3. Shear Series
4. Bulge Depth
5. Bulge Width
6. Bulge y Position
7. Yaw Error
8. Signal
9. Location
10. Material Exponent

For example, the file AWC_Database_Variables.bin would contain the following (spaces omitted) in binary format:

```
4 6 8 10 12 14 -1000 0.05 0.11 0.25 -1000 0.09 -1000 0.3 0.5 0.8 -1000 1.2 2.0 -1000 -1.8 -1.2 -0.6 0.6 1.2 1.8 -1000 -40 -25 -10 0 10 25 40 -1000 3 4 5 21 22 23 21 22 51 52 53 54 93 131 132 -133 231 232 -233 331 332 -333 131 132 -133 -1000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 42 42 42 -1000 2 3 4 5 6 7 8 9 10 11 12 13 14 -1000
```

**AWC Database.bin/Reference Database.bin**

These files contain the signal data generated from Phatas. This is written to and read out in a first-in/first-out method (FIFO) using a number of nested for loops. The order of the nested loops, from outer most loop to inner most loop is as follows:

1. Wind Speed
2. Turbulence Series
3. Shear Series
4. Bulge Depth
5. Bulge Width
6. Bulge y Position
7. Yaw Error
8. Signal/Location
9. Material Exponent

The number of elements for each loop is determined by the AWC_Database_Variables.bin/Reference_Database_Variables.bin files. All data is stored in 64 bit double format.

**C.3. Overview of the DLL**

The DLL is named Flow_Load_Database. This is essentially a function which is called in order to interpolated the file.
C.3.a. Required files

Files required for running of the DLL are identified below:
- Flow_Load_Database.dll
- Flow_Load_Database.h
- Flow_Load_Database.lib
- Flow_Load_Database_types.h
- rt_nonfinite.h
- rtwtypes.h

The header (*.h) may need to be modified before inclusion in Delphi (Farm Flow language).

C.3.b. Inputs to the DLL

The input to the DLL are in the following order:
1. Wind Speed (64 bit Double)
2. Bulge y Position (64 bit Double)
3. Bulge Width (64 bit Double)
4. Bulge Depth (64 bit Double)
5. Shear Series (64 bit Double)
6. Turbulence (64 bit Double)
7. Parameter7 (Unused) (64 bit Double)
8. Yaw Error (64 bit Double)
9. Interpolation Method (32 bit signed integer)
10. Signal/Location/Material Exponent (75 element array, 32 bit signed integers)

The program will determine the correct database to use. Furthermore, an error will be output if there are any problems with the data that was input. The error codes are discussed in more detail in a section which follows.

Signal/Location/Material Exponent

For the 9th input, Signal/Location/Material Exponent, the first 25 elements correspond to signals, the second 25 to locations and the third/last 25 to Material Exponents. If less than 25 combinations of signal/location and material exponent are required, the array should be set to zero in these elements. For example, if we require two data signals:

Signal/Location/Material Exponent = 3/0/4
Signal/Location/Material Exponent = 5/0/4

Signal_Location_Material_Exponent[1] = 3
Signal_Location_Material_Exponent[2] = 5
Signal_Location_Material_Exponent[26] = 0
Signal_Location_Material_Exponent[27] = 0

Signal_Location_Material_Exponent[51] = 4
Signal_Location_Material_Exponent[52] = 4

All other values of the array will be set to zero.
Interpolation Method
A further input is required to specify the interpolation method in the program. More details on the interpolation method are provided in the section which follows. Input integer 1 to the DLL in order to use the spline interpolation scheme. For the linear interpolation scheme, use any other integer value. The linear interpolation method is preferred and should be used as a default.

C.3.c. Outputs of the DLL

Outputs to the DLL are in the following order:
1. DEQL_LC_NOC (25 element array, 64 bit double)
2. Error_Number (12 element array, 32 bit unsigned integer)

C.4. Interpolation method

Matlab was chosen as the coding language for the DLL due to the presence of built in interpolation functions. Matlab contains functions for n-dimensional interpolation (interp1) as well as 1 dimensional (interp1) and 2 dimensional (interp2). Only the latter two methods are available for code conversion using the Matlab coder toolbox. For this reason, rather than performing a n-dimensional interpolation, seven single dimensional interpolations are performed sequentially. The order of the interpolation is as follows:
1. Wind Speed
2. Turbulence Series
3. Shear Series
4. Bulge Width
5. Bulge Depth
6. Bulge y Position
7. Yaw Error

The effect of changing the order of the interpolation is investigated in the Initial Results section. Also investigated is the difference in the results when comparing the interp1 method (single interpolation) and the interp1 method (7 sequential interpolations).

C.5. Error codes

The 12 element array output gives indication of errors in the data. Errors are indicated by 0 in the element, whereas a 1 in the element indicates no error.

- Error_Number[1] – Interpolation did not occur
- Error_Number[2] – No Material/Signal/Location in range
- Error_Number[3] – At least 1 Material Exponent not in range
- Error_Number[5] – File format of reference database is incorrect
- Error_Number[6] - Data is not in range of reference database
• Error_Number[7] – Are there equal signals and locations in reference database variable file
• Error_Number[8] – AWC database does not exist
• Error_Number[9] – File format of AWC database is incorrect
• Error_Number[10] - Data is not in range of AWC database
• Error_Number[11] – Are there equal signals and locations in AWC database variable file
• Error_Number[12] - At least 1 Material/Signal/Location not in range

Further to the error numbers, the output array DEQL_LC_NOC may give further indications to errors in the input data. Output of negative integers 1 to 4 indicate the database files do not exist or signal/location/material exponent data is not in range or corresponding to the database.
Appendix D. SwiftWake

SwiftWake is a modified version of Swift to generate a stochastic wind field of the wake using the results of FarmFlow. It was decided to update the wind farm software of ECN to include the wake meandering in the calculation of the loads of a wind turbine. A consequence of the wake meandering is that both SwiftWake and Phatas need to be modified.

In SwiftWake the possibility to select a wind field diameter different from the rotor diameter is added. Also time series of the wake meandering, displacement of the wake in lateral and vertical direction, is added.

Phatas is modified to read the wake displacement and to move the Swift wind field taking into account the wake meandering and its effect on the wind turbine loads.

D.1. SwiftWake modifications

In order to run SwiftWake for a wind field with a diameter larger than the rotor diameter the following modifications were done:

- A wind field diameter input variable is added;
- The (default) distance between the circles is estimated based on the wind field diameter instead of the rotor diameter;
- The velocities for grid points below the surface are modified;
- The value of the vertical length scale \( y_L \) used for the vertical velocity component \( w \) is modified.

In case the meandering of the wake is taken into account the centre of the wind field stays not at the rotor centre, but instead moves in the lateral and vertical direction. Because the rotor should be in the wind field during the complete simulation the diameter of the wind field needs to be larger than the rotor diameter. Therefore an input variable for the definition of the wind field diameter is added in SwiftWake.

In the previous version of SwiftWake the distance between the circles was estimated based on the number of circles required and the rotor diameter. Instead of using the rotor diameter from now on the diameter of the wind field is applied to determine the distance between the circles.

In general a stochastic wind field is generated with a cross sectional area equal to the rotor plane area and the centre of the wind field is located at hub height. To account for the wake meandering the diameter of the wind field is increased to two or three times the rotor diameter. In case the centre of the wind field remains at hub height this means that part of the wind field will be located below the ground surface level. Below surface there will be a conflict with the wind shear. The mean longitudinal wind speed in the grid points needs to be corrected. In SwiftWake the numbering of the grid points
is radial. When the position of a grid point \( i \) is identified to be below ground level. The velocity of this grid point \( i \) is set to the velocity of grid point \( i - 1 \), the previous grid point on the radial.

For the vertical velocity component \( w \) SwiftWake showed sensitivity for the size of the wind field diameter and the number of grid points on a circle. This resulted in a non-positive definite matrix causing an error during the Cholesky decomposition. Investigation revealed that the definition of the size of the vertical length scale \( y_{Lw} \) in relation to the wind field diameter caused the error. In SwiftWake the vertical length scale \( y_{Lw} \) is defined as:

\[
y_{Lw} = \min(0.35 \cdot z_m, 140)
\]

Where \( z_m \) is the hub height.

It was decided to depend \( y_{Lw} \) on the hub height \( z_m \) or the radius of the wind field \( w_r \) whatever is the largest. So

\[
\text{If } z_m \leq w_r \text{ then } \\
y_{Lw} = \min(0.35 z_m, 140) \\
\text{else} \\
y_{Lw} = \min(0.35 w_r, 140) \\
\text{endif}
\]

D.2. SwiftWake tests

Test for SwiftWake are defined to study the effect of changing the following parameters:

- Wind field diameter
- Mean length scale ratio \( \lambda_{ratm} \)
- Vertical Length scale \( y_{Lw} \)

Two tests load cases are defined with different wind field diameter. One load case, LCTest4, with a wind field of two times the rotor diameter and one load case, LCTest6, were the wind field has the same diameter as the rotor diameter. The wind fields have the same number of circles (18) and the same number of grid points on each circle (128).

The mean length scale ratio \( \lambda_{ratm} \) is the mean length scale over the wake field. This parameter is used to correct the mean wind speed at hub height \( v_g \) (10 m/s) and the longitudinal, lateral and vertical length scales \( X_{lc}, Z_{ly} \) and \( y_{Lw} \) respectively.

Finally the effect of a different vertical length scale is investigated. LCTest4 with the rotor diameter equal to the wind field diameter uses \( 0.35 \cdot \text{hub height} z_m \), while LTest4 will use \( 0.35 \cdot \text{wind field diameter} w_r \).
For the first grid point the mean in the longitudinal u direction is compared together with the standard deviations in the longitudinal, lateral and vertical velocities.

Test 1
In the first test the yLw is set 0.35 * rotor diameter for both load cases and vg (mean hub velocity), xlc, zlv and yLw are not corrected for the mean length scale in the wake. The correspondence between LCTest4 and LCTest6 looks good.

<table>
<thead>
<tr>
<th></th>
<th>Mean u</th>
<th>Std u</th>
<th>Std v</th>
<th>Std w</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCTest 4</td>
<td>8.964927307</td>
<td>1.827079885</td>
<td>1.4383734880110641</td>
<td>0.92656530</td>
</tr>
<tr>
<td>LCTest 6</td>
<td>8.964927429</td>
<td>1.827079867</td>
<td>1.4383734880110641</td>
<td>0.92656533</td>
</tr>
</tbody>
</table>

Test 2
Ylw =0.35 * hub height or 0.35* wind field radius whatever is largest. The mean wind speed vg and the length scales xlc, zlv and yLw are not corrected for the mean length scale in the wake. Compared to test 1 a small difference is observed in the standard deviation of the vertical velocity component w.

<table>
<thead>
<tr>
<th></th>
<th>Mean u</th>
<th>Std u</th>
<th>Std v</th>
<th>Std w</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCTest 4</td>
<td>8.964927307</td>
<td>1.827079885</td>
<td>1.4383734880110641</td>
<td>0.92656</td>
</tr>
<tr>
<td>LCTest 6</td>
<td>8.964927429</td>
<td>1.827079867</td>
<td>1.4383734880110641</td>
<td>0.92607</td>
</tr>
</tbody>
</table>

Test 3
Xlc is multiplied by lratm (mean of length scale over wind field area). Ylw =0.35 * hub height or 0.35* wind field radius whatever is largest. The mean wind speed vg and the length scales zlv and yLw are not corrected for the mean length scale in the wake. An increase in difference is now observed in the mean and standard deviation of the longitudinal component.

<table>
<thead>
<tr>
<th></th>
<th>Mean u</th>
<th>Std u</th>
<th>Std v</th>
<th>Std w</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCTest 4</td>
<td>8.964928</td>
<td>1.8246</td>
<td>1.4383734880110641</td>
<td>0.92656</td>
</tr>
<tr>
<td>LCTest 6</td>
<td>8.964926</td>
<td>1.8237</td>
<td>1.4383734880110641</td>
<td>0.92607</td>
</tr>
</tbody>
</table>

Test 4
Zlv multiplied by lratm. Ylw =0.35 * hub height or 0.35* wind field radius whatever is largest. The mean wind speed vg and the length scale yLw are not corrected for the mean length scale in the wake. The value of the standard deviation of the lateral component v now increases compared to the earlier tests.

<table>
<thead>
<tr>
<th></th>
<th>Mean u</th>
<th>Std u</th>
<th>Std v</th>
<th>Std w</th>
</tr>
</thead>
<tbody>
<tr>
<td>LCTest 4</td>
<td>8.964928</td>
<td>1.8246</td>
<td>1.4877</td>
<td>0.92656</td>
</tr>
<tr>
<td>LCTest 6</td>
<td>8.964926</td>
<td>1.8237</td>
<td>1.4869</td>
<td>0.92607</td>
</tr>
</tbody>
</table>

Test 5
yLw multiplied by lratm. Ylw =0.35 * hub height or 0.35* wind field radius whatever is largest. The mean wind speed is not corrected for the mean length scale in the wake. The length scale ratio has only a small effect on standard deviation w.
Mean u  Std u  Std v  Std w
LCTest 4   8.964928  1.8246  1.4877  0.9268 
LCTest 6   8.964926  1.8237  1.4869  0.9256 

Test 6
Finally the mean wind speed at hub height $v_g$ is multiplied by $\text{lratm}$ having a small effect on the statistics of the velocity time series at grid point 1.

Mean u  Std u  Std v  Std w
LCTest 4   8.964928  1.824  1.487  0.926 
LCTest 6   8.964923  1.823  1.486  0.9249 

General
To test whether SwiftWake can handle different wind field dimensions the following tests are performed.

- test 1 Diameter wind polar = rotor diameter = 128.3m
- test 2 Diameter wind polar = two times hub height = 180m
- test 3 Diameter wind polar = two times (grid height - hub height) = 212.6m
- test 4 Diameter wind polar = 2 times rotor diameter = 256.5m
- test 5 Diameter wind polar = 3 times rotor diameter = 384.8m

All tests run well.

D.3. SwiftWake meander

The phenomena of a wake not develop on a straight line behind the upwind turbine, but following a certain trajectory is called wake meandering.

DWM model
[Lar2008] has developed a dynamic wake meandering (DWM) model in recent years. In this model it is assumed that a wake released at an upwind turbine moves like a passive tracer on the large scale vortices of the ambient wind to the downwind wind turbine. Risoe applies a low-pass filter, exponential moving average, to select the large scale vortices and defines the following low pass frequency $f_c$:

$$f_c = \frac{U}{2D_w}$$

Where $U$ is the mean longitudinal velocity and $D_w$ the instantaneous wake deficit diameter. GL Garrad Hassan [Schmidt2011] applies the rotor diameter. In this report the rotor diameter is used.
Figure 18: Filtered and unfiltered (original) Lateral velocity of wind at hub height in single wake.

![Lateral Wind Velocity Graph]

Figure 19: Filtered and unfiltered (original) vertical velocity of wind at hub height in single wake.

![Vertical Wind Velocity Graph]
In the DWM model the meandering displacement is based on the assumption that each time step a wake deficit is released within a frozen turbulence field [Schmidt2011]. For the lateral and vertical position displacement at downwind position \( L_d \) this means:

\[
\begin{align*}
(2) y(i) &= v(i) \left( \frac{L_d}{u} \right) \\
(3) z(i) &= w(i) \left( \frac{L_d}{u} \right)
\end{align*}
\]

This implies that meandering is dependent on the distance behind the upwind turbine, which makes sense from a physical point of view, but will add the requirement for another dimension to the database (distance behind the upwind turbine).

**ECN implementation**

The mean wake conditions are calculated at the position of the downwind turbine using FarmFlow. The results of the FarmFlow calculation are used as input for SwiftWake.

Unlike Risoe, ECN uses the wind velocity at hub height in the wake to filter the wake meandering instead of using the ambient wind velocity at hub height. The Swift wind velocity at the hub is the mean of the time series of the gridpoints at hub height. The wind time series are filtered by ECN using a so called second order Butterworth filter. The filter is implemented using Fortran 1990 code by J-P Moreau. Figure 18 and Figure 19 show the filtered and unfiltered velocity time series in the lateral and vertical direction. The SwiftWake wind field is generated for a single wake condition with an ambient hub velocity \( v_{hub} = 10 \text{ m/s} \) and a turbulence intensity \( I = 16\% \).

A method to reduce complexity has been investigated. Assuming that the wake at the downwind turbine moves with the large scale vortices of the ambient wind results in the following lateral and vertical displacements of the wake (integration of the filtered lateral and vertical velocities):

\[
\begin{align*}
y(i + 1) &= y(i) + (v(i + 1) - v(i))\Delta t \\
z(i + 1) &= z(i) + (w(i + 1) - w(i))\Delta t
\end{align*}
\]

Figure 20 and Figure 21 show that between the SwiftWake displacements there is quite a difference with the DMW model. This can (partly) be explained because the applied integration using the trapezium rule can be seen as an (additional) filter.
Figure 20: Lateral displacement of wake for SwiftWake and DWM model.

Figure 21: Vertical displacement of wake for SwiftWake and DWM model.
Applying the DWM model results in the behavior as shown in Figure 22 and Figure 23. However, this does require an additional dimension in the loads database, which will blow up the number of case to be calculated. Currently, there is too much uncertainty on the correct way of modelling wake meandering; it should be confirmed by measurements.

**Figure 22**: Wake meandering as it develops downstream

![Graph showing wake meandering as it develops downstream](image)
Figure 23: Wake meandering (lateral displacement) at 10D behind the upwind rotor