Wind Farm Decentralized Dynamic Modeling With Parameters

AEOLUS EU-report, Deliverable 2.3

Soltani, Mohsen; Shakeri, Sayyed Mojtaba; Grunnet, Jacob Deleuran; Knudsen, Torben; Bak, Thomas

Publication date:
2010

Document Version
Early version, also known as pre-print

Link to publication from Aalborg University

Citation for published version (APA):
Soltani, M., Shakeri, S. M., Grunnet, J. D., Knudsen, T., & Bak, T. (2010). *Wind Farm Decentralized Dynamic Modeling With Parameters: AEOLUS EU-report, Deliverable 2.3*. http://ict-aeolus.eu/publications.html

AALBORG UNIVERSITY
DEPARTMENT OF ELECTRONIC SYSTEMS
AUTOMATION AND CONTROL SECTION

Fredrik Bajersvej 7C ▪ DK-9220 Aalborg East Phone 99 40 87 44

Wind Farm Decentralized Dynamic Modeling With Parameters

Aeolus Deliverable (D2.3)

Mohsen Soltani, Sayyed Mojtaba Shakeri, Jacob Grunnet, Torben Knudsen, Thomas Bak

May 25, 2010

Summary

Development of dynamic wind flow models for wind farms is part of the research in European research FP7 project AEOLUS. The objective of this report is to provide decentralized dynamic wind flow models with parameters. The report presents a structure for decentralized flow models with inputs from a set of spatially distributed measurements from wind turbines. The information has to be communicated only within neighboring wind turbines. This will both reduce the calculation load by distributing them on all turbines and make the infrastructure more robust against faults and uncertainties. Moreover, the presented flow model is formulated such that it is able to update the parameters in an adaptive on-line procedure. The report also investigates the effect of wind direction changes on the flow model and proposes a fusion algorithm which improves the wind predictions by fusing the predictions of available local models. The results of this report are especially useful, but not limited, to design a decentralized wind farm controller, since in centralized controller design one can also use the model and update it in a central computing node.

Contents

Chapter 1

Introduction

This report is deliverable 2.3 of the European FP7 project *Distributed Control of Large-Scale Offshore Wind Farms* with the acronym *Aeolus*. Part of the research in Aeolus deals with development of models that allow real-time predictions of flows and incorporate measurements from a set of spatially distributed sensor devices. In Aeolus we use the flow information as a basis for new control paradigms that acknowledges the uncertainty in the modeling and dynamically manages the flow resource in order to optimize specific control objectives.

Work package 2 of Aeolus aims to provide dynamic models based on the use of control engineering methods such as parameter estimation. To support the distributed control approach, distributed estimation methods are developed where only local measurements are processed at each turbine and only limited information is communicated with neighbors. This report contributes to two tasks of WP2 in Aeolus:

- **T2.2 Methodology for decentralized dynamic modeling**. Selection of the model structure with input from network of sensor information and output relevant for the control. A nonlinear time-varying state-space model is a candidate for the model structure. To support the decentralized control paradigm a distributed estimation and prediction methodology is investigated where only limited information is exchanged between neighboring turbines. Most of the data processing is then located at the single turbine which provides scalability.
- **T2.3 Parameter estimation using simulated and farm measurements**. A method for adaptive parameter estimation must be developed or chosen. The method should include uncertainty. The estimation must eventually be based on real farm data but is initially based on a simulation model.

The works presented in this report aim to complete the recent report (Knudsen and Soltani, 2009) on the above tasks. In (Knudsen and Soltani, 2009), data from OWEZ wind farm has been used to obtain a dynamic flow model from an upwind to a downwind turbine's Effective Wind Speed (EWS). EWS is the wind speed averages over the rotor disc (Østergaard et al., 2007; Leithead, 1992). The presented model is then useful when the turbines are operating at a fixed power reference (Power reference in single turbine operation is set to nominal power). In this report it is also investigated how the change of power reference will affect the downwind turbine operation. In fact, the model has been improved for a wind farm which is subject to changes by power reference from the wind farm controller (Contribution to T2.2).

The flow model is adaptive. This means that if the wind and/or turbine characteristics changes due to some uncertainties in the system, then the model parameters will be updated to achieve the best predictions in the new situation (Contribution to T2.3).

Furthermore, the model is improved so that it provide a online fusion of the wind speed predictions at downwind turbines. This is useful in many cases even if the downwind turbine is not located in the wake of upwind turbines but still is able to predict the wind speed using the measurements of the neighboring upwind turbines (Contribution to Aeolus).

The models are finally obtained based on the Aeolus simulation model (see (Soltani et al., 2009)) where the Aeolus benchmark in (Soltani et al., 2010) has been used as the layout of the wind farm when the decentralized wind flow model is obtained (Contribution to T2.3).

This report is organized as follows: chapter 2 formulates the problem for the decentralized wind flow model with parameter estimation. Both online and off-line wind flow models are presented in chapter 3. The predictions from on-line models have been fused to obtain prediction of wind at downwind turbines in chapter 4 where the Aeolus decentralized benchmark model is presented and chapter 5 brings the conclusions of this report.

Chapter 2

Problem Formulation

This report is part of the work package 2 in Aeolus which provides dynamic models of wind flow in wind farms. The results of this report are formulated in a way which are useful for decentralized wind farm control design in work package 4. In fact, the wind flow dynamic model has to be formulated in a decentralized way which exploits that each wind turbine is affected by its *upwind* neighboring wind turbines. Thus the wind flow will be modeled locally from turbine to turbine. The resulting wind flow model has the following characteristics:

- Model parameters are updated in an adaptive on-line procedure. Due to the fact that the wind flow model changes due to different low frequency effects such as mean wind speed and mean wind direction, these parameters have to be adapted to the new situation using an on-line parameter estimation algorithm.
- Since the wind experienced by the upwind turbines, to some extend, will be experienced by the downwind turbines, the output of the model can be used for prediction of the wind at downwind turbines.
- For each turbine inside the wind farm, several turbines might be laterally distributed around the upwind direction. Hence, the wind speed could be predicted from each of the upwind turbines. Then a sensor fusion algorithm is able to use all these predictions to provide a more precise prediction in many situations. These situations are very dependent on the layout of the wind farm and the mean wind direction. For example, if there is an upwind turbine exactly against the mean wind direction, then the predictions from other laterally distributed turbines are not so useful in sensor fusion, but if the mean wind is blowing in between two turbines toward the downwind turbine, then the sensor fusion algorithm can provide more precise predictions of wind.

2.1 Online Parameter Estimation

Assume the discrete time AutoRegressive with eXternal input (ARX) model from upwind turbine's *Effective Wind Speed* (EWS) and power reference P_{ref} to a downwind turbine EWS is given by

$$A(q)y(t) = \sum_{i=1}^{2} q^{-n_{k_i}} B_i(q)u_i(t) + e(t), \tag{2.1}$$

where q^{-1} is the delay operator, t is a sample time, n_{k_i} is the number of delay samples, $A(q)$ and $B_i(q), \quad i = 1, 2.$ are polynomials of q^{-1} with orders of n_a and $n_{b_i} - 1$ and parameters of $1, \quad a_1, \quad a_2, ..., a_{n_a}$ and $b_{i0}, \quad b_{i1}, \quad b_{i2}, ..., b_{in_{b_i}-1}$ respectively, $y(t)$ is the EWS at downwind turbine (EWS is estimated at turbine using Extended Kalman Filter (EKF) in (Knudsen and Soltani, 2009)), $e(t)$ is white noise, and $u_1(t) = u_{EWS}(t)$ is EWS at upwind turbine and $u_2(t) = u_{Pref}(t)$ is the power reference at upwind turbine.

An on-line parameter estimation algorithm for $\theta = [a_1, \quad a_2, ..., a_{n_a}, \quad b_{10}, \quad b_{11}, \quad b_{12}, ...$ $, \quad b_{1n_{b_1}-1}, \quad b_{20}, \quad b_{21}, \quad b_{22}, ..., b_{2n_{b_2}-1}]$ can be formulated by (See (Ljung, 1999))

$$\begin{aligned} X(t) &= \mathbf{H}(X(t-1), y(t), u(t), t) \\ \hat{\theta}(t) &= \mathbf{h}(X(t)), \end{aligned} \tag{2.2}$$

where X is the *information state vector*, **H** and **h** specify the state-space representation of the recursive parameter estimation algorithm, and $\hat{\theta}(t)$ is the vector of the updated parameters (coefficient vector of A and B polynomials) at time t.

2.2 Prediction and Fusion

The one-step ahead predictor for (2.1) corresponds to

$$\begin{aligned} \hat{y}(t|t-1) &= -a_1 y(t-1) - a_2(t-2) - ... - a_{n_a} y(t-n_a) + \sum_{i=1}^{2} q^{-n_{k_i}} B_i(q)u_i(t) \\ &= (1 - A(q))y(t) + \sum_{i=1}^{2} q^{-n_{k_i}} B_i(q)u_i(t). \end{aligned} \tag{2.3}$$

In the same manner, the k-step ahead predictor can be obtained by assuming that the measurements are only available until time $t - k$. Thus the predictions of y should be used from time $t - k + 1$ to time $t - 1$. These predictions can be obtained recursively starting from $\hat{y}(t - k + 1|t - k)$. Then the k-step ahead prediction will be given by

$$\begin{aligned} \hat{y}(t|t-k) &= -a_1 \hat{y}(t-1|t-k) - a_2 \hat{y}(t-2|t-k) - ... - a_{k-1} \hat{y}(t-k+1|t-k) \\ &- a_k y(t-k) - ... - a_{n_a} y(t-n_a) + \sum_{i=1}^{2} q^{-n_{k_i}} B_i(q)u_i(t). \end{aligned} \tag{2.4}$$

It is also worth to mention that in the wind flow model, traveling of the flow from turbine to turbine causes a delay on the observation of the wind from upwind turbine to the downwind turbine. This delay can then be used for k-step ahead prediction of wind where k represents the delay.

In the case that there are several upwind neighboring turbines which are laterally distributed against the wind direction, we might have different predictions of EWS at the downwind turbine. It is then possible to fuse the predictions from each model. The fusion algorithm can then update a gain on the predictions of each model according to the prediction error of each model. A straight-forward fusion algorithm is simply to take the variance weighted average of predicted EWSs and obtain the global prediction. To make the weightings more adaptive, fusing is simply done by using recursive algorithm. The main advantages of this approach is simplicity. Moreover, the effect of wind direction is inherently included in the weightings. The local EWS estimates are assumed as inputs of dynamical model whose output is global EWS estimate at considered wind turbine. This model is updated by using the previous recursive algorithm. The fusion algorithm can then be formulated as

$$y_g(k+1|k) + a_{f_1} y_g(k|k-1) = \sum_{i=1}^{N} b_{f_i} \hat{y}_i(k+1|k) + e(t) \tag{2.5}$$

where N is the number of neighboring upwind turbines, $\hat{y}_i$ is the local EWS prediction using i^{th} wind turbine data and y_g is the global EWS prediction at considered downwind turbine, $e(t)$ is the error term, which is assumed to be white noise, and a_{f_1} and b_{f_i} are fusion parameters to be updated recursively.

Chapter 3

Local Wind Flow Model

This chapter continues the previous work in WP2 of Aeolus where the model from upwind to downwind effective wind speed was obtained. It is investigated how to implement the model based on on-line parameter estimation and then the effect of the variations of the upwind turbine power reference is used to improve the predictability of the model. The models are obtained for two different situations regarding wind direction. They are developed based on the generated data from the Aeolus simulation model where there are only two turbines one upwind and one downwind. Compare with section 4.2 where the results from simulation of several turbines are explained. The wind field generated for these simulations includes turbulence and wake meandering. However, it also relies on the hypothesis of frozen turbulence, which makes it not entirely realistic.

3.1 Downwind EWS model

In (Knudsen and Soltani, 2009), the SISO model from an upwind to downwind turbine wind speed have been analyzed. The results show that the models which use effective wind speed have higher fitness and lower prediction error norm compared to the models which use nacelle wind speeds as input-outputs. This certainly concurs with the following physical fact.

> *"The nacelle measurement averages the wind speed on a small area and therefore is highly affected by the high frequency content of the wind speed. The high frequency contents of the wind are less correlated when the distance between the measurement points is long, e.g., distance between wind turbines in a wind farm. The effective wind speed, instead, is an average of the wind speed on the rotor disc. The effective wind speed estimator acts like a large scale anemometer to estimate the wind speed avraged on a broad area of turbine rotor disc. It includes the low frequency content of the wind speed and therefore the effective wind speeds of two points within long distances in wind farm are highly correlated. "*

Two general and simple cases have to be analyzed at this stage. Consider two turbines in a wind farm and consider that in one experiment the wind direction is in line with turbines, i.e., wind direction is parallel to the wind turbine row (See figure 3.1-A). In the other experiment, the wind direction hits the turbine row with constant angle γ as shown in figure 3.1-B. The goal here is to obtain a SISO model from EWS at upwind turbine to the downwind turbine in each case based on the simulated data from Aeolus simulation model (Soltani et al., 2009) [1]. The distance between two turbines is 600 meters in the first experiment and 583 meters in the second one while γ is 0 and 59 degrees respectively. Average wind speed is $15\frac{m}{s}$ and turbulence intensity is 10%. In the following, it is explained how these models are achieved in an offline and online manner.

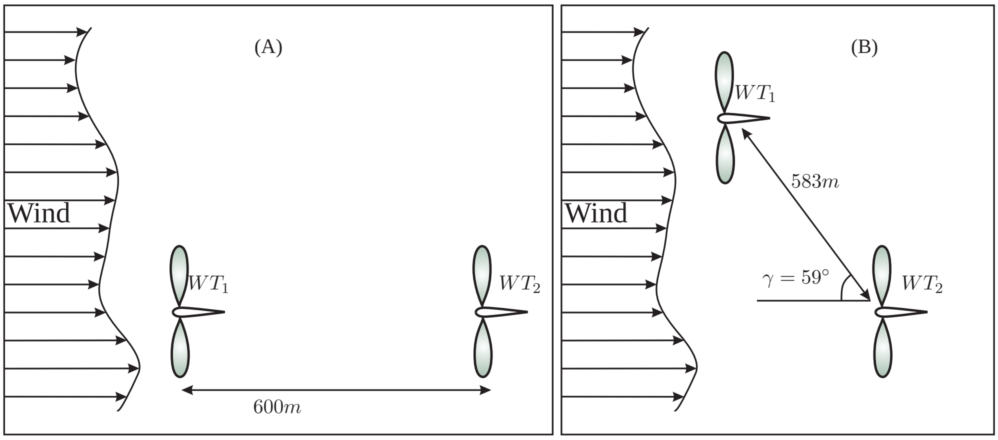

Figure 3.1: Layout of the experiments: (A) parallel to the wind direction and (B) with angle γ.

3.1.1 Off-line LTI models and prediction

Consider the LTI models described in (Knudsen and Soltani, 2009)[2] by

$$\begin{aligned} y(t) &= G(q^{-1})u(t-n_k) + H(q^{-1})e(t) \ \ , \ \ e(t) \in \text{ID}(0, \sigma^2) \\ G(q^{-1}) &= \frac{B(q^{-1})}{A(q^{-1})F(q^{-1})} \ \ , \ \ H(q^{-1}) = \frac{C(q^{-1})}{A(q^{-1})D(q^{-1})} \end{aligned} \tag{3.1}$$

[1] Notice that the experiments are done on the modified wind field generation where the coherence between lateral channels has been scaled. At the time of writing there is ongoing investigation on the most appropriate coherence between lateral channels. Consequently there is some uncertainty related to the coherence used here.

[2] Notice that in the current report the order of the polynomial $B(q)$ is $n_b - 1$ which is different than the notation in the reference. This modification is to support The System Identification Toolbox in Matlab

where ID denotes independent distribution, and

$$
\begin{aligned}
B(q^{-1}) &= b_0 + b_1 q^{-1} + \cdots + b_{n_b - 1} q^{-n_b + 1} = \sum_{i=0}^{n_b - 1} b_i q^{-i} \\
A(q^{-1}) &= \sum_{i=0}^{n_a} a_i q^{-i}, \;\; a_0 = 1 \\
C(q^{-1}) &= \sum_{i=0}^{n_c} c_i q^{-i}, \;\; c_0 = 1 \\
D(q^{-1}) &= \sum_{i=0}^{n_d} d_i q^{-i}, \;\; d_0 = 1 \\
F(q^{-1}) &= \sum_{i=0}^{n_f} d_i q^{-i}, \;\; f_0 = 1
\end{aligned}
\tag{3.2}
$$

Table 3.1 shows the model order as well as fitness and the norm of prediction error for different identified models corresponding to the experiment (A). The following models are used for identification: Output Error (OE), AutoRegressive (AR), AutoRegressive with eXternal input (ARX), Box-Jenkins(BJ)and persistence (Per). These models are well explained in (Ljung, 1999) and specially in this case in Knudsen and Soltani (2009). The first half of data sets are used for obtaining the models and the second half used for validation of fit and standard prediction error estimate (RMS). A model order of 3 is used for all model structures discussed above. The reasons are that third order models can improve the fitness better than the second order and there seems to be no substantial improvement from using higher order models. The delay is assumed to be fixed. It could be defined by calculating the time which takes the mean wind to travel between turbines.

The best results are achieved using ARX and BJ models with delay (ArxDel and BJDel) which confirms the propagation of wind toward downwind turbine in the simulation program. The prediction errors for 40sec horizon in both models are limited to less than $0.32m/s$ corresponding to the 95% significance level of ± 0.64 which is small compared to the mean wind speed of $15m/s$. For 20sec predictions, the BJDel model shows slightly higher fitting (84%) and lower prediction error $0.25m/s$ compared to the ArxDel model (81% and $0.28m/s$ respectively). However, for prediction horizons more than 20sec, the fit value for the BJDel model decreases and prediction error estimate increases while the ArxDel model shows almost the same fit and RMS for longer prediction horizons. The infinite prediction horizon gives a pure simulation from the input only. It means that the prediction of the output does not use any information of the past outputs, i.e. there is no correction from the measurement of the output.

Table 3.2 corresponds to the experiment (B) for system identification. The best results are again obtained from delay augmented models (ArxDel and BJDel) where the predicted values for the horizon of 20 sec[3] are about 71% and 73% respectively and the prediction error doesn't

[3] The reason to look at prediction horizon of 20 sec is that in the experiment (B) the delay for the wind to travel between two turbines is 20 sec which is due to the different direction of the wind.

Models						
	OE	Arx	ArxDel	BJDel	Ar	Per
n_a	0	3	3	0	3	1
n_b	3	3	3	3	-	-
n_c	0	0	0	3	0	0
n_d	0	0	0	3	0	0
n_f	3	0	0	3	-	-
n_k	0	0	40	40	-	-
Fit (%)						
Pred. hor.						
1 sec.	22.019	51.526	94.005	94.796	50.974	50.905
20 sec.	22.019	-7.665	81.435	83.547	-29.301	-30.560
40 sec.	22.019	-5.331	81.418	78.753	-39.722	-38.904
∞	22.019	-4.129	81.418	-1.918	-	-
RMS (m/s)						
Pred. hor.						
1 sec.	1.166	0.725	0.089	0.077	0.733	0.734
20 sec.	1.166	1.610	0.277	0.246	1.933	1.952
40 sec.	1.166	1.575	0.277	0.317	2.089	2.077
∞	1.166	1.557	0.277	1.524	-	-

Table 3.1: Downwind EWS predictability in the experiment (A) using LTI models and upwind turbine EWS data. The model structures marked with “del” includes the 40 sec. delay.

Fit (%)						
Pred. hor.	OE	Arx	ArxDel	BJDel	Ar	Per
1 sec.	16.8012	52.138	78.386	79.189	50.974	50.905
10 sec.	16.8012	6.794	71.419	74.165	-14.096	-15.649
20 sec.	16.8012	11.946	71.342	73.185	-29.301	-30.560
∞	16.8012	15.404	71.341	72.399	-	-
RMS (m/s)						
Pred. hor.						
1 sec.	1.2444	0.7159	0.3233	0.3112	0.7332	0.7343
10 sec.	1.2444	1.3940	0.4275	0.3864	1.7065	1.7297
20 sec.	1.2444	1.3170	0.4286	0.4010	1.9339	1.9527
∞	1.2444	1.2653	0.4286	0.4128	-	-

Table 3.2: Downwind EWS predictability in the experiment (B) using LTI models and upwind turbine EWS data.

exceed $0.43m/s$ corresponding to significance level of ± 0.86 which is also small compared to the mean wind speed at $15m/s$.

The comparison of the results in table 3.1 and table 3.2 shows that the models in the experiment (A) can better describe the flow between two turbines, i.e., the predictions of EWS have higher fit and lower norm of prediction error. This is also according to what is expected since *"the downwind turbine in the experiment (A) would certainly experience most of the wind experienced by the upwind turbine"*.

3.1.2 Online LTI model and prediction

Many real-world applications, such as adaptive control, adaptive filtering, and adaptive prediction, require a model of the system to be available on-line while the system is in operation. Due to the time varying properties of the wind, using recursive identification method to adapt the model parameter at each time step is likely to be relevant. In recursive modeling, parameter estimates are computed recursively in time. In this regard, some algorithms such as the Kalman Filter (KF), recursive least square with forgetting factor (RLS), and least mean square (LMS) can be used. The last two may be viewed as special cases of the KF (Ljung and Gunnarsson, 1990).

The general recursive identification algorithm is given by the following equation:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)(y(t) - \hat{y}(t)) \tag{3.3}$$

$\hat{\theta}(t)$ is the parameter estimate at time t. $y(t)$ is the observed output at time t and $\hat{y}(t)$ is the prediction of $y(t)$ based on observations up to time $t-1$. The gain, $K(t)$, determines how much the current prediction error $y(t) - \hat{y}(t)$ affects the update of the parameter estimate. The estimation algorithms minimize the prediction-error $y(t)-\hat{y}(t)$ term. The gain has the following general form:

$$K(t) = Q(t)\psi(t) \tag{3.4}$$

The recursive algorithms differ based on different approaches for choosing the form of $Q(t)$ and computing $\psi(t)$, where $\psi(t)$ represents the gradient of the predicted model output $\hat{y}(t|\theta)$ with respect to the parameters. The simplest way to visualize the role of the gradient $\psi(t)$ of the parameters, is to consider models with a linear-regression form:

$$y(t) = \psi^T(t)\theta_0(t) + e(t) \tag{3.5}$$

In this equation, $\psi(t)$ is the *regression vector* that is computed based on previous values of measured inputs and outputs. $\theta_0(t)$ represents the true parameters. $e(t)$ is the noise source (innovations), which is assumed to be white noise (Ljung, 1999). For linear regression equations, the predicted output is given by the following equation:

$$\hat{y}(t) = \psi^T(t)\hat{\theta}(t-1) \tag{3.6}$$

The well-known Kalman Filter (KF) algorithm is used to define $Q(t)$. It is assumed that the true parameters are described by a random walk. It is a default description for the parameter variation when no specific information at hand. It could be formulated as follows:

$$\theta_0(t) = \theta_0(t-1) + \omega(t) \tag{3.7}$$

where $\omega(t)$ is Gaussian white noise with the covariance matrix R_1. This covariance matrix determines the rate of changes in parameters. The following set of equations summarizes the Kalman filter adaptation algorithm:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)(y(t) - \hat{y}(t)) \tag{3.8a}$$

$$\hat{y}(t) = \psi^T(t)\hat{\theta}(t-1) \tag{3.8b}$$

$$K(t) = Q(t)\psi(t) \tag{3.8c}$$

$$Q(t) = \frac{P(t-1)}{R_2 + \psi^T(t)P(t-1)\psi(t)} \tag{3.8d}$$

$$P(t) = P(t-1) + R_1 - \frac{P(t-1)\psi(t)\psi^T(t)P(t-1)}{R_2 + \psi^T(t)P(t-1)\psi(t)} \tag{3.8e}$$

where R_2 is the variance of the innovations $e(t)$. It can be shown that if R_1 and R_2 are chosen as below then this special case of the KF will be equivalent to RLS with forgetting factor $|\lambda(t)| \leq 1$ (Ljung and Gunnarsson, 1990).

$$\hat{R}_1(t) = \left(\frac{1}{\lambda(t)} - 1\right) \times \left[P(t-1) - \frac{P(t-1)\varphi(t)\varphi^T(t)P(t-1)}{\lambda(t) + \varphi^T(t)P(t-1)\varphi(t)}\right] \tag{3.9a}$$

$$\hat{R}_2(t) = \lambda(t) \tag{3.9b}$$

Now assume that the model structure has the following form:

$$y(t) + a_1 y(t-1) + ... + a_{n_a} y(t-n_a) = b_1 u(t-n_k) + ... + b_{n_b} u(t-n_b-n_k) + e(t) \tag{3.10}$$

where $e(t)$ denotes an equation error, which can describe disturbances or un-modeled dynamics. For the sake of simplicity single input single output (SISO) model is assumed here. The model (3.10) can be equivalently expressed as the linear regression form (3.5) with:

$$\begin{aligned} \psi^T(t) &= [-y(t-1)... - y(t-n_a) \;\; u(t-n_k)...u(t-n_b-n_k)] \\ \theta_0(t) &= [a_1...a_{n_a} \;\; b_1...b_{n_b}]^T \end{aligned} \tag{3.11}$$

The parameter vector $\theta_0(t)$ is computed at each step by using the KF algorithm, and then model (3.10) could be updated.

The above algorithm is applied to both experiment. The model which updated recursively is of the form (3.10) with $n_a = n_b = 3$. The forgetting factor λ is 0.996 and the input delays are 40 and 20 for experiment (A) and (B) respectively. The results are shown in table 3.3 and table 3.4.

Pred. hor.	1 sec.	20 sec.	40 sec.
Fit (%)	94.007	81.185	80.733
RMS (m/s)	0.089	0.279	0.285

Table 3.3: Downwind EWS predictability in the experiment (A) using online LTI models and upwind turbine EWS data.

Pred. hor.	1 sec.	10 sec.	20 sec.
Fit (%)	78.260	71.838	71.522
RMS (m/s)	0.322	0.417	0.422

Table 3.4: Downwind EWS predictability in the experiment (B) using online LTI models and upwind turbine EWS data.

The comparison of tables 3.3 and 3.4 and corresponding offline results in tables 3.1 and 3.2 shows that the recursive method gives similar accuracy. Notice that the wind characteristics such as mean wind speed would potentially have significant effect on model parameters, however, in these experiments, the variations of the wind is limited to high frequency turbulences and therefore variations of the parameters in online model are negligible (See figure 3.2). Consequently, this will result in similar accuracy of the model for both online and offline methods.

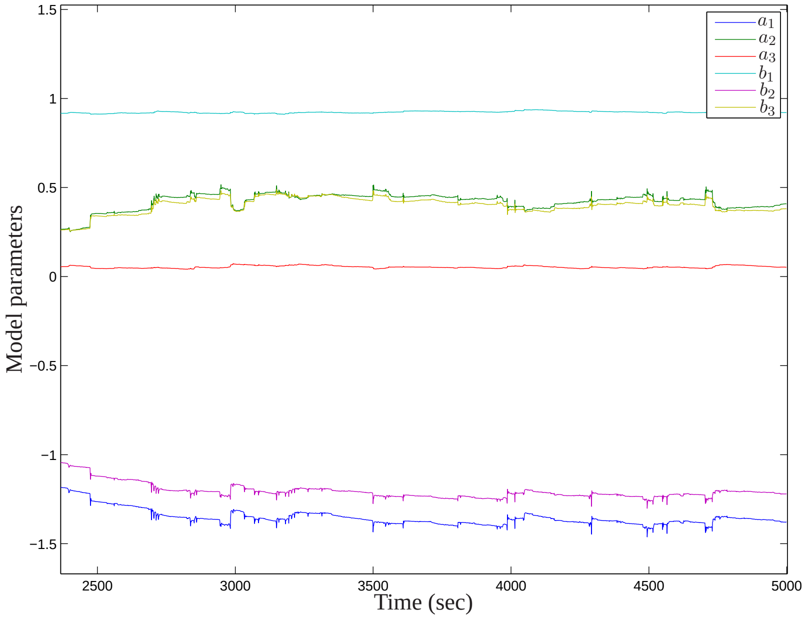

Figure 3.2: Model parameters in the experiment (A) using recursive model

3.2 Downwind power reference/EWS model

The models presented in the last section are useful when the turbines are working with a fixed power reference. The operating points of wind turbine could be determined by knowing the mean wind speed and the power reference. Assuming that the mean wind speed is constant during simulation/experiment, the power reference would change the turbine state and therefore the wake behind the turbine. This implies the model to be different when the wind farm controller changes the power reference. The model then should be able to show the effect of the change in power reference at downwind turbine.

To identify such model, we have to excite the power reference at the upwind turbine. The test signal for power reference has to be persistently exciting which means that the input has to be informative in a way that it include enough different frequencies which might be seen as eigenvalues of the system.

In the experiments, the power reference input signal is generated as a random binary signal which is scaled and biased to obtain a value of either 1MW or 5MW with the cut-off frequency of $0.1Hz$ as shown in figure 3.3-(b).

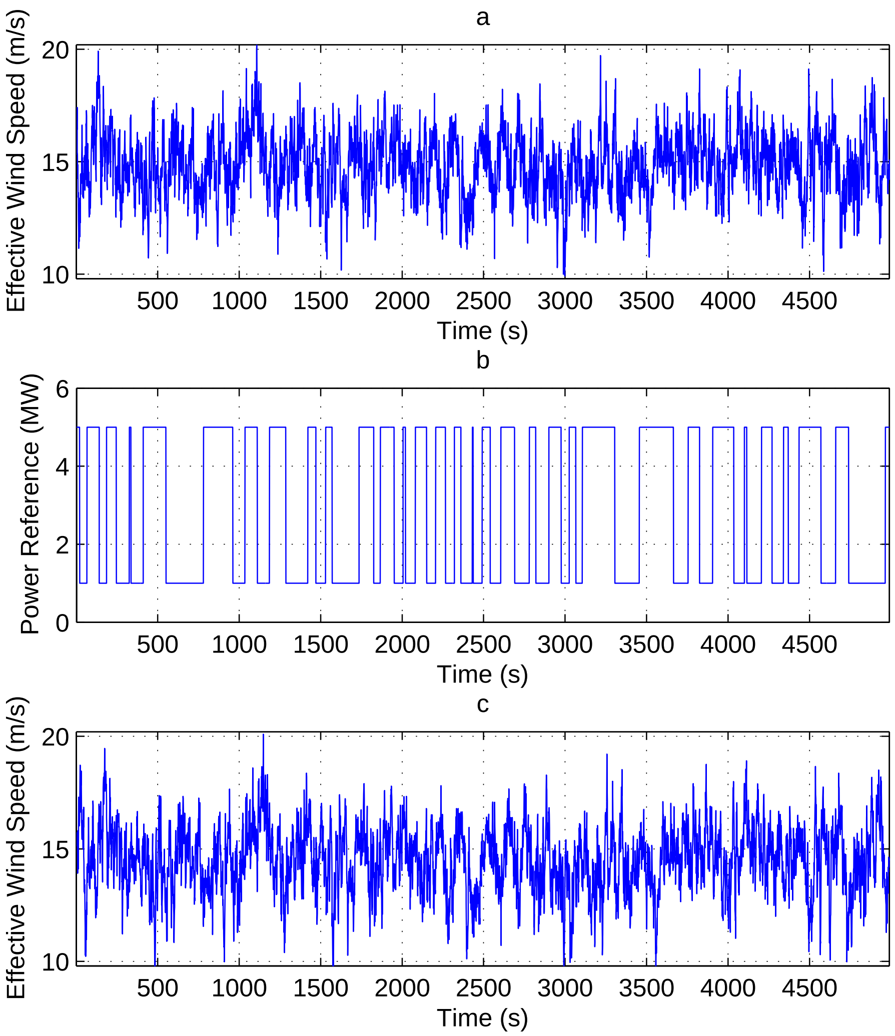

Figure 3.3: (a) EWS at WT_1, (b) P_{ref} excitations on WT_1, and (c) EWS at WT_2, in the experiment (A).

3.2.1 Offline LTI models and prediction

Table 3.5 shows the results of predictions using different models and both EWS and P_{ref} data from the upwind turbine. The best result obtained from the ArxDel and the BJDel models where the fit values for prediction horizon of 40 sec are about 90% and 92% corresponding to the prediction error norm of $0.15m/s$ and $0.114m/s$ respectively. The BJDel model has a higher fit and lower prediction error at 40 sec horizon compared to the ArxDel model, but its fitting drops for longer prediction horizons while the ArxDel model seems to stay at the same level of fitting for infinity horizon.

Fit (%)						
Pred. hor.	OE	Arx	ArxDel	BJDel	Ar	Per
1 sec.	22.706	51.499	94.793	95.402	50.975	50.905
20 sec.	22.706	-7.600	89.939	92.357	-29.302	-30.560
40 sec.	22.706	-5.152	89.942	92.351	-39.722	-38.904
∞	22.706	-3.933	89.942	18.064	-	-
RMS (m/s)						
Pred. hor.						
1 sec.	1.156	0.725	0.078	0.069	0.733	0.734
20 sec.	1.156	1.609	0.150	0.114	1.934	1.953
40 sec.	1.156	1.573	0.150	0.114	2.090	2.078
∞	1.156	1.554	0.150	1.225	-	-

Table 3.5: Downwind EWS predictability in the experiment (A) using LTI models and upwind turbine EWS and P_{ref} data.

A comparison of the ArxDel model in table 3.1 and table 3.5 shows that inclusion of P_{ref} has a significant effect on achieving a model with higher degree of explanation. For example, the fitting level for prediction horizon of 40 sec has been increased from 81% to about 90% as the prediction error has been decreased from $0.277m/s$ to $0.15m/s$.

Table 3.6 shows the results of predictions for different models while the upwind turbine is not in line with the downwind turbine regarding the wind direction. The best results are obtained using the ArxDel and the BJDel models for a prediction horizon of 20 sec. A comparison of the results of this table to the results of table 3.2 shows that inclusion of P_{ref} in this case did not make any significant change in the predictability of EWS at downwind turbine. This complies with the physical fact that "*when the downwind turbine is not in the wake of upwind turbine, the change in the power reference at upwind turbine will not contribute to the wind experienced at downwind turbine. The EWS at upwind turbine instead can be used to predict the wind since effective wind speeds at neighboring turbines are highly correlated.*"

Fit (%)						
Pred. hor.	OE	Arx	ArxDel	BJDel	Ar	Per
1 sec.	15.049	52.133	78.371	79.157	50.975	50.905
10 sec.	15.049	6.775	71.452	74.114	-14.096	-15.649
20 sec.	15.049	11.883	71.376	73.192	-29.302	-30.560
∞	15.049	15.316	71.375	72.438	-	-
RMS (m/s)						
Pred. hor.						
1 sec.	1.271	0.716	0.324	0.312	0.733	0.734
10 sec.	1.271	1.394	0.427	0.387	1.706	1.730
20 sec.	1.271	1.318	0.428	0.401	1.934	1.953
∞	1.271	1.267	0.428	0.412	-	-

Table 3.6: Downwind EWS predictability in the experiment (B) using LTI models and upwind turbine EWS and P_{ref} data.

3.2.2 Online LTI model and prediction

The presented algorithm in section 3.1.2 for SISO model could be easily extended to multi input single output (MISO) model with online parameter estimation. It is assumed that MISO model has following form:

$$\begin{aligned} y(t) + a_1 y(t-1) + ... + a_{n_a} y(t-n_a) = b_1 u_1(t-n_k) + ... + b_{n_b} u_1(t-n_b-n_k) + ... \\ d_1 u_2(t-n_k) + ... + d_{n_d} u_2(t-n_d-n_k) + e(t) \end{aligned} \tag{3.12}$$

The delay is considered on the inputs. The coefficient and previous values of new input then has to be added to parameter vector and regression vector respectively as below:

$$\begin{aligned} &\psi^T(t) = [-y(t-1)... - y(t-n_a) \;\; u_1(t-n_k)...u_1(t-n_b-n_k) \;\; u_2(t-n_k)...u_2(t-n_d-n_k)] \\ &\theta_0(t) = [a_1...a_{n_a} \;\; b_1...b_{n_b} \;\; d_1...d_{n_d}]^T \end{aligned} \tag{3.13}$$

Tables 3.7 and 3.8 show the results of using the recursive algorithm for experiment (A) and (B) respectively with inclusion of P_{ref}. The model which updated recursively is of the form (3.12) with $n_a = n_b = n_d = 3$. The results are again similar to the offline ArxDel model shown in table 3.5 and 3.6.

Pred. hor.	1 sec.	20 sec.	40 sec.
Fit (%)	94.761	89.774	89.593
RMS (m/s)	0.078	0.151	0.154

Table 3.7: Downwind EWS predictability in the experiment (A) using online LTI models and upwind turbine EWS and P_{ref} data.

Pred. hor.	1 sec.	10 sec.	20 sec.
Fit (%)	78.151	71.504	71.030
RMS (m/s)	0.324	0.422	0.429

Table 3.8: Downwind EWS predictability in the experiment (B) using online LTI models and upwind turbine EWS and P_{ref} data.

Chapter 4

Decentralized Wind Flow Model and Prediction

This chapter consists of two parts. First, a fusion algorithm is suggested for predictions of wind speed at local turbines in which the set of turbines are considered to include at least three turbines, i.e., two upwind turbines and one downwind turbine. A structure is then proposed for the decentralized model for the Aeolus benchmark where the local models are indeed subsystems of the dynamic decentralized flow model in this wind farm.

4.1 Fusion of on-line predictions

4.1.1 Effect of wind direction on performance of model identification

In order to address the direction issue and its effect on identification performance, the measurement data from OWEZ wind farm are used. The OWEZ wind farm seen in figure 4.1 consists of large modern wind turbines. Measurements are available from a set of six turbines surrounded by the red box.

Variation of the wind direction is shown in figure 4.2 . For the first 4-hour data, the direction of wind is approximately 138° from north geographic, which is aligned with wind turbines row consisting WTG04, WTG03, WTG02 or WTG16, WTG15, WTG14. The ArxDel model is identified for this segment of data in (Knudsen and Soltani, 2009). The effective wind speed of WTG03 is used as input and that of WTG02 as output. This model is evaluated here for 20 hours data and percentage of fitness is calculated for each 4-hour segment using both offline and recursive algorithms. Results are shown in figure 4.3. It can be concluded that when wind direction degrades from 138°, the fitness percentage decreases in both online and offline models, however, online/recursive model shows slightly higher fitting.

This analysis shows that, also in reality, the models are highly affected by the wind direction

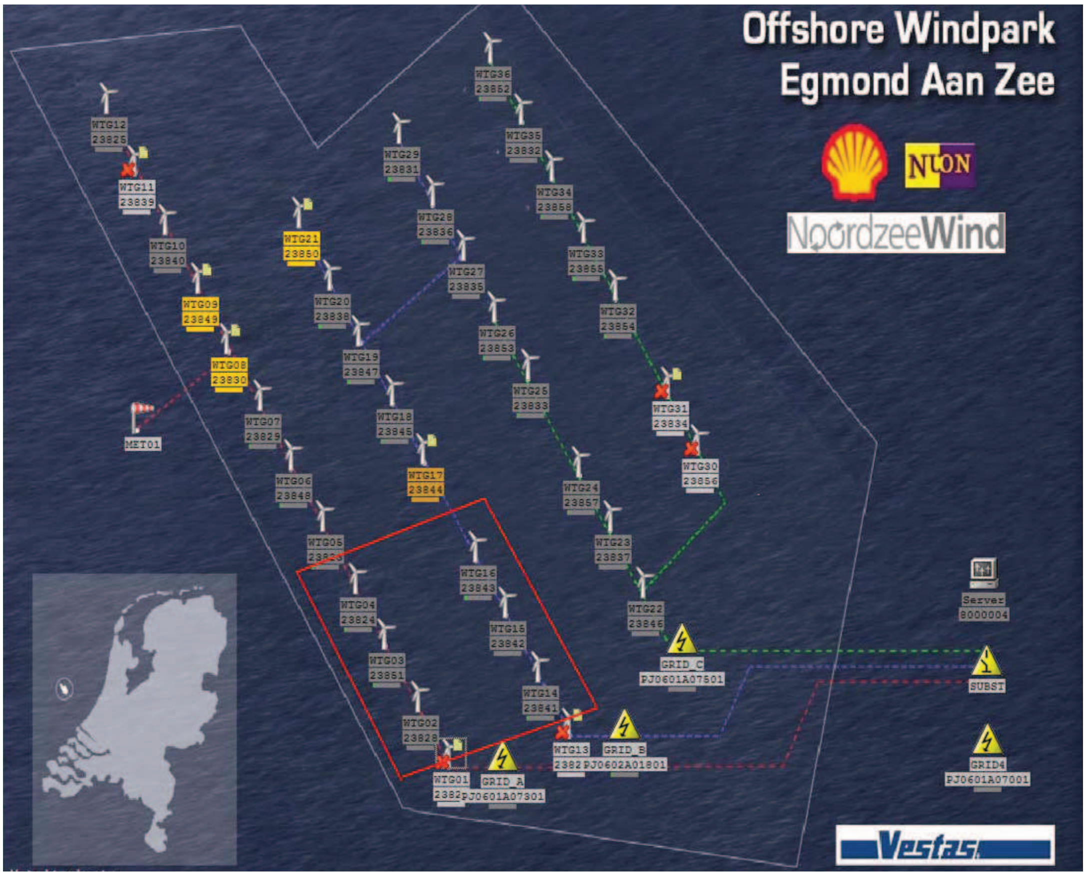

Figure 4.1: OWEZ wind farm layout.

and therefore we should choose a strategy to improve the predictions under these variations.

Now consider that generally there are two upwind neighboring turbines for any turbine in the wind farm (except the front row with respect to wind direction) at a time. A logical algorithm can be easily implemented on each turbine to find these two upwind turbines at a time regarding the wind farm layout, wind turbine location, and wind direction.

The wind direction then is either such that the wind blows from between two upwind turbines toward the downwind turbine, or it is exactly aligned with one upwind-downwind turbine connecting line. The fusion algorithm should subsequently be used to improve the predictions in the former situation while it gives the same results as using the models in section 3.2 in the latter situation.

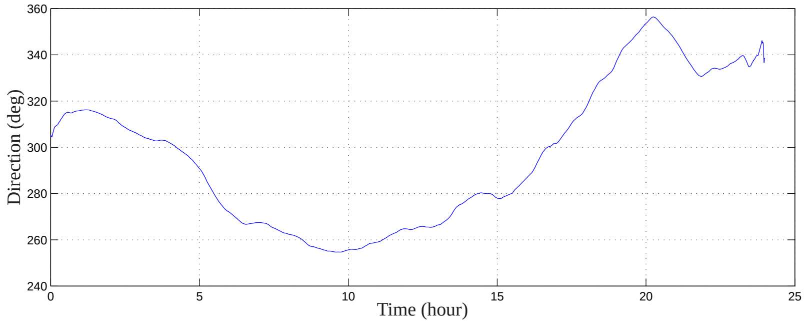

Figure 4.2: Wind direction at OWEZ wind farm.

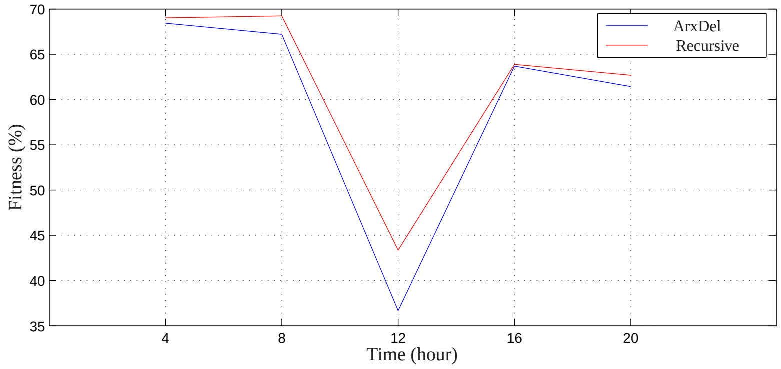

Figure 4.3: Comparison of fitness percentage of the delayed ARX model and recursive algorithm with prediction horizon of 60sec.

4.1.2 Fusion of Predictions

In a single sensor system one sensor is selected to monitor the system or its surrounding environment. A multi-sensor system employs several sensors to obtain information in a real world environment full of uncertainty and change. This means various types of sensors and different sensor technologies are employed, where some of these sensors have overlapping measurement domains. Multiple sensors provide more information and hence a better and more precise understanding of a system. Moreover, a single sensor is not capable of obtaining all the required information reliably at all times in varying environments.

In order for the advantages of multi-sensor systems to be realized, it is essential that the information provided by the sensors is interpreted and combined in such a way that a reliable, complete and coherent description of the system is obtained. This is the data fusion problem. Multi-sensor fusion is the process by which information from many sensors is combined to yield an improved description of the observed system (Mutambara, 1998).

In here each upwind turbine affect the downwind wind field in wind farm. Thus, it is necessary to use the information of upwind neighbor turbines to form a dynamic model. In addition the wind direction has an important effect. As it is shown in previous section, the effect degree of each upwind turbine on the behavior of a downwind turbine, is heavily dependent on the wind direction. Therefore, wind direction and information of upwind neighbor turbines must be combined in order to have the more accurate prediction for the downwind turbine. If each wind turbine is assumed as a sensor, the problem of predicting the EWS by using information of upwind neighbors, is translated to a multi-sensor fusion problem.

As it is stated, there is a dynamic model for each upwind neighboring turbine and considered downwind turbine. These models are updated recursively and their outputs are the local predictions of EWS at considered wind turbine. The problem is how to fuse these local predictions and eventually obtain the global prediction of EWS at the downwind wind turbine. Each model gives the predicted EWS at the downwind wind turbine with its prediction error covariance matrix. A straight-forward fusion algorithm is simply to take the variance weighted average of predicted EWSs and obtain the global prediction y_g as follows:

$$\begin{aligned} y_g(k+1|k) &= P_T(k+1|k)\sum_{i=1}^{N} P_i^{-1}(k+1|k)\hat{y}_i(k+1|k) \\ P_T(k+1|k) &= [\sum_{i=1}^{N} P_i^{-1}(k+1|k)]^{-1} \end{aligned} \tag{4.1}$$

where $\hat{y}_i$ is the i^{th} predicted EWS and $P_i(k+1|k)$ is the corresponding error covariance. The weightings are dependent only to error covariances. To make the weightings more flexible, fusing is simply done by using recursive modeling. The main advantages of this approach is simplicity. Moreover, the effect of wind direction is inherently included in the weightings. In this regard, the local EWS predictions are assumed as inputs of a dynamical model with an output that is global EWS prediction at the downwind wind turbine. The fusion algorithm has the from

$$y_g(k+1|k) + a_1 y_g(k|k-1) = \sum_{i=1}^{N} b_i\hat{y}_i(k+1|k) + e(t) \tag{4.2}$$

where each local prediction $\hat{y}_i$ is assumed as input and global prediction y_g as output. Parameters a_1 and b_i are computed by using the same recursive algorithm as before.

Above fusion algorithm is evaluated for two cases as shown in figure 4.4 . In the experiment (C), wind direction is parallel to the connecting line between WT_1 and WT_2 turbines. The aim is to predict EWS at WT_2 by fusing the predicted output of online LTI models from WT_1 and WT_3 to WT_2. Finally in the experiment (D), the wind direction is 15 degrees more than the experiment (C).

Table 4.1 represents the results of experiment (C). It shows that in this case the predicted EWS by using WT_3 data can not improve the prediction since the downwind turbine WT_2 experienced less of the wind which experienced by WT_3. The results of experiment (D) are shown in table 4.2. It can be seen that fusing the local EWS predictions increases the fitness by about

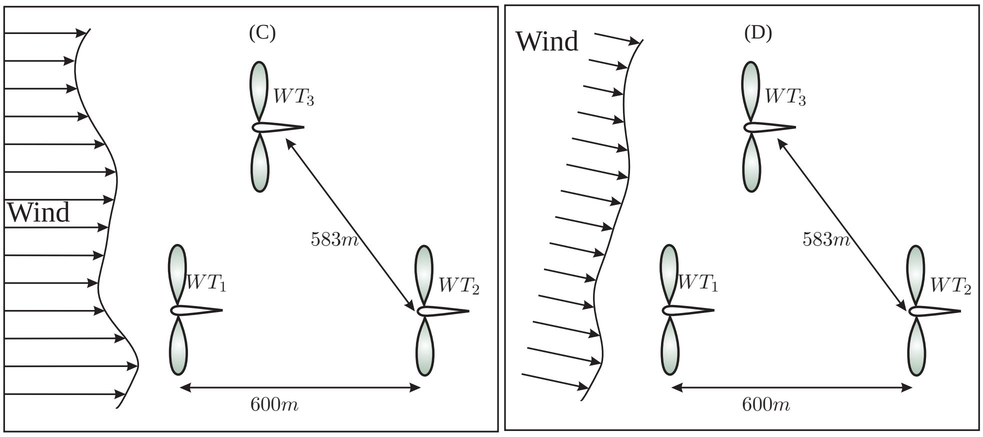

Figure 4.4: Layout of the experiments: (C) parallel to the connecting line between WT_1 and WT_2 turbines and (D) with 15° deviation from former situation in (C).

1%. It implies that improvement in prediction is dependent to the wind direction. Therefore, in a case that the upwind turbine is located exactly against the wind direction, the predictions from the other upwind turbines are not useful.

Pred. hor.	1 sec.	10 sec.	20 sec.
Fitness of Model from WT_1 to WT_2 (%)	94.761	90.010	89.774
Fitness of Model from WT_3 to WT_2 (%)	78.151	71.504	71.030
Fitness of fused prediction (%)	94.732	89.988	89.730
Error RMS of Model from WT_1 to WT_2 (m/s)	0.078	0.148	0.151
Error RMS of Model from WT_3 to WT_2 (m/s)	0.324	0.422	0.429
Error RMS of fused prediction (m/s)	0.078	0.148	0.152

Table 4.1: Results of fusion algorithm in the experiment (C)

Pred. hor.	1 sec.	10 sec.	20 sec.
Fitness of Model from WT_1 to WT_2 (%)	85.915	84.222	84.130
Fitness of Model from WT_3 to WT_2 (%)	79.617	76.599	76.453
Fitness of fused prediction (%)	87.310	85.289	85.165
Error RMS of Model from WT_1 to WT_2 (m/s)	0.202	0.226	0.227
Error RMS of Model from WT_3 to WT_2 (m/s)	0.292	0.335	0.337
Error RMS of fused prediction (m/s)	0.181	0.211	0.213

Table 4.2: Results of fusion algorithm in the experiment (D)

4.2 Aeolus Benchmark Layout

The basic farm configuration of Aeolus benchmark (Soltani et al., 2010) uses 10 NREL 5MW (Jonkman et al., 2009). wind turbines placed in a lattice structure as shown in figure 4.5. The inter-turbine distances are 600 meters in the horizontal and 500 meters in vertical directions.

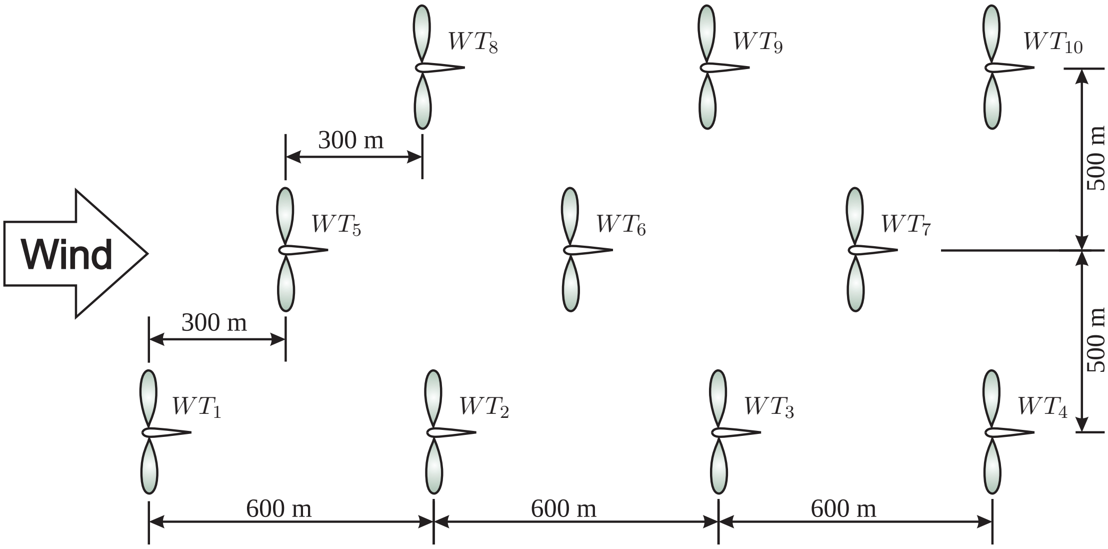

Figure 4.5: The configuration of the wind farm. The mean wind direction is marked with an arrow.

The mean wind speed is chosen at $15\frac{m}{s}$ which ensures that the turbines are operating in full load. Again this is a deliberate choice to simplify the closed form control modeling. Other scenarios could be chosen for future versions of the model e.g. a scenario where the wind speed is $11.4\frac{m}{s}$ which is the rated wind speed.

The wind turbine model is a simplified aero-elastic model based on static CP/CT tables, a simple 3rd order drive train model and a 1st order generator model. The turbines are controlled using the control strategy from (Jonkman et al., 2009) which includes a simplified start up procedure and pitch control for full load operation. The most important turbine parameters are listed in

Table 4.3: Gross properties for NREL 5-MW turbine

Rated power	$5MW$
Configuration	Upwind, 3 Blades
Control	Variable speed, collective pitch
Drive train	High speed, Multi-stage gearbox
Rotor, Hub diameter	$126m$, $3m$
Hub height	$90m$
Cut-in, Rated, Cut-Out Wind speed	$3\frac{m}{s}$, $11.4\frac{m}{s}$, $25\frac{m}{s}$
Cut-in, Rated rotor speed	$6.9rpm$, $12.1rpm$

Table 4.4: Wind field parameters

Mean wind speed	$15\frac{m}{s}$
Turbulence intensity	10%
Length of wind field (x)	2000 m
Length of wind field (y)	1800 m
Accuracy of the spatial grid	$15m$
Sample time	$1s$
Simulation lenght	5000 s

the table 4.3.

The wind field model characteristics are provided in the table 4.4. These characteristics are used to generate data for simulation. The model which will be used for control design can then use the generated data for evaluation.

Decentralized flow model is obtained for the benchmark in two cases. In the first case, the wind direction is aligned with the rows of wind turbines. So, EWS and P_{ref} of a wind turbine at a row does not contribute to EWS at another row here. Therefore, the decentralized model is a combination of local models from each front turbine to one turbine downwind. Figure 4.6 shows the decentralized wind flow model for this case where M_j^i represents a model from i^{th} wind turbine EWS to j^{th} wind turbine EWS.

The wind direction in the second case is 15 degrees more than that of the first case. Thus, the model is more complex compared to the first case, i.e., two upwind turbines will contribute to the EWS prediction at each downwind turbine. Figure 4.7 shows the decentralized wind flow model for this case where $M_j^{i,k}$ represents two models from i^{th} and k^{th} wind turbine's EWS and P_{ref} to j^{th} wind turbine's EWS.

Both off-line and on-line models for two cases have been saved in a Matlab package.

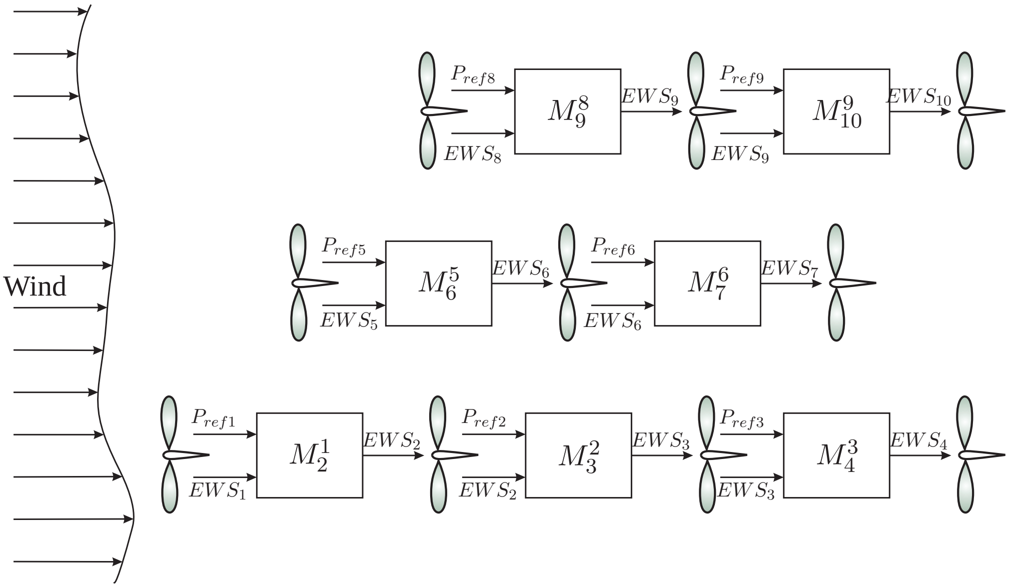

Figure 4.6: Decentralized wind flow model when the wind direction is parallel to the wind turbine rows.

The fusion algorithm is applied to the first row of the benchmark wind farm in two cases, which are displayed in figure 4.6 and 4.7. In each case, the EWS at the downwind wind turbines is predicted. The results are presented in figure 4.8 and 4.9. In each figure, plot "a" (blue line) shows the fitness of the predicted EWS at the downwind wind turbine by using only the front upwind wind turbine information (e.g. information of WT_1 for predicting the EWS at WT_2 location), plot "b" (green line) reveals the same by using only the lateral upwind wind turbine information (e.g. information of WT_5 for predicting the EWS at WT_2 location) and finally in the plot "c" (red line) both upwind wind turbines' informations are fused.

Figure 4.8 shows that the information of a lateral upwind turbine does not improve the accuracy

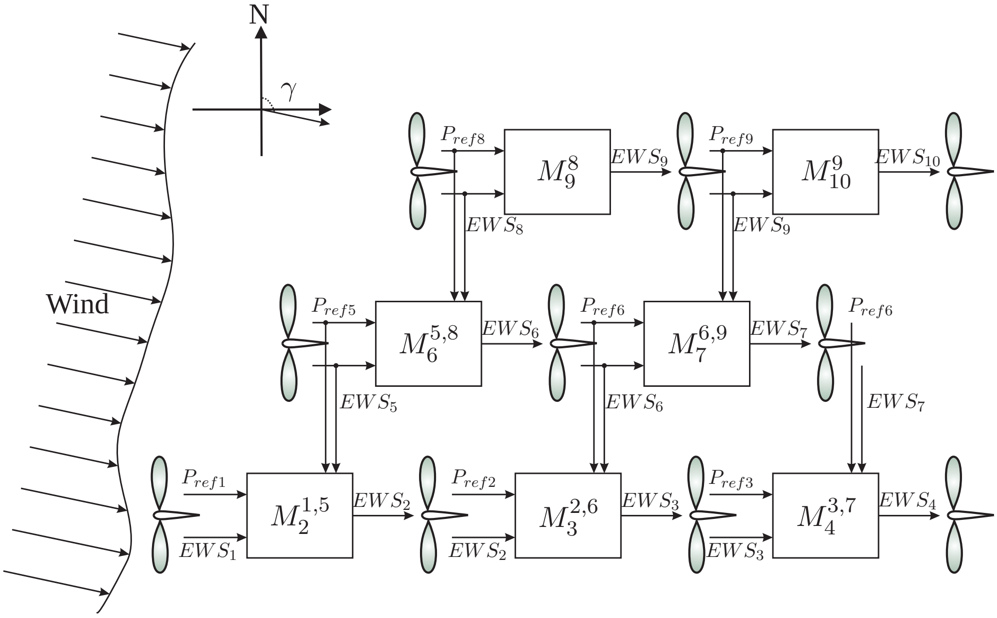

Figure 4.7: Decentralized wind flow model in the benchmark wind farm with different wind direction.

of a fused prediction at the downwind wind turbine, when the wind direction is parallel to the wind turbine rows. On the other hand, figure 4.9 indicates improvements in fused prediction (see plot "c"), when the wind has an angle with respect to the wind turbine rows, i.e., the downwind wind turbine experienced the more wake of the lateral upwind wind turbine.

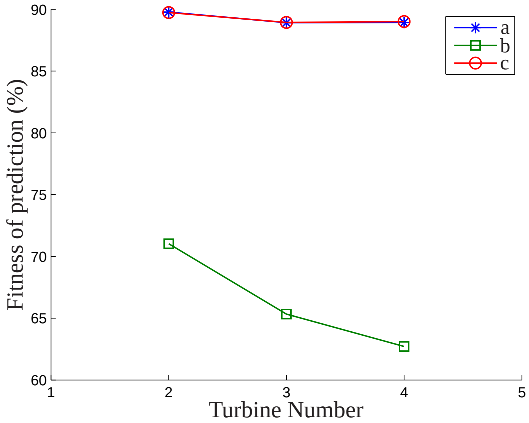

Figure 4.8: Fitness of the predicted EWS at downwind wind turbines in the benchmark wind farm when the wind direction is parallel to the wind turbine rows.

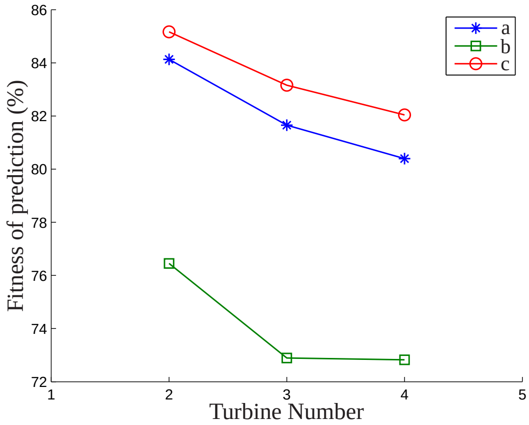

Figure 4.9: Fitness of the predicted EWS at downwind wind turbines in the benchmark wind farm with different wind direction.

Chapter 5

Discussion and Conclusion

This report is part of the deliverable D2.3 of the European research FP7 project, Aeolus. In this report a *Decentralized Dynamic Wind Flow Model* for wind farms is investigated. The presented model uses the estimated effective wind speed of the neighboring turbines in a wind farm to predict the wind speed at each wind turbine. This means that the calculation load will be distributed on all turbines locally. The results are especially useful when designing a decentralized wind farm controllers in a large-scale wind farm.

The presented models are also obtained both in an off-line and on-line manner. The off-line model is a valid model for a specific wind condition (average wind speed/direction) where the parameters of the model are obtained from recorded data. It is also less useful when there are uncertainties in the real experiment (for example, temperature and seasonal changes which are not considered in the wind farm model). The on-line model instead updates the parameter at each time step and therefore adapts the model to the new condition in order to obtain less prediction error from the model.

It is also observed that the model from a neighboring upwind turbine to downwind turbine is highly affected by the wind direction. The on-line prediction will be affected with the change in wind direction but still it can be improved when a number of predictions are available from upwind turbines. It is rather easy to implement an algorithm to find the neighboring upwind turbines for each wind turbine when the layout of the wind farm is available. The fusion algorithm, presented in this report then provides a fusion on the wind predictions at downwind using the information from upwind neighboring turbines. The results of the fusion shows an improvement on the prediction especially when the wind direction is not parallel to the wind turbine rows.

Bibliography

J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system development. Technical report, National Renewable Energy Laboratory, 2009.

T. Knudsen and M. Soltani. Preliminary dynamic model - aeolus - wp2. Technical report, Aalborg University, 2009. Deliverable no.: 2.2.

W. Leithead. Effective wind speed models for simple wind turbine simulations. In *Wind energy conversion 1992: proceedings of the 14th British Wind Energy Association Conference, Nottingham, 25-27 March 1992*, page 321. Amer Society of Mechanical, 1992.

L. Ljung. *System Identification Theory for the User*. Prentice Hall, second edition, 1999.

L. Ljung and S. Gunnarsson. Adaptation and tracking in system identification–a survey. *Automatica*, 26(1):7 – 21, 1990. ISSN 0005-1098. doi: DOI:10.1016/0005-1098(90)90154-A.

A. G. O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, 1998.

M. Soltani, T. Knudsen, and T. Bak. Simulation models for case studies. Confidential, AAU, 2009. Deliverable no.: 5.4.

M. Soltani, J. Grunnet, T. Knudsen, and T. Bak. Aeolus toolbox for dynamics wind farm model, simulation and control. In *European Wind Energy Conference and Exhibition (EWEC) 2010*, Warsaw, Poland, Tuesday 20 - Friday 23 April 2010, 2010. European Wind Energy Association (EWEA). Submitted.

K. Z. Østergaard, P. Brath, and J. Stoustrup. Estimation of effective wind speed. *Journal of Physics: Conference Series*, 75(1):012082, 2007.