



Wind Velocity Estimation for Wind Farms

Eduardo Bezerra Rufino Ferreira Paiva

► To cite this version:

Eduardo Bezerra Rufino Ferreira Paiva. Wind Velocity Estimation for Wind Farms. Automatic. Université Paris sciences et lettres, 2023. English. NNT : 2023UPSLM046 . tel-04496158

HAL Id: tel-04496158

<https://pastel.hal.science/tel-04496158>

Submitted on 8 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à Mines Paris – PSL

Wind Velocity Estimation for Wind Farms
Estimation de la Vitesse du Vent dans un Parc Éolien

Soutenue par

Eduardo Bezerra
Rufino Ferreira Paiva

Le 20 décembre 2023

École doctorale n°621

Ingénierie des Systèmes,
Matériaux, Mécanique,
Énergétique

Spécialité

Mathématique et Au-
tomatique

Composition du jury :

Catherine BONNET Directrice de Recherche Inria, L2S- CentraleSupélec	<i>Présidente du jury</i>
Gildas BESANÇON Professeur, Grenoble - INP (GIPSA-lab)	<i>Rapporteur</i>
Franck PLESTAN Professeur, LS2N, École Centrale de Nantes	<i>Rapporteur</i>
Delphine BRESCH-PIETRI Maître assistante, Mines Paris – PSL	<i>Examinatrice</i>
Olivier LEPREUX Ingénieur contrôle, IFP Energies nouvelles	<i>Examineur</i>
Nicolas PETIT Professeur, Mines Paris – PSL	<i>Directeur de thèse</i>

Résumé

Cette thèse propose des algorithmes pour estimer la vitesse et la direction du vent pour des éoliennes et des parcs éoliens.

Tout d'abord, nous proposons des méthodes basées sur les données pour estimer la vitesse effective du rotor (REWS) sans nécessiter la connaissance de certains paramètres physiques de l'éolienne, qui pourraient être inconnus de l'opérateur. Nous fournissons deux méthodes basées sur les données, l'une basée sur la régression par processus gaussien et l'autre combinant la régression par processus gaussien avec un observateur grand gain.

Ensuite, en nous basant sur cette estimation locale de la REWS, au niveau d'une éolienne, nous abordons la question de l'estimation du vent en écoulement libre au niveau du parc éolien.

Nous commençons par nous concentrer sur l'estimation de la vitesse du vent, pour une direction du vent connue. Pour un parc éolien de géométrie simple, nous démontrons qu'une mesure locale de la vitesse perturbée par la présence des éoliennes peut être utilisée pour estimer la vitesse du vent en écoulement libre. Nous fondons notre méthodologie d'estimation sur une modélisation simplifiée de l'effet de sillage qui consiste en des équations aux dérivées partielles hyperboliques du premier ordre en cascade, et dont la vitesse de transport est la vitesse du vent en écoulement libre. Nous proposons d'utiliser une solution analytique de ces équations, impliquant des retards de transport, pour effectuer une estimation de la mesure locale et mettre à jour l'estimation de la vitesse du vent en écoulement libre. Nous démontrons formellement la convergence de cette estimation et illustrons numériquement l'efficacité de cette méthode.

Enfin, nous passons à une configuration plus générale où à la fois la vitesse et la direction du vent en écoulement libre sont inconnues. Nous proposons d'utiliser une modélisation bidimensionnelle du sillage et de nous appuyer sur une méthode basée sur l'optimisation. Le problème d'identification que nous formulons se révèle être particulièrement difficile en raison de l'apparition de retards de transport, mais nous montrons comment contourner cette difficulté en considérant une valeur moyenne de l'historique de la vitesse du vent en écoulement libre. Des résultats de simulation obtenus avec le simulateur FAST.Farm illustrent l'intérêt de la méthode proposée.

Mots-clés : estimation de la vitesse du vent, estimation de parc éolien,

énergie éolienne, régression par processus gaussien, observateur grand-gain, retards de transport, équations aux dérivées partielles hyperboliques du premier ordre.

Abstract

This thesis designs algorithms to estimate the wind speed and direction for wind turbines and wind farms.

First, we propose data-based methods to estimate the Rotor Effective Wind Speed (REWS) for a single turbine without prior knowledge of certain physical parameters of the turbine that might be unknown to an operator. We provide two data-based methods, based respectively on Gaussian Process Regression (GPR) and on a combination of GPR with high-gain observers.

Second, grounding on this REWS estimation at the local level of one turbine, we address the question of estimating the free-flow wind at the level of a wind farm.

We start by focusing on wind speed estimation, for a given known wind direction. For a wind farm with a simple geometry, we prove that a local speed measurement disturbed by the presence of the turbines can be used to estimate the free-flow wind speed. We ground our estimation methodology on a simplified wake model, which consists of first-order hyperbolic partial differential equations, the transport speed of which is the free-flow wind speed. We propose to use an analytical solution of these equations, involving transport delays, to perform an estimate of the local measurement and to update the free-flow wind speed estimate. We formally prove the convergence of this estimate and numerically illustrate the efficiency of this method.

Finally, we move to a more general setup where both the free-flow wind speed and direction are unknown. We propose to use a two-dimensional wake model and to rely on an optimization-based method. This identification problem reveals to be particularly challenging due to the appearance of transport delays, but we illustrate how to circumvent this issue by considering an average value of the free flow wind speed history. Simulation results obtained with the simulator FAST.Farm illustrate the interest of the proposed method.

Keywords: wind speed estimation, wind farm estimation, wind energy, Gaussian process regression, high-gain observer, transport delays, first-order hyperbolic partial differential equations.

List of Figures

1.1	Croissance annuelle de la capacité installée mondiale d'énergie éolienne. Données de l'IRENA [57].	14
1.2	Schéma d'une éolienne.	15
1.3	Courbe de puissance pour une éolienne Senvion MM-82. La vitesse de démarrage est de 4 m/s, la vitesse nominale est de 14,5 m/s, la vitesse de coupure est de 22 m/s et la puissance nominale est de 2 050 kW. Les données de la figure et les valeurs de référence proviennent de [127].	16
1.4	Parc éolien offshore de Horns Rev 1 au Danemark. Photographie de Christian Steiness / Vattenfall.	21
1.5	Schéma du sillage d'une éolienne. Figure adaptée de [107, Fig. 2]	22
1.6	Le modèle de sillage proposé dans [61]. Figure adaptée de [61, Fig. 5].	24
2.1	Yearly growth of the installed wind energy capacity from 2011 to 2021 (one data point per year). Data from IRENA [57]. . .	30
2.2	Schematic of a wind turbine.	31
2.3	Power curve for a Senvion MM-82. The cut-in speed is 4 m/s, the rated speed is 14.5 m/s, the cut-off speed is 22 m/s, and the rated power is 2,050 kW. The data for the plot and the reference values were taken from [127].	32
2.4	Offshore wind farm Horns Rev 1 in Denmark. Photograph by Christian Steiness / Vattenfall.	36
2.5	Schematic of the wake of a wind turbine. Adapted from [107, Fig. 2].	37
2.6	The wake model proposed in [61]. Adapted from [61, Fig. 5].	39
3.1	Schematic of the algorithm for one cluster.	52
3.2	Rotor speed time-series corresponding to the selected day. The points are color-coded according to the cluster used for the estimation of the wind speed at the corresponding time. .	54

3.3	Pitch angle time-series corresponding to the selected day. The points are color-coded according to the cluster used for the estimation of the wind speed at the corresponding time. . . .	54
3.4	Produced power time-series corresponding to the selected day. The points are color-coded according to the cluster used for the estimation of the wind speed at the corresponding time. .	55
3.5	Estimation for consecutive points to simulate a real-time application. The parameters are $n_c = 5$, $r = 20$, $N = 10^4$, and $M = 3 \times 10^3$. We use only the nearest cluster estimate. . . .	56
3.6	Boxplots of the estimation error for the day shown in Figs. 3.5a and 3.5b.	57
3.7	Relative error for different choices of the number of clusters. In this case $r = 20$, $N = 10^4$, $M = 3000$, and we only use the nearest cluster estimate.	57
3.8	Relative error for different values of M . The remaining parameters are $n_c = 5$, $r = 20$, and $N = 10^4$. We use only the nearest cluster estimate.	58
3.9	Relative error for different values of N . The remaining parameters are $n_c = 5$, $r = 20$, and $M = 3000$. We use only the nearest cluster estimate.	58
3.10	The C_q map for the turbine used in this study.	66
3.11	The term $C_q v^2$ as a function of v and ω with the pitch angle fixed to zero. Recall that $T_a \propto C_q v^2$	66
3.12	Comparison in performance among different values of λ for the same reference. The reference data is the same as in Fig. 3.5a.	69
3.13	Relative error for different values of λ corresponding to estimation results shown in Fig. 3.12. The crosses represent the mean values, the red lines are the medians, the boxes are in between the 25th and 75th percentiles and the whiskers are in between the 5th and 95th percentiles.	69
3.14	Estimation with $\lambda = 10$ for a different reference.	70
4.1	Array of wind turbines and downwind measurement device. In this case, we have an array of two wind turbines and the measurement is given by a third turbine (equipped with a REWS estimation algorithm, for instance).	74
4.2	Maps of a portion of the Ablaincourt wind farm (left-hand side) [86, p. 30] and the Horns Rev wind farms (right-hand side) [123, Fig. 2].	75
4.3	Steady-state behavior of the estimation error $U_\infty(t) - \hat{U}_\infty(t)$ for $U_\infty(t) = 10 + \sin(2\pi ft)$ for different frequencies f and different gains k . The initial guess is $\hat{U}_\infty(0) = 15$ m/s in these cases.	91

4.4	Time-evolution of the estimation error $U_\infty(t) - \hat{U}_\infty(t)$ for $U_\infty(t) = 10 + \sin(2\pi ft)$ for different frequencies f and different gains k . The initial guess is $\hat{U}_\infty(0) = 15$ m/s in these cases.	91
4.5	Free-flow wind speed estimation for the case $N = 1$ with no modeling error and no measurement noise for an experimental free-flow condition. We used $k = 10$ and $\hat{U}_\infty(0) = 10$ m/s. . .	92
4.6	Region of guaranteed convergence for the choice of parameters used in Subsection 4.6.1.	93
4.7	Free-flow wind speed estimation results using the local measurement of the second ($N = 1$ case) and, respectively, third ($N = 2$ case) turbine for a window of 10 minutes. The reference for the free-flow wind speed comes from experimental data. The gain is $k = 10$ and $\hat{U}_\infty(0) = 10$ m/s.	94
4.8	Free-flow wind speed estimation error for the cases presented in Fig. 4.7.	94
4.9	Estimation of the local speed measurement in the $N = 2$ case corresponding to Fig. 4.7.	95
4.10	Speed deficit in the $N = 2$ case corresponding to Fig. 4.7. . .	95
4.11	Free-flow wind speed estimation results using noisy local measurements of the third turbine ($N = 2$) for different gains. We used $\hat{U}_\infty(0) = 10$ m/s.	96
5.1	Top-view of a wind farm with $N = 2$ turbines. The system of coordinates (X, Y) is the reference frame, (x, y) is the wind frame, the yaw angles γ_1 and γ_2 and wind direction γ_∞ are measured with respect to the direction X . We also show the x - and y -distances between the turbines, which are important to quantify how much the second turbine is affected by the wake of the first.	103
5.2	Simulation of Shapiro's 2-D model. The wake centerline is shown in red and the turbine is shown in black. The speed at each point in space is represented in meters per second. The arrows show the wind direction.	105
5.3	Case with sinusoidal wind speed with the estimation computed for time $t = 100$ s so that $t - \ell\Delta t = 0$. The negative time values correspond to a portion of the time window $[t - 2\bar{\tau} - \ell\Delta t, t - \ell\Delta t] = [-2\bar{\tau}, 0]$, where (5.57) holds, that is, the speed estimate is constant. We show a few seconds of this interval here to illustrate this point.	119

5.4	Case with experimental wind speed. The estimation is computed at time $t = 100$ s so that $t - \ell\Delta t = 0$. The negative time values correspond to a portion of the time window $[t - 2\bar{\tau} - \ell\Delta t, t - \ell\Delta t] = [-2\bar{\tau}, 0]$, where (5.57) holds, that is, the speed estimate is constant. The initial guess for the wind is constant equal to 12 m/s.	120
5.5	Case with yaw angle 10 degrees and y -distance 0.	121
5.6	Case with yaw angle 0 degree and y -distance 0.5 D	122
5.7	Case with yaw angle 10 degrees and y -distance 0.5 D	123

Contents

1	Introduction (en français)	13
1.1	Motivation	13
1.2	Avant-propos sur les éoliennes	14
1.3	Contrôle des parcs éoliens et estimation du vent	18
1.3.1	Contrôle local et REWS	19
1.3.2	Contrôle centralisé avec estimation du sillage	20
1.3.3	Modélisation du sillage	22
1.4	Contributions et organisation de la thèse	27
1.5	Publications	28
2	Introduction (in English)	29
2.1	Motivation	29
2.2	Preliminaries on Wind Turbines	30
2.3	Wind farm control and wind estimation	34
2.3.1	Local control and REWS	35
2.3.2	Centralized control with wake estimation	35
2.3.3	Wake modeling	37
2.4	Thesis contributions and outline	41
2.5	Publications	43
3	Rotor effective wind speed estimation	45
3.1	Introduction	45
3.2	Estimating the REWS with a machine learning approach	46
3.2.1	Gaussian process regression	47
3.2.2	Clustering	49
3.2.3	Finding an estimate per cluster	49
3.2.4	Finding the final estimate	51
3.3	Numerical results for the REWS estimation with GPR	53
3.3.1	Performance measurement	53
3.3.2	Results	53
3.4	Estimating the wind speed with a mixed data/model-based approach	59
3.4.1	Estimating the aerodynamic torque with GPR	59

3.4.2	High-gain observer to estimate the REWS	60
3.5	Numerical results for the REWS estimation with GPR and high-gain observer	68
3.6	Conclusion	70
4	Free-flow wind speed estimation in a wind farm from mea- surements inside the wake	73
4.1	Introduction	73
4.2	Shapiro's wake modeling and problem statement	75
4.3	Analytical expression of the speed deficit with transport delays	78
4.4	Proposed estimation method	81
4.4.1	Convergence result	81
4.4.2	Proof of Theorem 4.1	83
4.5	Free-flow speed estimation	88
4.5.1	Application of Theorem 4.1 to Shapiro's model	88
4.5.2	Estimation strategy implementation	89
4.6	Numerical results	90
4.6.1	Cases with no modeling error and no noise	90
4.6.2	Cases with modeling error and measurement noise	92
4.7	Conclusion	96
5	Free-flow wind speed and direction estimation in a wind farm from local wind speed measurements	99
5.1	Introduction	99
5.2	Shapiro's two-dimensional wake model	100
5.2.1	Wind farm modeling	101
5.2.2	Analytical solutions of the PDEs (5.1)–(5.3) with BCs (5.9)–(5.11)	105
5.3	Estimation through nonlinear least squares	106
5.3.1	Formulation of the estimation problem as a continuous- time optimization problem	107
5.4	Discrete-time formulation of the optimization problem	111
5.5	Implementation and numerical results	117
5.5.1	Numerical results with Shapiro's model as a reference	118
5.5.2	Numerical results with FAST.Farm as a reference model	119
5.6	Conclusion	124
6	Conclusions and perspectives	127
6.1	Conclusions	127
6.2	Perspectives for future research	128

Chapitre 1

Introduction (en français)

1.1 Motivation

À la fin de 2022, la capacité installée d'énergie éolienne a atteint le chiffre impressionnant de 906 GW à l'échelle mondiale, avec 77,6 GW supplémentaires ajoutés au cours de cette seule année [55], une augmentation incroyable par rapport aux 7,5 GW de capacité installée en 1997 [57].

La figure 1.1 montre la croissance de la capacité installée de 2011 à 2021. On y constate une croissance exponentielle globale de la capacité installée mondiale et l'on peut donc s'attendre à ce que l'énergie éolienne joue un rôle majeur dans le secteur énergétique dans un avenir proche.

Avec l'expansion des parcs éoliens et le nombre croissant d'éoliennes rapprochées, les interactions aérodynamiques entre les éoliennes sont devenues une préoccupation majeure. Ces interactions peuvent entraîner des effets de sillage, dans lesquels les éoliennes en aval subissent des vitesses de vent réduites et des turbulences plus élevées causées par le sillage des éoliennes en amont. Par conséquent, des pertes de production d'électricité et une diminution potentielle de la durée de vie des éoliennes sont observées, posant des défis opérationnels pour l'efficacité des parcs éoliens et la production globale d'énergie. Les pertes de production d'électricité varient généralement de 5% à 10% [114], mais des pertes allant jusqu'à 23% de production d'électricité ont été rapportées dans la littérature (voir [11] et les références qui y sont présentées).

L'impact négatif de ces interactions aérodynamiques est particulièrement préoccupant car il fait grimper le coût de la production d'énergie des parcs éoliens. Réduire ce coût est un objectif essentiel que les sources d'énergie renouvelables, y compris l'énergie éolienne, doivent atteindre si elles veulent remplacer complètement les combustibles fossiles [108]. Pour surmonter ces défis et optimiser le fonctionnement des parcs éoliens, la technologie de contrôle avancée joue un rôle crucial [26, 104]. En fait, en employant des algorithmes de contrôle sophistiqués, les exploitants de parcs éoliens peuvent

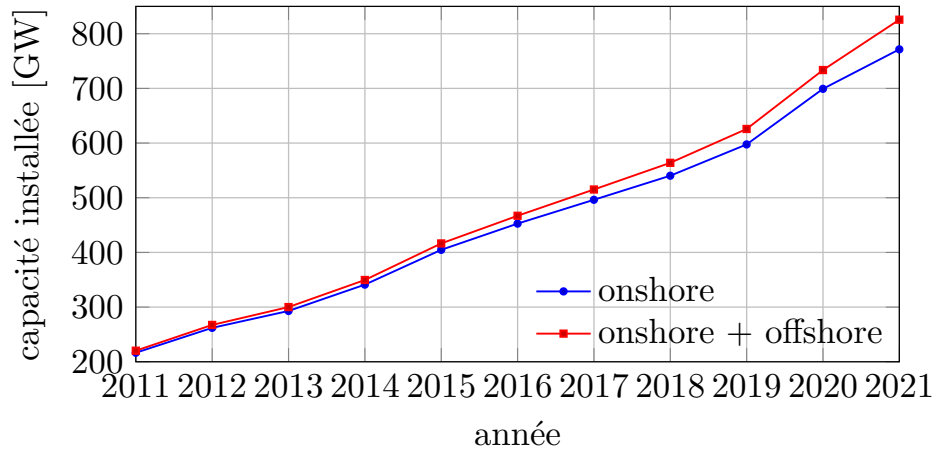


FIGURE 1.1 : Croissance annuelle de la capacité installée mondiale d'énergie éolienne. Données de l'IRENA [57].

gérer activement les paramètres des éoliennes, tels que l'alignement de lacet et les angles de calage, afin de minimiser les effets de sillage et de maximiser la capture d'énergie de la ressource éolienne disponible [72].

Ainsi, alors que l'énergie éolienne continue de jouer un rôle essentiel dans la transition énergétique mondiale, il est de la plus haute importance de relever les défis posés par l'augmentation de la densité des parcs éoliens. L'interaction entre la technologie de contrôle et les techniques d'estimation avancées est la clé pour atténuer les interactions aérodynamiques et optimiser les performances des parcs éoliens. En exploitant la puissance combinée du contrôle et de l'estimation, les centrales éoliennes peuvent exploiter efficacement l'abondante ressource éolienne et contribuer de manière significative aux efforts mondiaux de transition vers une énergie durable.

L'objectif de cette thèse est de contribuer à relever ces défis avec le développement d'algorithmes d'estimation pour les parcs éoliens.

1.2 Avant-propos sur les éoliennes

Ici, nous décrivons brièvement les éoliennes du type qui sera considéré dans cette thèse et présentons une modélisation mathématique de la dynamique du rotor.

Description et géométrie d'une éolienne

Une éolienne est un appareil qui convertit l'énergie cinétique du vent en énergie électrique. Nous en montrons un schéma sur la figure 1.2. L'éolienne représentée sur la figure 1.2 est une éolienne à 3 pales à axe horizontal qui est le type que nous considérons dans cette thèse et le plus courant. Le principe

Wind turbine

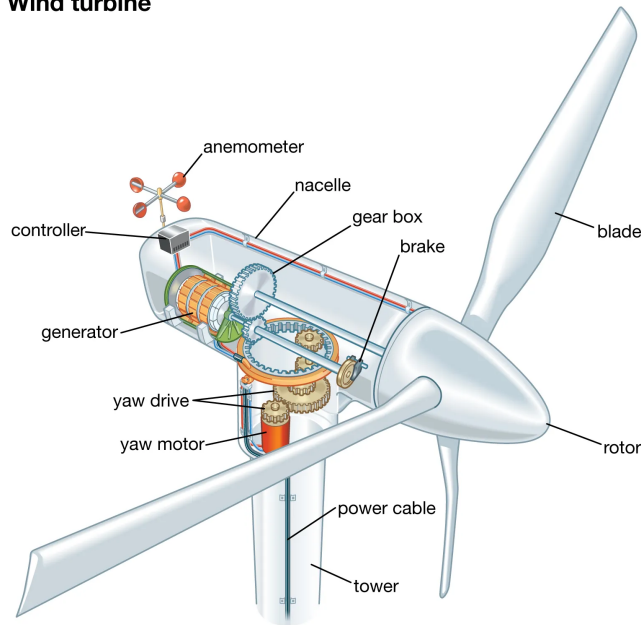


FIGURE 1.2 : Schéma d'une éolienne.

de base du fonctionnement est que le vent fait tourner le rotor et que cette énergie mécanique est transférée à un générateur électrique via une boîte de vitesses pour finalement produire de l'électricité ; pour des descriptions plus détaillées des composants et des différents types d'éoliennes à axe horizontal, reportez-vous, par exemple, à [8, 50]. Notez que d'autres types d'éoliennes existent ou ont été proposés, par exemple, les éoliennes aéroportées [28, 128], les éoliennes sans pales [126], les éoliennes jumelles [45, 106] et les éoliennes à axe vertical [9, 79], mais ils ne sont pas pris en compte ici.

En termes de contrôle, l'objectif est généralement de maximiser la puissance électrique produite, de minimiser la fatigue mécanique et/ou de se conformer aux exigences du réseau et les entrées de contrôle disponibles sont généralement les angles de calage des pales, l'angle de lacet de la nacelle (qui détermine la direction dans laquelle la turbine fait face), et le couple du générateur [3]. Dans le cas des éoliennes offshore flottantes, on souhaite également minimiser le mouvement de tangage de la plate-forme [134]. De plus, le fonctionnement d'une éolienne est généralement caractérisé par une courbe de puissance statique qui donne la puissance produite en fonction de la vitesse du vent et cette courbe est divisée en trois régions de fonctionnement [104] :

- dans la région 1, le vent est trop lent, inférieur à la vitesse dite de démarrage, pour que la puissance disponible compense les pertes dans

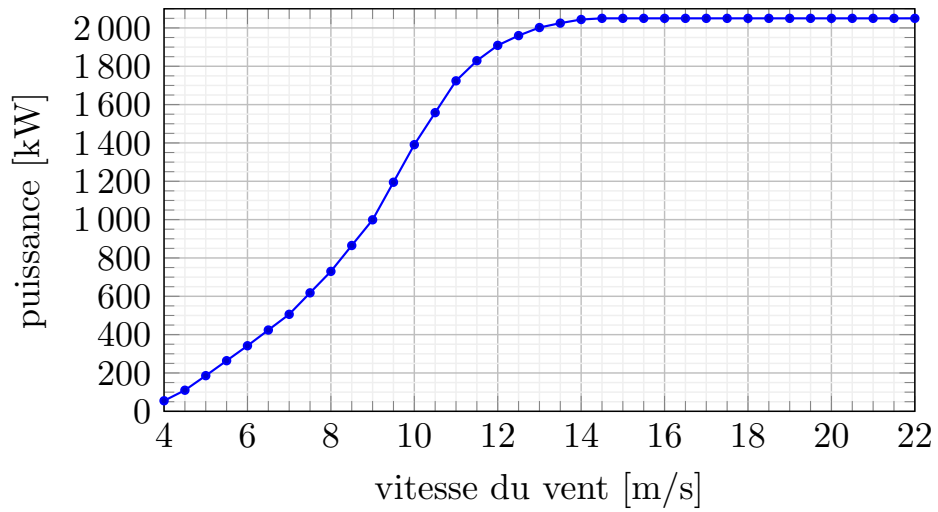


FIGURE 1.3 : Courbe de puissance pour une éolienne Senvion MM-82. La vitesse de démarrage est de 4 m/s, la vitesse nominale est de 14,5 m/s, la vitesse de coupure est de 22 m/s et la puissance nominale est de 2 050 kW. Les données de la figure et les valeurs de référence proviennent de [127].

les systèmes de turbine et la turbine ne fonctionne pas ;

- dans la région 2, le vent est suffisamment rapide et la turbine fonctionne, mais le vent n'est toujours pas assez rapide pour que la turbine génère sa puissance nominale, c'est-à-dire en dessous de la vitesse nominale ;
- dans la région 3, le vent est suffisamment rapide pour que l'éolienne génère sa puissance nominale.

Les objectifs de contrôle peuvent changer en fonction de la région d'exploitation : dans la région 2, normalement l'objectif est de maximiser la production d'énergie, les angles de calage sont maintenus fixes à leur valeur optimale à cet effet, et l'entrée de contrôle utilisée est le couple du générateur [1, 103, 112], alors que dans la région 3, l'objectif est généralement de minimiser les charges mécaniques sur la turbine tout en maintenant la production d'énergie au niveau nominal et des contrôleurs d'angle de calage sont utilisés [27, 30, 134]. On peut également définir une région 4, qui se situe au-dessus de la vitesse dite de coupure, où la turbine ne tourne pas. Par exemple, la figure 1.3 montre la courbe de puissance du modèle d'éolienne Senvion MM-82, qui est le modèle de l'éolienne dont nous utilisons les données expérimentales dans cette thèse.

Modélisation dynamique

A la suite, nous présenterons une modélisation de base de la dynamique du rotor qui est souvent invoquée dans la littérature sur le contrôle des éoliennes [87, 104].

Nous commençons par introduire l'expression de la puissance aérodynamique captée par le rotor, P_a ,

$$P_a = \frac{1}{2} \rho \pi R^2 C_p \left(\frac{\omega R}{U_\infty}, \beta \right) U_\infty^3, \quad (1.1)$$

où ρ est la densité de l'air, R est le rayon du rotor, C_p le coefficient de puissance, qui est une quantité adimensionnelle qui peut être interprétée comme le rendement de la turbine, ω est la vitesse (angulaire) du rotor, β est l'angle de calage des pales (supposé le même pour toutes les pales), et U_∞ est la vitesse du vent loin en amont de l'éolienne. Cette formule suppose que la vitesse du vent est uniforme dans l'espace et perpendiculaire au rotor. Lorsque ce n'est pas le cas, U_∞ doit être remplacé par la vitesse effective du vent au rotor, qui est une moyenne de la composante perpendiculaire de la vitesse du vent sur la zone du rotor.

Le couple aérodynamique sur le rotor est $T_a = P_a/\omega$. On peut alors écrire

$$T_a = \frac{1}{2} \rho \pi R^3 C_q \left(\frac{\omega R}{U_\infty}, \beta \right) U_\infty^2, \quad (1.2)$$

où le coefficient de couple C_q est donné par

$$C_q \left(\frac{\omega R}{U_\infty}, \beta \right) = \frac{C_p \left(\frac{\omega R}{U_\infty}, \beta \right)}{\omega R / U_\infty}. \quad (1.3)$$

Une autre variable qui nous intéresse est le coefficient de poussée, C_T , qui relie la poussée sur le rotor en fonction de la vitesse du vent :

$$F_a = \frac{1}{2} \rho \pi R^2 C_T \left(\frac{\omega R}{U_\infty}, \beta \right) U_\infty^2. \quad (1.4)$$

Enfin, le déficit de quantité de mouvement subi par l'air circulant devant une éolienne est caractérisé par une quantité adimensionnelle a , le facteur d'induction, défini comme

$$a = \frac{U_\infty - U_d}{U_\infty}, \quad (1.5)$$

où U_d est la vitesse du vent au niveau du disque du rotor (rappelons que U_∞ est la vitesse en amont de l'éolienne). En effet, à mesure que la turbine

extrait l'énergie cinétique de l'air qui la traverse, la vitesse du vent diminue. Le facteur d'induction capte ainsi ce rapport de réduction. Le facteur d'induction est lié au coefficient de poussée C_T par la formule

$$a = \frac{1}{2} \left(1 - \sqrt{1 - C_T} \right). \quad (1.6)$$

A partir de ces définitions, un modèle dynamique simple pour la dynamique du rotor peut être écrit avec la conservation du moment angulaire appliqué au rotor de turbine :

$$J\dot{\omega} = T_a - T_g - T_l, \quad (1.7)$$

où J est l'inertie du rotor, T_a est le couple aérodynamique défini en (1.2), T_g est le couple de réaction causé par le générateur (qui est une entrée de contrôle) et T_l est un terme qui dénote les pertes dues, par exemple, à la friction. Ces pertes peuvent être modélisées plus précisément, comme cela est fait par exemple dans [22], mais sont aussi souvent négligées et transformées en incertitudes du modèle. De plus, dans le cas des éoliennes offshore flottantes, on s'intéresse également à la modélisation du mouvement de la plate-forme [12]. Notez que, à partir de (1.2), le modèle dynamique est non linéaire, et dépend également de manière non linéaire de la vitesse du vent U_∞ . En tant que telle et comme détaillé dans la suite, la fonction C_q (coefficient de couple) ou, de manière équivalente, C_p (coefficient de puissance) sont importantes pour le contrôle du parc éolien et l'estimation de la vitesse du vent. Mais il s'agit malheureusement de fonctions très non linéaires, variables d'une turbine à l'autre, pouvant ne pas être disponibles et de toute façon incertaines.

Nous détaillons maintenant ces enjeux de contrôle et d'estimation.

1.3 Contrôle des parcs éoliens et estimation du vent

Le contrôle des parcs éoliens est le processus d'optimisation des performances et du fonctionnement d'un parc éolien, c'est-à-dire d'un groupe d'éoliennes. Le contrôle du parc éolien vise à maximiser la production d'énergie, à minimiser les charges mécaniques et l'usure, et à assurer la stabilité et la fiabilité du réseau ; il peut être divisé en deux niveaux : le contrôle local et le contrôle central. Le contrôle local fait référence au contrôle individuel de chaque éolienne sur la base de ses propres mesures et paramètres, tandis que le contrôle central fait référence au contrôle coordonné de l'ensemble du parc éolien sur la base des informations provenant de toutes les éoliennes et de l'opérateur du réseau [36]. Dans un cas comme dans l'autre, les entrées de commande disponibles sont généralement : les angles de calage des pales, le

couple du générateur et l'angle de lacet (la direction dans laquelle l'éolienne est orientée), comme mentionné dans la section précédente.

Pour les deux niveaux de contrôle, local et central, il est possible de tirer parti de la connaissance des conditions de vent pour obtenir de meilleurs résultats, et c'est là que l'estimation du vent entre en jeu. Même si les éoliennes sont généralement équipées d'anémomètres et de girouettes pour mesurer la vitesse et la direction du vent, ces mesures sont fortement influencées par la présence des éoliennes elles-mêmes, et les quantités que l'on souhaite connaître sont les caractéristiques du vent en écoulement libre, c'est-à-dire sans les perturbations causées par les éoliennes.

1.3.1 Contrôle local et REWS

Dans le cas d'un contrôle local, on peut s'intéresser à l'estimation de ce que l'on appelle la vitesse effective du vent au rotor (*Rotor Effective Wind Speed*, REWS). Contrairement aux mesures ponctuelles de la vitesse du vent fournies par les anémomètres, la REWS représente la vitesse moyenne du vent agissant sur le rotor d'une éolienne.

L'importance de l'estimation de la REWS réside dans sa capacité à fournir des informations sur la vitesse angulaire optimale du rotor pour obtenir une puissance maximale. En s'appuyant sur la connaissance de la REWS, les contrôleurs d'éoliennes peuvent ajuster avec précision la vitesse du rotor, ce qui permet d'améliorer l'extraction d'énergie et les performances de l'éolienne. Par conséquent, l'estimation de la REWS est souvent utilisée pour optimiser le fonctionnement des éoliennes et, en fin de compte, augmenter la production d'énergie [59].

Une myriade de techniques d'estimation de la REWS ont été proposées dans la littérature, mettant en évidence la diversité des approches pour s'attaquer à ce problème difficile. Des chercheurs ont exploré des méthodes utilisant le filtre de Kalman, par exemple [22, 23], des observateurs non linéaires, par exemple [91, 113], des algorithmes d'apprentissage automatique, par exemple [2, 77, 132], et d'autres encore pour estimer la REWS à partir des données fournies par les éoliennes. Les méthodes peuvent-être les plus simples s'appuient sur une forte corrélation entre la puissance produite et la REWS et tentent simplement d'estimer la REWS sur la base de la puissance active [51]. Certaines méthodes plus sophistiquées utilisent une approche d'observateur d'état avec la REWS considérées comme un état dont la dynamique est pilotée par un bruit, par exemple [22], ou un observateur d'entrée [92], ce qui permet d'inclure la dynamique du rotor ou même la vibration de la tour dans la procédure d'estimation. Une autre approche est plus proche de celle que nous présentons ici, c'est l'approche *soft-sensor*, où l'éolienne est utilisée comme un dispositif de mesure indirect en trouvant une relation entre ce que l'éolienne mesure réellement (puissance produite, vitesse du rotor, etc.) et la REWS, ce qui est fait, par exemple, dans [132].

La plupart de ces méthodes reposent toutefois sur un modèle physique de l'éolienne à contrôler qui peut ne pas être disponible pour les exploitants de parcs éoliens, tels que les coefficients de poussée ou de couple, par exemple. Un aperçu de ces techniques est disponible dans [59].

Une autre approche consiste à utiliser des capteurs à longue distance tels que un dispositif lidar (du anglais *light detection and ranging*) pour obtenir des informations sur la REWS. Les dispositifs lidar permettent de mesurer à distance la vitesse du vent à différentes distances et hauteurs, offrant ainsi une vue d'ensemble du champ de vent. Plusieurs études [46, 49, 115] ont exploré l'utilisation du lidar pour l'estimation de la REWS, démontrant le potentiel de ces dispositifs dans l'optimisation des performances des parcs éoliens. Toutefois, l'une des limites est la disponibilité des dispositifs lidar, qui sont des capteurs coûteux et qui ne sont donc pas forcément accessibles dans tous les parcs éoliens et encore moins sur toutes les éoliennes.

Pour surmonter la contrainte de la disponibilité du lidar et le problème de la dépendance à un modèle physique, cette thèse propose une nouvelle approche qui utilise des algorithmes d'apprentissage automatique formés avec des données obtenues à partir de capteurs lidar. En entraînant ces algorithmes sur les données d'un modèle spécifique d'éolienne, ils peuvent être appliqués à d'autres éoliennes de ce même modèle dans des conditions similaires (telles que d'autres éoliennes dans le même parc éolien). Cette approche offre une solution pratique et rentable, permettant aux exploitants de parcs éoliens d'estimer la REWS et d'optimiser les performances des éoliennes, même en l'absence de données lidar directes et de modèles physiques.

1.3.2 Contrôle centralisé avec estimation du sillage

Les éoliennes ont traditionnellement été contrôlées individuellement, en employant des algorithmes gourmands où chaque éolienne vise à maximiser sa propre production d'énergie. Cependant, des recherches ont montré que ces approches ne sont pas nécessairement optimales lorsqu'on considère une fonction objective qui implique la performance de l'ensemble du parc éolien, comme la production globale d'énergie dans le parc [19]. Une approche plus efficace pour résoudre ce problème consiste à concevoir un contrôleur qui prend en compte les interactions aérodynamiques entre les turbines [124]. Dans la figure 1.4, nous avons une image du sillage des turbines mettant en évidence l'impact significatif des interactions aérodynamiques.

Les interactions entre les éoliennes proviennent de leur influence sur le vent au sein du parc éolien. Lorsqu'une éolienne extrait de l'énergie du vent, elle crée une zone de sillage en aval où la vitesse du vent est réduite par rapport au vent entrant. Par conséquent, les éoliennes situées dans cette zone subissent des vitesses de vent plus faibles, ce qui entraîne une diminution de la production d'énergie. La connaissance des conditions d'écoulement



FIGURE 1.4 : Parc éolien offshore de Horns Rev 1 au Danemark. Photographie de Christian Steiness / Vattenfall.

libre et la modélisation du sillage sont donc importantes pour prévoir la production d'énergie d'un parc éolien [93]. Les caractéristiques du sillage sont influencées par divers facteurs, tels que les conditions de vent libre, l'angle de désalignement entre l'éolienne et la direction du vent, et le facteur d'induction de l'éolienne, qui peut être contrôlé indirectement [3].

En principe, si nous connaissons les caractéristiques du vent en écoulement libre et un modèle pour le sillage généré par les turbines, il devient possible de contrôler l'ensemble du parc éolien afin de maximiser la production d'énergie ou d'atteindre d'autres objectifs spécifiques. Cela ouvre la voie à la conception de stratégies de contrôle sophistiquées qui optimisent les performances globales du parc éolien, en tenant compte des interactions dynamiques entre les turbines. Ces stratégies comprennent l'orientation du sillage, qui consiste à faire pivoter une éolienne de manière à la désaligner de la direction du vent afin de dévier son sillage de manière à réduire son influence sur les éoliennes situées en aval, et les méthodes de contrôle axial, qui consistent à faire fonctionner une éolienne à une vitesse de rotation sous-optimale (du point de vue du contrôle local) afin de réduire son effet sur les éoliennes situées en aval. Pour une analyse des méthodes de contrôle des parcs éoliens, consulter [3, 72, 88] et leurs références.

Dans tous les cas, comme nous l'avons mentionné, ces méthodes de contrôle reposent sur une modélisation du sillage. Par souci de complétude et pour une meilleure contextualisation, nous présentons donc certains de

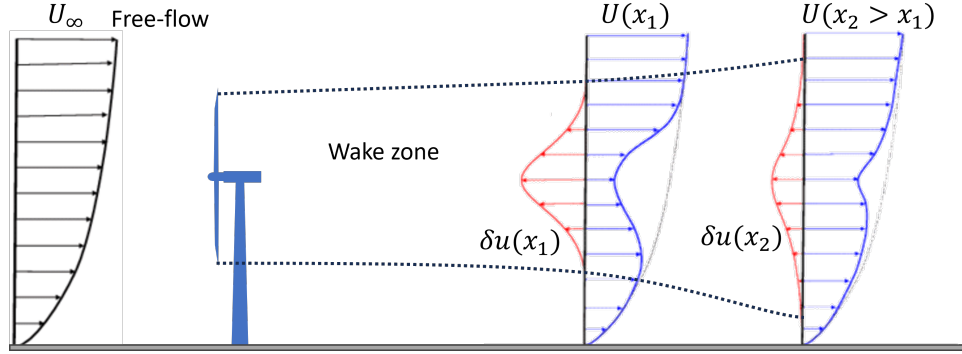


FIGURE 1.5 : Schéma du sillage d'une éolienne. Figure adaptée de [107, Fig. 2]

ces modèles dans la sous-section suivante.

1.3.3 Modélisation du sillage

Comme décrit précédemment, la zone de sillage d'une éolienne est la zone en aval affectée par l'éolienne où le vent souffre d'un déficit de vitesse par rapport à la condition d'écoulement libre en raison de son interaction avec l'éolienne. Le phénomène de sillage est complexe du point de vue de la dynamique des fluides car il implique des turbulences à différentes échelles de temps et de longueur [111] et, par conséquent, sa modélisation précise est une tâche difficile.

Différents modèles de sillage ont été présentés dans la littérature, en fonction de l'objectif considéré, et vont de modèles géométriques statiques à des solveurs sophistiqués de dynamique des fluides numériques (*Computational Fluid Dynamics*, CFD) [19]. Les deux caractéristiques fondamentales de tous les modèles sont les suivantes :

1. le déficit de vitesse tend vers zéro lorsque la distance sous le vent de l'éolienne tend vers l'infini et
2. la largeur de la zone de sillage augmente à mesure que la distance en aval de l'éolienne augmente.

Ces phénomènes sont illustrés à la figure 1.5, où la vitesse de l'écoulement libre U_∞ est supérieure à la vitesse U à l'intérieur de la zone de sillage, mais U est plus important à une position plus éloignée de la turbine sous le vent qu'à une position plus proche de la turbine, et le rayon de la zone de sillage augmente dans la direction du vent.

Cette thèse se concentre sur les modèles introduits dans [119, 120], qui sont expliqués en détail plus loin dans les chapitres concernés. Dans l'introduction, nous allons brièvement passer en revue quelques modèles pour

donner au lecteur une certaine intuition du phénomène de sillage et de ses implications dans les tâches d'estimation que nous visons ici. Pour d'autres modèles et une discussion détaillée sur le problème de la modélisation des sillages, voir, par exemple, [43, 68, 111] et leurs références.

Modèles statiques

Le modèle de sillage le plus simple que l'on puisse trouver dans la littérature est probablement le modèle de Jensen présenté dans [60], qui est également appelé modèle de Park par certaines sources telles que [6]. Ce modèle suppose une turbine d'un diamètre D fonctionnant à une vitesse d'écoulement uniforme constante U_∞ à un facteur d'induction constant a . Le plan du rotor est supposé être perpendiculaire à la direction constante du vent. Considérons un système de coordonnées cylindriques dont l'origine est le centre du plan du rotor et désignons par x la distance en aval et par r la distance radiale à partir de l'origine. Alors, la vitesse du vent au point (x, r) est donnée par

$$U(x, r) = U_\infty - \delta u(x, r), \quad (1.8)$$

où

$$\delta u(x, r) = \begin{cases} 2a \left(\frac{D}{D+2\kappa x} \right)^2, & \text{if } r \leq \frac{D}{2} + \kappa x, \\ 0, & \text{if } r > \frac{D}{2} + \kappa x, \end{cases} \quad (1.9)$$

et $\kappa > 0$ est un paramètre à régler. Les points clés de ce modèle sont qu'il suppose que le rayon du sillage croît linéairement avec la distance en aval et que le déficit de vitesse pour une distance en aval donnée est le même pour toute distance radiale tant qu'elle se trouve à l'intérieur de la zone de sillage.

Au lieu d'utiliser cette distribution uniforme pour le déficit, on peut aussi en utiliser une autre. C'est le cas du modèle proposé par [13], qui est similaire au modèle de Jensen, sauf que le déficit de vitesse est distribué radialement comme une gaussienne, de sorte qu'il est plus important au centre de la zone de sillage et devient progressivement plus petit dans la direction radiale.

Il est à noter que ces modèles ne tiennent pas compte de la déviation du vent causée par un désalignement entre l'éolienne et la direction du vent. Cette caractéristique a été incorporée dans [61]. Considérons que la turbine qui crée le sillage présente un angle de désalignement γ avec la direction de l'écoulement libre, comme sur la Fig. 2.6. Le déficit de vitesse dans [61] est alors calculé comme dans le modèle de Jensen, mais avec une zone de sillage tournée avec un angle α calculé comme suit

$$\alpha = \frac{\cos^2(\gamma) \sin(\gamma) C_T}{2 \left(1 + \frac{\kappa x}{D} \right)^2}, \quad (1.10)$$

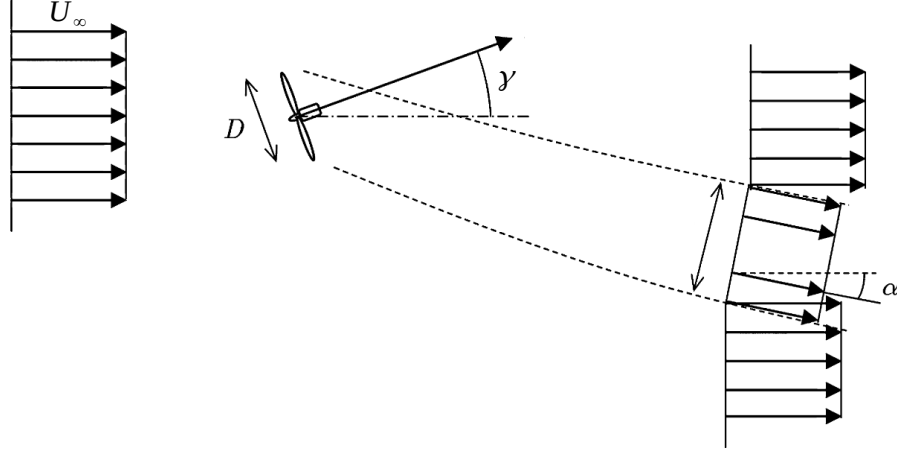


FIGURE 1.6 : Le modèle de sillage proposé dans [61]. Figure adaptée de [61, Fig. 5].

avec le même κ que dans le modèle de Jensen.

Même si ces modèles offrent un premier aperçu du phénomène du sillage, ils sont trop simplifiés pour fournir une modélisation précise pour contrôle et, plus important, ils ne fournissent que des relations statiques, ce qui les empêche de être utilisé dans le contexte d'une vitesse du vent variable dans le temps.

Modèles plus fidèles

D'autre part, il existe des modèles plus fidèles, basés sur la dynamique des fluides numérique.

L'une de ces modèles est FAST.Farm [66, 67], développé par le National Renewable Energy Laboratory (NREL), qui peut être considéré comme un simulateur de fidélité moyenne. Il est basé sur la combinaison du modèle *Dynamic Wake Meandering* (DWM) proposé dans [47] (qui est l'une des variations du modèle DWM proposé à l'origine par Larsen et al. [74, 75]) avec FAST¹ du NREL [64], qui est un simulateur d'éolienne haute-fidélité basé sur la théorie du momentum des éléments de pale (*blade-element momentum theory* en anglais) [76].

Le modèle DWM capture un mouvement latéral apparemment aléatoire du sillage (les méandres) ainsi que les variations temporelles et spatiales du déficit. Il s'agit d'un modèle tridimensionnel capable de gérer des conditions d'afflux variables dans le temps et dans l'espace. Pour ce faire, une version

¹Maintenant remplacé par OpenFAST [97].

simplifiée des équations de Navier-Stokes est résolue, les différentes échelles étant traitées séparément. Le DMW intègre trois sous-modèles différents : un modèle d'état stable pour le déficit de vitesse à l'intérieur de la zone de sillage basé sur une approximation des équations de Navier-Stokes, un modèle de transport de sillage qui propage le déficit, et un modèle de turbulence de sillage qui tient compte de la turbulence ajoutée par le sillage.

Le modèle de turbine, FAST, peut inclure des effets tels que la déflexion élastique de la tour et des pales, et le mouvement de la base (pour les éoliennes offshore). FAST.Farm fournit ainsi une représentation précise de l'effet de la turbine sur le champ de vent et de la propagation du sillage, ainsi que de la production d'énergie et des charges mécaniques sur les éoliennes d'un parc éolien. Le coût de calcul de FAST.Farm est suffisamment bas pour qu'il puisse être exécuté en un temps raisonnable sur un ordinateur personnel, mais il donne des résultats précis et a été validé par rapport à des simulateurs haute-fidélité et des données expérimentales par de multiples études [63, 73, 117, 118]. Dans cette thèse, nous l'utilisons donc comme référence pour valider nos techniques d'estimation.

Un exemple de simulateur haute fidélité est le "Simulator for Off / On-shore Wind Farm Applications" (SOWFA) du NREL [37], qui combine FAST avec un solveur CFD basé sur le logiciel CFD à usage général OpenFOAM [98]. Il utilise des simulations à grande échelle (*Large-Eddy Simulation*, LES) pour calculer très précisément le champ de vitesse du vent sur l'ensemble du parc éolien. SOWFA est probablement le simulateur haute fidélité le plus couramment utilisé par la communauté de recherche sur le contrôle des parcs éoliens, mais il existe d'autres simulateurs haute fidélité tels que le Parallelized Large-Eddy Simulation Model (PALM) [81] et d'autres modèles LES tels que ceux proposés dans [83, 89]. En général, les codes LES tels que SOWFA sont très gourmands en calculs, ne peuvent pas fonctionner en temps réel, et nécessitent un grappe d'ordinateurs pour être exécutés dans un temps raisonnable.

La charge de calcul de ces modèles les écarte de toute application en temps réel et les limite à des fins d'analyse. C'est pourquoi des modèles plus orientés vers le contrôle ont également été développés.

Modèles dynamiques simplifiés

À mi-chemin entre les modèles statiques simples et les modèles de plus haute fidélité que nous avons présentés, il existe des modèles dynamiques simplifiés.

Shapiro et ses collaborateurs ont introduit un modèle dynamique unidimensionnel [119] qui peut tenir compte de la variation temporelle des facteurs d'induction des turbines, et qui a été utilisé plus tard pour le cas où la vitesse du vent varie dans le temps [121], et une version bidimensionnelle qui tient compte du désalignement entre les turbines et la direction du vent [120].

Les modèles de Shapiro sont dérivés des premiers principes utilisant la conservation de la masse et de la quantité de mouvement ainsi que certaines hypothèses simplificatrices inspirées au modèle de Jensen pour obtenir une description du sillage donnée par des Équations Différentielles Partielles (EDP) de transport, ou une seule EDP de transport dans le cas unidimensionnel. Ces EDP décrivent la variation du déficit de vitesse causé par une turbine dans le temps et dans l'espace. Les turbines sont modélisées comme des puits de quantité de mouvement et sont paramétrées par le diamètre de leur rotor et les coefficients de poussée (ou facteurs d'induction). Le modèle 2D capture la déflexion du sillage et distribue le déficit de vitesse à l'intérieur de la zone de sillage dans la direction l'écoulement libre comme une gaussienne. Comme ces modèles sont ceux retenus dans cette thèse pour la conception des estimateurs de vent, leur mathématiques détaillées sont données plus loin dans ce manuscrit, au chapitre 4 pour le modèle unidimensionnel et au chapitre 5 pour le modèle bidimensionnel, respectivement.

Un autre modèle dynamique simple est SimWindFarm [44], où le déficit de vitesse en chaque point est calculé comme une simple fonction du coefficient de poussée et de la distance en aval, et où l'expansion du sillage ne dépend que de la distance. Ce modèle inclut le mouvement latéral de la ligne centrale du sillage en la modélisant comme un traceur passif. Il est donc en quelque sorte similaire au modèle de Shapiro, puisqu'il implique un phénomène de transport, mais avec un rayon d'expansion du sillage qui ressemble davantage aux modèles statiques présentés précédemment.

Un modèle plus complexe est FLORIDyn [15, 40], qui utilise des "points d'observation" créés à la position de l'éolienne et qui se déplacent ensuite sous le vent, de sorte que le sillage dans une certaine région à un certain moment est caractérisé par les états des points d'observation qui s'y trouvent. Un dernier exemple de modèle dynamique est WFSim [20], qui est dérivé directement des équations de Navier-Stokes avec quelques simplifications pour produire un modèle 2D axé sur le contrôle avec les turbines représentées par des termes de puits liés à leurs coefficients de poussée dans les EDP pour la vitesse du vent. Cependant, la question du coût de calcul de WFSim peut s'avérer non triviale [34].

Il est intéressant de noter que tous ces modèles ont en commun de présenter un phénomène de transport pour le sillage, afin de rendre compte de son expansion. De plus, grâce à leur coût de calcul relativement faible, ils peuvent être utilisés en temps réel [19] et donc dans des applications orientées vers le contrôle.

En comparaison avec l'estimation de la REWS, la littérature propose beaucoup moins de méthodes pour l'estimation du vent à l'échelle d'un parc éolien. Quelques études ont proposé d'utiliser un filtre de Kalman (ou ses variations) avec des mesures de puissance ou de vitesse [34, 40, 121], mais sans garantie de convergence et sans estimation de la direction de la vitesse du vent. D'autre part, certaines études se sont concentrées sur l'estimation

de la direction du vent, en utilisant généralement des mesures locales bruitées fournies par les éoliennes [5, 21, 35]. Cependant, l'estimation conjointe de la vitesse et de la direction du vent, sans mesures locales de la direction, n'a pas encore été réalisée.

1.4 Contributions et organisation de la thèse

Au vu de ces éléments, cette thèse aborde deux questions.

Premièrement, une nouvelle technique d'estimation de la REWS est conçue, au niveau d'une éolienne individuelle, dans chapitre 3. Pour surmonter la contrainte de la disponibilité du lidar et le problème de la dépendance à un modèle physique, cette thèse propose une nouvelle approche qui utilise des algorithmes d'apprentissage automatique formés avec des données obtenues à partir d'un capteur lidar (temporairement disponible). Suivant l'approche *soft-sensor* [132], où l'éolienne est utilisée comme dispositif de mesure indirect, nous proposons d'entraîner un modèle de régression par processus gaussien avec des mesures de bas niveau de l'éolienne (puissance produite, vitesse du rotor, etc.) pour estimer la REWS. Une fois ces algorithmes entraînés, le capteur lidar peut être retiré et utilisé dans une autre configuration. Cette approche offre une solution pratique et rentable, permettant aux exploitants de parcs éoliens d'estimer la REWS et d'optimiser les performances des turbines, même en l'absence de données lidar directes et de modèles physiques.

Deuxièmement, en se basant sur cette technique d'estimation locale de la REWS, le problème de l'estimation du vent au niveau du parc éolien est abordé. En supposant que des mesures locales des REWS soient disponibles, nous visons à reconstruire la vitesse et la direction du vent en écoulement libre, qui sont incertaines et non mesurées. Pour atteindre cet objectif, nous proposons de représenter le sillage avec les modèles de Shapiro (voir Section 2.3.3)[119, 120]. En effet, le cadre de Shapiro décrit le sillage par des EDP de transport 1-D en cascade, dont la vitesse de transport est la vitesse du vent en écoulement libre, c'est-à-dire la variable à estimer. Ces équations admettent des solutions analytiques exprimées par des retards de transport, ce qui ouvre la voie à des techniques d'estimation conformes en temps réel basées sur des estimations de retards. Dans cette thèse, nous étudions deux configurations différentes.

Dans le chapitre 4, nous considérons, à titre d'étude préliminaire, le cas où les éoliennes sont placées de manière à être alignées dans la direction du vent. Nous cherchons à estimer la vitesse du vent en écoulement libre pour le parc éolien sur la base d'une mesure locale de la vitesse affectée par le sillage. En utilisant le modèle de Shapiro et sa solution analytique du délai de transport, nous proposons une méthode d'intégrateur simple de "type observateur" pour effectuer cette estimation et nous fournissons un résultat

de convergence correspondant.

Ensuite, dans le chapitre 5, nous passons à une configuration plus générale, où aucune hypothèse n'est faite sur la géométrie du parc éolien. Notre objectif est d'estimer la vitesse et la direction du vent en écoulement libre à partir des mesures de vitesse locales fournies par les éoliennes du parc éolien. Dans cette optique, en modélisant le sillage avec le modèle de Shapiro, nous reformulons le problème considéré comme un problème d'identification. En plus des difficultés inhérentes aux problèmes à dimension infinie, un autre défi spécifique se pose : en raison de l'apparition de retards de transport dépendant de la vitesse du vent en écoulement libre, la variable estimée, l'étendue de la variable à estimer dépend elle-même de ses valeurs (inconnues). Pour résoudre ce problème, nous proposons de simplifier le problème en jeu en considérant un passé moyen constant pour la vitesse du vent en écoulement libre. Nous considérons et résolvons le problème équivalent à temps discret à l'aide de méthodes d'optimisation sous contraintes de dimension finie standards. Les simulations réalisées FAST.Farm soulignent l'intérêt de la méthode proposée.

1.5 Publications

Trois articles de conférence ont été publiés au long de cette thèse :

- Le référence [102] présente une méthode d'estimation de la REWS. Son contenu est présenté ici au chapitre 3.
- Les références [100, 101] sont liées à l'estimation de la vitesse de l'écoulement libre dans les parcs éoliens et leur contenu est présenté ici au chapitre 4.

Chapter 2

Introduction (in English)

2.1 Motivation

As of the end of 2022, the installed capacity of wind energy reached an impressive 906 GW globally, with 77.6 GW added during that year alone [55], an incredible increase compared to 7.5 GW of installed capacity in 1997 [57]. Figure 2.1 shows the growth of the installed capacity from 2011 to 2021. We can notice an overall exponential growth of the world-wide installed capacity and we can expect wind energy to play a major role in the energy sector in the foreseeable future.

With the expansion of wind farms and the increasing number of turbines placed in closer proximity, aerodynamic interactions among turbines have become a significant concern. These interactions can lead to wake effects, where downstream turbines experience reduced wind speeds and higher turbulence caused by the wakes of upstream turbines. Consequently, power production losses and a potential decrease in the lifespan of the turbines are observed, posing operational challenges for wind farm efficiency and overall energy production. Losses in power production usually range from 5% to 10% [114], but losses of up to 23% in power production have been reported in the literature, see [11] and references therein.

The negative impact of these aerodynamic interactions is particularly concerning because it augments the cost of energy production from wind farms. Reducing this cost is a critical objective that renewable energy sources, including wind energy, must achieve if they are to fully displace fossil fuels [108]. To overcome these challenges and optimize the operation of wind farms, advanced control technology plays a crucial role [26, 104]. In fact, by employing sophisticated control algorithms, wind farm operators can actively manage turbine settings, such as yaw alignment and pitch angles, to minimize wake effects and maximize energy capture from the available wind resource [72].

However, for such control strategies to be effective, they heavily rely

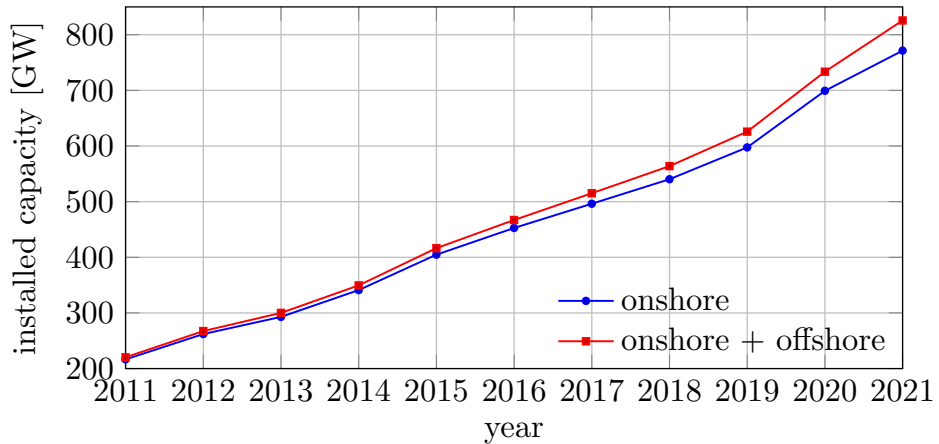


Figure 2.1: Yearly growth of the installed wind energy capacity from 2011 to 2021 (one data point per year). Data from IRENA [57].

on accurate and real-time information about the wind conditions within the wind farm. This is where estimation techniques come into play. The successful implementation of control technology is closely linked to the development of algorithms capable of estimating wind conditions in real-time. These estimation techniques provide essential information to the controllers, enabling them to make precise decisions on turbine settings and optimize power production in the face of dynamic and fluctuating wind patterns.

Thus, as wind energy continues to play a critical role in the global energy transition, addressing the challenges posed by increased wind farm density is of utmost importance. The interplay between control technology and advanced estimation techniques holds the key to mitigating aerodynamic interactions and optimizing wind farm performance. By harnessing the combined power of control and estimation, wind energy plants can efficiently harvest the abundant wind resource and contribute significantly to the global efforts in transitioning to sustainable energy.

The goal of this thesis is to contribute to addressing these challenges with the development of estimation algorithms for wind farms.

2.2 Preliminaries on Wind Turbines

Here, we briefly describe wind turbines of the type considered in this thesis, along with some modeling elements of the rotor dynamics

Description and geometry of a wind turbine

A wind turbine is a device that converts the wind's kinetic energy into electrical energy. We show a schematic of one in Fig. 2.2. The turbine

Wind turbine

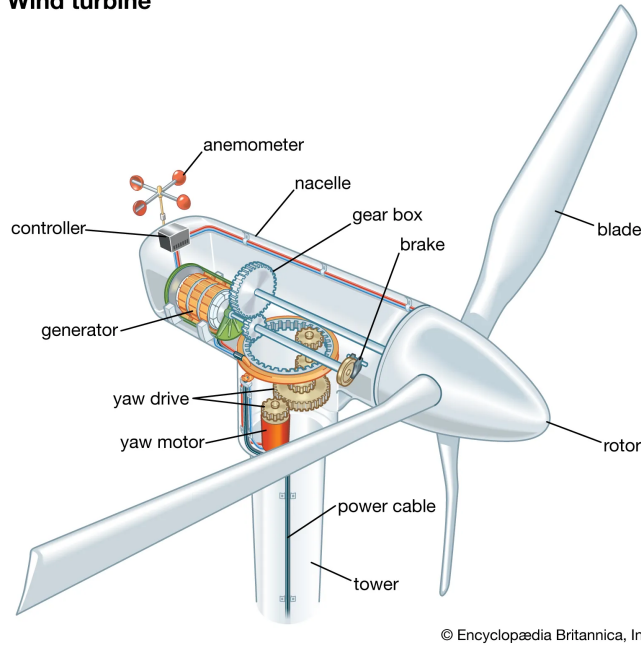


Figure 2.2: Schematic of a wind turbine.

depicted in Fig. 2.2 is a 3-bladed horizontal axis wind turbine which is the type that we consider in this thesis and the most common one. The basic principle of the operation is that the wind causes the rotor to spin and this mechanical energy is transferred to an electrical generator through a gear box to finally generate electricity; for more detailed descriptions of the components and different types of horizontal axis wind turbines, refer to, e.g., [8, 50]. Notice other types of wind turbines exist or have been proposed, e.g., airborne wind turbines [28, 128], bladeless wind turbines [126], twin wind turbines [45, 106], and vertical axis wind turbines [9, 79], but they are not considered here.

In terms of control, the goal is usually to maximize the produced electrical power, to minimize the mechanical fatigue, and/or to comply with grid requirements and the available control inputs generally are the pitch angles of the blades, the yaw angle of the nacelle (which determines the direction the turbine faces), and the generator torque [3]. In the case of offshore floating wind turbines, one also wants to minimize the pitch motion of the platform [134]. Furthermore, the operation of a turbine is typically characterized by a static power curve that gives the produced power as a function of the wind speed and this curve is divided into three regions of operation [104]:

- in Region 1, the wind is too slow, lower than the so-called cut-in speed,

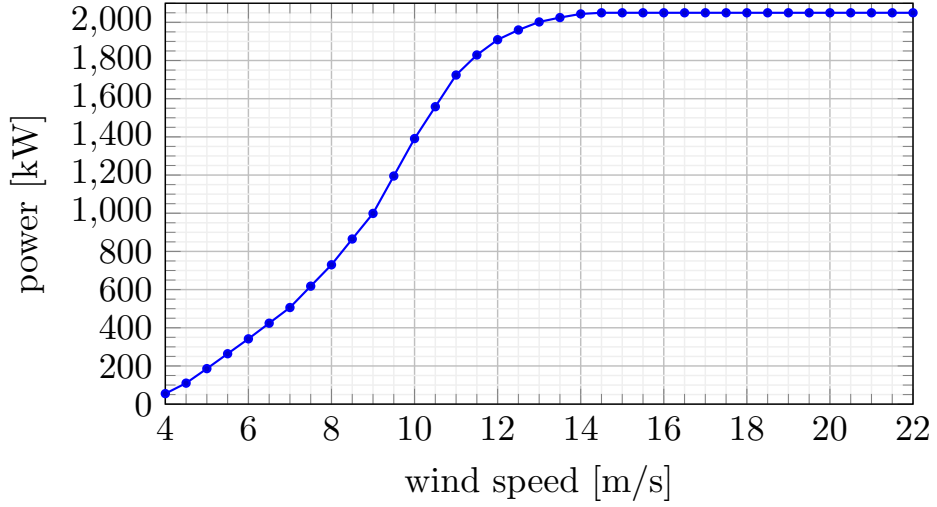


Figure 2.3: Power curve for a Senvion MM-82. The cut-in speed is 4 m/s, the rated speed is 14.5 m/s, the cut-off speed is 22 m/s, and the rated power is 2,050 kW. The data for the plot and the reference values were taken from [127].

for the available power to compensate the losses in the turbine systems and the turbine is not run;

- in Region 2, the wind is fast enough and the turbine is operated, but the wind is still not fast enough for the turbine to generate its rated power, i.e., below the rated speed, this is known as the below rated region;
- in Region 3, the wind is fast enough for the turbine to generate its rated power, this region is known as the above rated region.

The control goals can change depending on the operating region: in Region 2, normally the goal is to maximize the power production, the pitch angles are kept fixed in their optimal value for this purpose and the control input used is the generator torque [1, 103, 112], whereas in Region 3, the goal is usually to minimize the mechanical loads on the turbine while maintaining the power production at the rated level and pitch angle controllers are used [27, 30, 134]. We can also define a Region 4, which is above the so-called cut-off speed, where the turbine does not run. For instance, Fig. 2.3 shows the power curve for a Senvion MM-82, which is the model of the wind turbine we use experimental data of in this thesis.

Dynamic modeling

In the following, we will present a basic modeling of the rotor dynamics that is often invoked in the wind turbine control literature [87, 104].

We start by introducing the expression for the aerodynamic power captured by the rotor, P_a ,

$$P_a = \frac{1}{2} \rho \pi R^2 C_p \left(\frac{\omega R}{U_\infty}, \beta \right) U_\infty^3, \quad (2.1)$$

where ρ is the air density, R is the radius of the rotor, C_p the power coefficient, which is a non-dimensional quantity that can be interpreted as the efficiency of the turbine, ω is the rotor (angular) speed, β is the pitch angle of the blades (assumed the same for all blades), and U_∞ is the wind speed far upwind from the wind turbine. This formula assumes the wind velocity is uniform in space and perpendicular to the rotor. When this is not the case, U_∞ should be replaced by the Rotor Effective Wind Speed (REWS), which is an average of the perpendicular component of the wind velocity over the rotor area.

The aerodynamic torque on the rotor is $T_a = P_a/\omega$. Then, we can write

$$T_a = \frac{1}{2} \rho \pi R^3 C_q \left(\frac{\omega R}{U_\infty}, \beta \right) U_\infty^2, \quad (2.2)$$

where the torque coefficient C_q is given by

$$C_q \left(\frac{\omega R}{U_\infty}, \beta \right) = \frac{C_p \left(\frac{\omega R}{U_\infty}, \beta \right)}{\omega R / U_\infty}. \quad (2.3)$$

Another variable of interest is the thrust coefficient, C_T , that relates the thrust on the rotor to the wind speed:

$$F_a = \frac{1}{2} \rho \pi R^2 C_T \left(\frac{\omega R}{U_\infty}, \beta \right) U_\infty^2. \quad (2.4)$$

Finally, the momentum deficit experienced by air flowing past a wind turbine is characterized by a non-dimensional quantity a , the induction factor, defined as

$$a = \frac{U_\infty - U_d}{U_\infty}, \quad (2.5)$$

where U_d is the wind speed at the rotor disk (recall that U_∞ is the speed far upwind from the turbine). Indeed, as the turbine extracts kinetic energy from the air that passes through it, the wind speed reduces. The induction factor thus captures this ratio of reduction. The induction factor is related to the thrust coefficient C_T by the formula

$$a = \frac{1}{2} \left(1 - \sqrt{1 - C_T} \right). \quad (2.6)$$

From these definitions, a simple dynamic model for the rotor dynamics can be written with the conservation of angular momentum applied to the turbine rotor:

$$J\dot{\omega} = T_a - T_g - T_l, \quad (2.7)$$

where J is the rotor inertia, T_a is the aerodynamic torque defined in (2.2), T_g is the reaction torque caused by the generator (which is a control input), and T_l is a lump term that denotes losses due to, e.g., friction. These losses can be modeled more precisely, as done, e.g., in [22], but are also often neglected and recast as model uncertainties. Furthermore, in the case of offshore floating wind turbines, one is also interested in modeling the motion of the platform [12].

Notice that, from (2.2), the dynamical model is nonlinear, and also depends nonlinearly on the wind speed U_∞ . As such and as detailed in the sequel, the torque coefficient map C_q or, equivalently, the power coefficient map C_p are important for wind farm control and wind speed estimation. Yet, unfortunately, they are highly nonlinear functions which vary from one turbine to another, may not be available and are, in any case, uncertain.

We now detail these control and estimation challenges.

2.3 Wind farm control and wind estimation

Wind farm control is the process of optimizing the performance and operation of a wind farm, i.e., a group of wind turbines. Wind farm control aims to maximize the energy production, minimize the mechanical loads and wear, and ensure the stability and reliability of the grid; it can be divided into two levels: local control and central control. The local control level refers to the individual control of each wind turbine based on its own measurements and parameters, whereas central control level refers to the coordinated control of the whole wind farm based on the information from all the wind turbines and the grid operator [36]. In either case, the control inputs available generally are: the pitch angles of the blades, the generator torque, and the yaw angle (the direction at which the wind turbine is pointing) as mentioned in the previous section.

For both levels of control, local and central, one can leverage knowledge about the wind conditions to achieve better results, and that is where wind estimation comes in. Even though wind turbines are generally equipped with anemometers and wind vanes to measure the wind speed and direction, these measurements are highly impacted by the presence of the turbines themselves and the quantities we desire to know are the free-flow characteristics of the wind, i.e., without the disturbances caused by the turbines.

2.3.1 Local control and REWS

In the case of local control, one can be interested in estimating the so called Rotor Effective Wind Speed (REWS). Unlike single-point wind speed measurements provided by anemometers, the REWS represents the average wind speed acting on the rotor of a wind turbine.

The significance of REWS estimation lies in its ability to inform the optimal rotor angular speed for extracting maximum power. By leveraging the knowledge of the REWS, wind turbine controllers can fine-tune the rotor speed, leading to improved power extraction and enhanced turbine performance. As a result, REWS estimation is often used for optimizing wind turbine operation and ultimately increasing power production [59].

A myriad of REWS estimation techniques have been proposed in the literature, showcasing the diversity of approaches to tackle this challenging problem. Researchers have explored methods utilizing the Kalman filter, e.g., [22, 23], nonlinear observers, e.g., [91, 113], machine learning algorithms, e.g., [2, 77, 132], and more to estimate the REWS from the data provided by the wind turbines. Perhaps the simplest methods are the ones that rely on a large correlation between the produced power and the REWS and simply try to estimate the REWS based on the active power alone [51]. Some more sophisticated methods use a state-observer approach with the REWS considered as a state whose dynamics are driven by noise, e.g., [22] or an input observer [92], this allows the inclusion of the rotor dynamics or even the tower vibration into the estimation procedure. Most of these methods, however, rely on a physical model of the turbine to be controlled which might not be available to the wind farm operators, such as the power or torque coefficients, for instance. Refer to [59] for a comprehensive review of these techniques.

Another approach is to use long-distance sensors like LiDAR (Light Detection And Ranging) for obtaining the REWS information. LiDAR devices allow for remote measurements of wind speeds at different distances and heights, providing a comprehensive view of the wind field. Several studies [46, 49, 115] have explored the use of LiDAR measurements for REWS estimation, demonstrating the potential of these devices in optimizing wind farm performance. However, one limitation is the availability of LiDAR devices, which are expensive sensors and thus may not be universally accessible in all wind farm locations and even less on every wind turbine.

2.3.2 Centralized control with wake estimation

Wind turbines have conventionally been controlled individually, employing a greedy approach where each turbine aims to maximize its own power production. However, research has shown that these greedy approaches may not necessarily be optimal when considering an objective function that involves



Figure 2.4: Offshore wind farm Horns Rev 1 in Denmark. Photograph by Christian Steiness / Vattenfall.

the entire wind farm's performance, such as the overall power production of the wind farm [19]. A more effective approach to address this issue is to design a controller that takes into account the aerodynamic interactions among turbines [124]. In Fig. 2.4, we have a picture of a wind farm taken under specific meteorological conditions such that the turbines' impact on the wind can be observed highlighting the significant impact of aerodynamic interactions.

The interactions among turbines stem from their influence on the wind within the wind farm. As a turbine extracts energy from the wind, it creates a wake zone downwind of itself where the wind speed is reduced compared to the incoming wind. Consequently, downstream turbines within this wake zone experience lower wind speeds, leading to decreased power production. Knowledge of the free-flow conditions and wake modeling are thus also important for forecasting the power production of a wind farm [93]. The characteristics of the wake are influenced by various factors, such as the free-flow wind conditions, the misalignment angle between the turbine and the wind direction, and the turbine's induction factor.

In principle, if one possesses knowledge of the free-flow wind characteristics and a model for the wake generated by the turbines, it becomes possible to control the ensemble of the wind farm to maximize power production or achieve other specific objectives. This opens the door to the design of sophisticated control strategies that optimize the overall performance of the

wind farm, taking into account the dynamic interactions among turbines. These strategies include (i) wake steering, where a turbine is yawed so that it is misaligned with the wind direction on purpose to deflect its wake so that its influence on downwind turbines is reduced and (ii) axial control methods where one turbine operates under a sub-optimal rotational speed (from the local control perspective) to reduce its effect on the downwind turbines. For reviews on wind farm control methods, see [3, 72, 88] and references therein.

In any case, as mentioned, these control methods rely on a modeling of the wake. For completeness and better contextualization, we thus present some of these models in the following subsection.

2.3.3 Wake modeling

As previously described, the wake zone of a turbine is the downwind area affected by the turbine where the wind suffers a speed deficit compared to the free-flow condition due to its interaction with the turbine. The wake phenomenon is complex from the point of view of fluid dynamics as it involves turbulence at different time and length scales [111] and, thus, precise modeling of it is a difficult task.

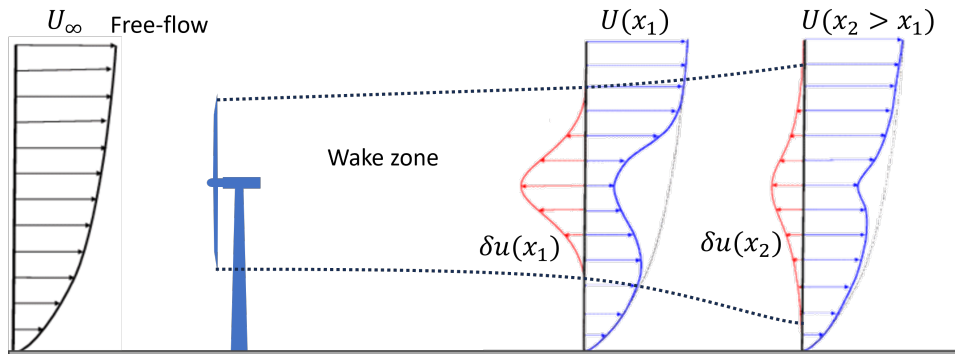


Figure 2.5: Schematic of the wake of a wind turbine. Adapted from [107, Fig. 2].

Different wake models have been introduced in the literature, depending on the objective under consideration, and range from static geometric models to sophisticated Computational Fluid Dynamics (CFD) solvers [19]. The two basic characteristics that all the models have are that:

1. the speed deficit tends to zero as the downwind distance from the turbine tends to infinity; and
2. the width of the wake zone increases as the downwind distance from the turbine increases.

These are illustrated in Fig. 2.5, where the free-flow speed U_∞ is larger than the speed U inside the wake zone, but U is larger at a position further

downwind from the turbine than at a position closer to the turbine, and the radius of the wake zone increases in the downwind direction.

We now provide a brief overview of the wake models available in the literature. For more models and detailed discussion on the wake modeling problem, the interested reader is referred to, e.g., [43, 68, 111] and references therein.

Static models

The simplest wake model one can encounter in the literature is probably the Jensen model introduced in [60], which is also referred to as the Park model by some sources such as [6]. This model assumes a turbine with diameter D operating in a constant uniform free-flow speed U_∞ at a constant induction factor a . The rotor plane is assumed to be perpendicular to the constant wind direction.

Consider a cylindrical coordinate system with its origin at the center of the rotor plane and denote by x the downwind distance and by r the radial distance from the origin. Then, the wind speed at the point (x, r) is given by

$$U(x, r) = U_\infty - \delta u(x, r), \quad (2.8)$$

where

$$\delta u(x, r) = \begin{cases} 2a \left(\frac{D}{D+2\kappa x} \right)^2, & \text{if } r \leq \frac{D}{2} + \kappa x, \\ 0, & \text{if } r > \frac{D}{2} + \kappa x. \end{cases} \quad (2.9)$$

and $\kappa > 0$ is a parameter to be tuned. The key points of this model are that it assumes the radius of the wake grows linearly with the downwind distance and the speed deficit for a given downwind distance is the same for any radial distance as long as it is inside the wake zone.

Instead of using this top-hat distribution for the deficit, one can also use a smoother distribution. This is the case of the model proposed by [13], which is similar to the Jensen model, except that the speed deficit is distributed radially as a Gaussian so that it is larger in the center of the wake zone and becomes gradually smaller in the radial direction.

Observe that these models do not account for the wind deflection caused by a misalignment between the turbine and the wind direction. This characteristic was incorporated in [61]. Consider that the turbine creating the wake presents a misalignment angle γ with the free flow direction, as in Fig. 2.6. The speed deficit in [61] is then computed as in the Jensen model but with a wake zone rotated with an angle α computed as

$$\alpha = \frac{\cos^2(\gamma) \sin(\gamma) C_T}{2 \left(1 + \frac{\kappa x}{D} \right)^2}, \quad (2.10)$$

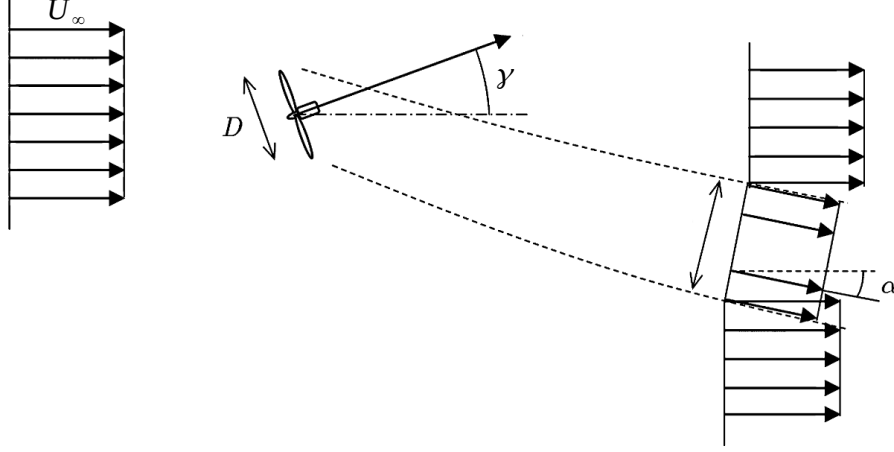


Figure 2.6: The wake model proposed in [61]. Adapted from [61, Fig. 5].

with the same κ as in the Jensen model.

Even if these models provide some insights on the wake phenomenon, they are too simplified to provide an accurate modeling for control and, more importantly, they only provide static relations, discarding them from any use in the context of a time-varying wind speed.

Higher-fidelity models

On the other hand, higher-fidelity models exist, based on Computational Fluid Dynamics.

One of these models is FAST.Farm [66, 67], developed by the National Renewable Energy Laboratory (NREL), which can be considered as a mid-fidelity simulator. It is based on combining the Dynamic Wake Meandering (DWM) model proposed in [47] (which is one of the variations of the DWM model originally proposed by Larsen et al. [74, 75]) with NREL's FAST¹ [64], a high-fidelity wind turbine simulator based on the blade-element momentum theory [76].

The DWM model captures a seemingly random sideways motion of the wake (the meandering) and also the temporal and spatial variations of the deficit. It is a 3-D model capable of handling time- and space-varying free-flow conditions. This is done by solving a simplified version of the Navier-Stokes equations, with different scales treated separately. Namely, the DMW incorporates three different submodels: a steady-state wake-deficit²

¹Now replaced by OpenFAST [97].

²The speed deficit inside the wake zone.

model based on an approximation of the Navier-Stokes equations, a wake-transport model that propagates the deficit, and a wake-turbulence model that accounts for the turbulence added by the wake.

The turbine model, FAST, can include effects such as elastic deflection of the tower and the blades, and movement of the base (for offshore wind turbines). FAST.Farm thus provides an accurate representation of the effect of the turbine on the wind field and of the propagation of the wake and also of the power production and the mechanical loads on the wind turbines of a wind farm. FAST.Farm’s computational cost is low enough that it can run in reasonable time on a personal computer, but it yields precise results and has been validated against high-fidelity simulators and experimental data by multiple studies [63, 73, 117, 118]. In this thesis, we thus use it as a reference to validate our estimation techniques.

An example of high-fidelity simulator is NREL’s Simulator for Off / On-shore Wind Farm Applications (SOWFA) [37], which combines FAST with a CFD solver based on the general purpose CFD software OpenFOAM [98]. It uses Large-Eddy Simulations (LESs) to very accurately compute the wind velocity field on the entire wind farm. SOWFA is probably the high-fidelity simulator most commonly used by the wind farm control research community, but there are other high-fidelity simulators such as the Parallelized Large-Eddy Simulation Model (PALM) [81] and other LES models such as the ones proposed in [83, 89]. Generally, LES codes such as SOWFA are very computationally intensive, cannot run in real-time, and require a cluster of computers to be executed in a reasonable time.

The computational burden of these models discard them from any real-time implementation, and limit them to analysis purposes. This is why more control-oriented models were also developed.

Simplified dynamic models

As a compromise between the simple static models and the computationally intensive higher-fidelity models presented previously, simplified dynamic control-oriented models were also developed.

Shapiro and collaborators introduced a 1-D dynamic model [119] that can account for the time-variation of the turbines’ induction factors, and was later used for the case of time-varying wind speed [121], and a 2-D version that accounts for misalignment between the turbines and the wind direction [120].

Shapiro’s models are derived from first principles using mass and momentum conservation together with some simplifying assumptions inspired by Jensen’s model to achieve a description of the wake given by cascaded transport Partial Differential Equations (PDEs), or a single transport PDE in the 1-D case. These PDEs describe the variation over time and space of the speed deficit caused by a turbine. The turbines are modeled as sinks

of momentum and they are parameterized by their rotor diameters and thrust coefficients (or induction factors). The 2-D model also captures the wake deflection and distributes the speed deficit inside the wake zone in the streamwise direction as a Gaussian. As these models are the ones retained in this thesis for wind estimators design, their detailed mathematical descriptions are given later in this manuscript, in Chapter 4 for the 1-D model and in Chapter 5 for the 2-D model, respectively.

Another simple dynamic model is SimWindFarm [44], where the deficit at each point is computed as a simple function of the thrust coefficient and the downwind distance and the wake expansion depends only on the distance. This model includes the sideways motion of the wake centerline by modeling it as a passive tracer. It is thus somehow similar to Shapiro’s model, as it involves transport phenomenon, but with a wake expansion radius resembling more the static models presented earlier.

A more involved model is FLORIDyn [15, 40], which uses “observation points” that are created at the position of the turbine and then travel downwind so the wake at a certain region at a certain time is characterized by the states of the observation points that are there. Finally, let us mention WFSim [20], which is a model derived directly from the Navier-Stokes equations with some simplifications to yield a control-oriented 2-D model with the turbines represented by sink terms related to their thrust coefficients in the PDEs for the 2-D wind speed velocity. Yet, the issue of computational cost of WFSim may reveal nontrivial [34].

Interestingly, note that all these models have in common to feature a transport phenomenon for the wake, to account for its expansion. Besides, due to their relatively low computational cost, they can be used in real time [19] and thus in control-oriented applications.

Overall, compared to REWS estimation, much fewer methods are available in the literature to wind farm-wide estimation. A few studies have proposed to use a(n) (ensemble) Kalman filter together with power measurements or speed measurements [34, 40, 121], but without any convergence guarantee and without wind direction estimation. On the other hand, some studies have focused on wind direction estimation, usually using some noisy local measurements provided by the turbines [5, 21, 35]. Yet, the joint estimation of wind speed and wind direction, without local wind direction measurements, remains to be carried out.

2.4 Thesis contributions and outline

In light of these elements, this thesis addresses two questions.

First, a new REWS estimation technique is designed, at the level of an individual turbine, in Chapter 3. To overcome the constraint of LiDAR availability and the problem of dependence on a physical model, this thesis

proposes a novel approach that utilizes machine learning algorithms trained with data obtained from a (temporarily available) LiDAR sensor. Following the so-called soft-sensor approach [132], where the turbine is used as an indirect measurement device, we proposed to train a Gaussian process regression model with low-level measurements taken by the turbine (produced power, rotor speed, and pitch angle) to estimate the REWS. After these algorithms are trained, the LiDAR sensor can be removed, and used in an alternative set-up. This approach offers a practical and cost-effective solution, enabling wind farm operators to estimate the REWS and optimize turbine performance even when direct LiDAR data and physical models are not available.

Second, building on this local REWS estimation technique, the problem of wind estimation at the level of the wind farm is then addressed. Assuming that local REWS measurements are available, we aim at reconstructing the free-flow wind speed and wind direction, which are uncertain and unmeasured. To achieve this objective, we propose to represent the wake with Shapiro’s models (see Subsection 2.3.3) [119, 120]. Indeed, Shapiro’s framework describes the wake through cascaded 1-D transport PDEs, the transport speed of which is the free-flow wind speed, that is, the variable to be estimated. As these equations admit analytical solutions expressed through transport delays, this opens the door for real-time compliant estimation techniques based on time-delay estimates. In this thesis, we investigate two different set-ups.

In Chapter 4, we consider, as a preliminary study, the case where the turbines are placed so that they are aligned along the wind direction. We aim at estimating the free-flow wind speed for the wind farm based on one local speed measurement affected by the wake. Using Shapiro’s model and its transport delay analytical solution, we propose a simple “observer-like” integrator method to perform this estimation and we provide a corresponding convergence result.

Then, in Chapter 5, we move to a more general set-up, where no assumption is made on the geometry of the wind farm. We aim at estimating both the free-flow wind speed and wind direction from local speed measurements provided by the turbines in the wind farm. With this aim in view, modeling the wake with Shapiro’s model, we recast the problem under consideration as an identification problem. In addition to the inherent difficulties related to infinite-dimensional problems, another specific challenge arises: due to the appearance of transport delays depending on the free-flow wind speed, i.e., the estimated variable, the range of the variable to be estimated depends itself on its (unknown) values. To address this issue, we propose to simplify the problem at stake by considering an average constant past history for the free-flow wind speed. We consider and solve the equivalent discrete-time problem using standard constrained finite-dimensional optimization methods. Simulations performed with FAST.Farm emphasize the interest of the

proposed method.

2.5 Publications

Three conference papers were published throughout this thesis:

- Reference [102] presents a REWS estimation method. The contents of it are presented here in Chapter 3.
- References [100, 101] are related to free-flow speed estimation in wind farms and their contents are presented here in Chapter 4.

Chapter 3

Rotor effective wind speed estimation

Résumé. Ce chapitre s'intéresse à l'estimation de la vitesse effective du rotor (REWS) d'une éolienne individuelle, sans avoir recours de façon pérenne à des données de vitesse de vent fournies par capteurs lidar ou à la cartographie de coefficient de couple de l'éolienne, informations qui ne sont pas toujours disponibles.

Pour contourner ces contraintes, ce chapitre propose d'utiliser des algorithmes d'apprentissage automatique formés à l'aide de données de vitesse du vent obtenues à partir d'un capteur lidar (disponible temporairement, pour l'apprentissage). Ces algorithmes utilisent des données de bas niveau provenant du système de contrôle et d'acquisition de données (SCADA) d'une éolienne industrielle. Ces données incluent la vitesse angulaire du rotor, la puissance active et l'angle de calage des pales. Une fois les algorithmes entraînés, le capteur lidar n'est plus nécessaire.

Deux méthodologies alternatives d'estimation sont détaillées. La première est entièrement basée sur les données et utilise la Régression par Processus Gaussien (GPR) pour estimer directement la vitesse du vent à partir des mesures de l'éolienne. La seconde combine une approche basée sur les données et une approche par modèle pour augmenter la robustesse de l'estimation.

3.1 Introduction

In this chapter, we investigate the problem of estimating the Rotor Effective Wind Speed (REWS) of an individual turbine, that is, the average of the wind speed acting on the turbine the rotor area without the effect of the presence of the turbine itself. The REWS is not measurable by an anemometer placed at the turbine's nacelle, since such anemometer only produces a punctual measurement that is affected by the presence of the turbine, in any

case. As mentioned in the Introduction, most REWS estimation techniques rely on knowledge of the torque coefficient map C_q of the turbine, which may not be known by the operator of the turbine, or on Light Detection And Ranging (LiDAR) sensors, which are expensive sensors.

Hence, to overcome this constraint of LiDAR availability or dependence on a physical model, in this chapter, we propose to use machine learning algorithms trained with wind speed data obtained from a (temporarily available) LiDAR sensor with low-level data obtained from the Supervisory Control And Data Acquisition (SCADA) system of an industrial-scale wind turbine, namely, the rotor angular speed, the active power, and the pitch angle of the blades. After these algorithms are trained, the LiDAR sensor can be removed and used in an alternative set-up.

We detail two alternative estimation methodologies. The first one is a purely data-based method, where Gaussian Process Regression (GPR) is performed to estimate the wind speed directly from the low-level measurements of the turbine (rotor angular speed, active power, and pitch angle). The second one, alternatively, consists in identifying the aerodynamic torque with a GPR similarly trained and to recover the wind speed from a high-gain observer. This second mixed data/model-based technique aims at increasing the robustness of the estimation, by providing the user with a measure of the current accuracy of the GPR (which could vary with time, due to aging of the turbine with fatigue, for instance).

In all this chapter, the SCADA data used come from a 3-bladed horizontal axis wind turbine model Senvion MM82 (82 m rotor diameter and 2.05 MW rated power) and the LiDAR data comes from a LiDAR sensor mounted on the nacelle of this same turbine. The SCADA data were provided by Engie Green and the LiDAR data by Leosphere within the framework of the project “SmartEole” of the French National Research Agency (ANR).

3.2 Estimating the REWS with a machine learning approach

This section presents an approach to estimate the REWS with GPR. In detail, this method estimates the REWS at time t given a sequence of measurements of the turbine’s rotor angular speed, active power, and pitch angle (assumed to be the same for all blades) from time $t - r\Delta t$ to time t , where r is a natural number chosen a priori, and Δt is the sampling period. For simplicity, we assume $\Delta t = 1$ throughout this section.

Let ω_t , P_t , β_t , and v_t denote the rotor angular speed, active power, pitch

angle, and the REWS at time t for a given turbine. We define

$$\mathbf{x}_t = \begin{bmatrix} \boldsymbol{\omega}_{t-r:t} \\ \boldsymbol{\beta}_{t-r:t} \\ \mathbf{P}_{t-r:t} \end{bmatrix}, \quad (3.1)$$

where the time-series of SCADA measurements $\boldsymbol{\omega}_{t-r:t}$, $\boldsymbol{\beta}_{t-r:t}$, and $\mathbf{P}_{t-r:t}$ are defined as

$$\boldsymbol{\omega}_{t-r:t} = [\omega_{t-r}, \dots, \omega_t]^T, \quad (3.2)$$

$$\boldsymbol{\beta}_{t-r:t} = [\beta_{t-r}, \dots, \beta_t]^T, \quad (3.3)$$

$$\mathbf{P}_{t-r:t} = [P_{t-r}, \dots, P_t]^T. \quad (3.4)$$

Then, the goal is to estimate $y_t = v_t$ given $\mathbf{x}_t$. To achieve this goal, we will use GPR, the basis of which we thus briefly present in the next subsections, before detailing the REWS estimation method itself.

3.2.1 Gaussian process regression

This section briefly describes the basics of GPR. We start with the definition of a Gaussian process.

Definition 3.1. *A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution [130, Definition 2.1].*

Let f be a process from $\mathbb{R}^{n_x}$ to $\mathbb{R}$, with $n_x \in \mathbb{N}$. Define its mean function $m : \mathbb{R}^{n_x} \mapsto \mathbb{R}$ and its covariance function (or kernel) $k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \mapsto \mathbb{R}$, such that

$$m(\mathbf{x}) = \mathcal{E}[f(\mathbf{x})], \quad (3.5)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathcal{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))], \quad (3.6)$$

where $\mathcal{E}$ denotes mathematical expectation and $\mathbf{x}$ and $\mathbf{x}'$ are points in $\mathbb{R}^{n_x}$. We write $f \sim \mathcal{GP}(m, k)$ to denote f as a Gaussian process with mean m and covariance k . The random variables are the values outputted by f , hence, given an input location $\mathbf{x} \in \mathbb{R}^{n_x}$, we have $f(\mathbf{x}) | \mathbf{x} \sim \mathcal{N}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}))$. Similarly, for N input locations $\mathbf{x}_i \in \mathbb{R}^{n_x}$, $i = 1, \dots, N$, the values outputted by f for these inputs have a joint Gaussian distribution,

$$\begin{bmatrix} f(\mathbf{x}_1) \\ \vdots \\ f(\mathbf{x}_N) \end{bmatrix} | \mathbf{X} \sim \mathcal{N}(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X})), \quad (3.7)$$

where $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T \in \mathbb{R}^{N \times n_x}$, $m(\mathbf{X})$ is a vector in $\mathbb{R}^N$ such that its i th element is $m(\mathbf{x}_i)$, $k(\mathbf{X}, \mathbf{X})$ is a matrix in $\mathbb{R}^{N \times N}$ such that its element at position (i, j) is $k(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathcal{N}(m(\mathbf{X}), k(\mathbf{X}, \mathbf{X}))$ is the Gaussian distribution with mean $m(\mathbf{X})$ and covariance matrix $k(\mathbf{X}, \mathbf{X})$.

To use GPR, we first assume a certain measured variable y_i is related to the input $\mathbf{x}_i$ by

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad (3.8)$$

with $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ and $f \sim \mathcal{GP}(m, k)$ for some suitable choice of m and k . Thus, for N training pairs $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$, we have

$$\mathbf{y}|\mathbf{X} \sim \mathcal{N}(m(\mathbf{X}), \mathbf{K}_y), \quad (3.9)$$

where $\mathbf{y} = [y_1, \dots, y_N]^T$, $\mathbf{K}_y = k(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}$, and $\mathbf{I}$ is the identity matrix of appropriate dimensions.

Next, consider test input locations $\mathbf{x}_{*i}$, $i = 1, \dots, N_*$, for which one wants to estimate the corresponding values of f . Let $\mathbf{f}_* = [f(\mathbf{x}_{*1}), \dots, f(\mathbf{x}_{*N_*})]^T$ and $\mathbf{X}_* = [\mathbf{x}_{*1}, \dots, \mathbf{x}_{*N_*}]^T$. Then, from the joint Gaussian distribution property of the Gaussian process,

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} | \mathbf{X}, \mathbf{X}_* \sim \mathcal{N} \left(\begin{bmatrix} m(\mathbf{X}) \\ m(\mathbf{X}_*) \end{bmatrix}, \begin{bmatrix} \mathbf{K}_y & k(\mathbf{X}, \mathbf{X}_*) \\ k(\mathbf{X}_*, \mathbf{X}) & k(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix} \right), \quad (3.10)$$

where $m(\mathbf{X}_*) = [m(\mathbf{x}_{*1}), \dots, m(\mathbf{x}_{*N_*})]^T$, $k(\mathbf{X}, \mathbf{X}_*) = [k(\mathbf{X}_*, \mathbf{X})]^T$ is a matrix in $\mathbb{R}^{N \times N_*}$ such that its element at position (i, j) is $k(\mathbf{x}_i, \mathbf{x}_{*j})$, and $k(\mathbf{X}_*, \mathbf{X}_*)$ is a matrix in $\mathbb{R}^{N_* \times N_*}$ such that its element at position (i, j) is $k(\mathbf{x}_{*i}, \mathbf{x}_{*j})$.

Using a property of joint Gaussian distributions (see, e.g., [130, Eq. (A.6)]), we are able to compute the distribution conditioned on $\mathbf{y}$:

$$\mathbf{f}_* | \mathbf{y}, \mathbf{X}, \mathbf{X}_* \sim \mathcal{N}(\mu(\mathbf{X}_*), \Sigma(\mathbf{X}_*, \mathbf{X}_*)), \quad (3.11)$$

where

$$\mu(\mathbf{X}_*) = m(\mathbf{X}_*) + k(\mathbf{X}_*, \mathbf{X}) \mathbf{K}_y^{-1} (\mathbf{y} - m(\mathbf{X})), \quad (3.12)$$

$$\Sigma(\mathbf{X}_*, \mathbf{X}_*) = k(\mathbf{X}_*, \mathbf{X}_*) - k(\mathbf{X}_*, \mathbf{X}) \mathbf{K}_y^{-1} k(\mathbf{X}, \mathbf{X}_*). \quad (3.13)$$

We can use $\mu(\mathbf{X}_*)$ as an estimate for $\mathbf{f}_*$ and $\Sigma(\mathbf{X}_*, \mathbf{X}_*)$ as a measure of the uncertainty in this estimation. Notice we can understand GPR as a Bayesian inference method in which we start with a prior $\mathcal{GP}(m, k)$ for f and, after observing the training data, we find a posterior distribution: $f | \mathbf{X}, \mathbf{y} \sim \mathcal{GP}(\mu, \Sigma)$.

The kernel k and the mean m are chosen so as to encode prior beliefs on the latent function f (such as smoothness, periodicity, etc) and are often parameterized by the so-called hyper-parameters $\boldsymbol{\theta}$. The requirements on k are that it is symmetric ($k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$) and that it is positive semidefinite in the sense that all the eigenvalues of $k(\mathbf{X}, \mathbf{X})$ are non-negative for all $\mathbf{X}$ for any choice of N (see [130, Chapter 4]). A fact that will be important in this chapter is that the sum of valid kernels is a valid kernel.

A common choice of kernel is the squared exponential kernel with automatic relevance determination parameters, which is given by

$$k(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp \left(-\frac{1}{2} (\mathbf{x} - \mathbf{x}')^T \mathbf{M}_L^{-1} (\mathbf{x} - \mathbf{x}') \right), \quad (3.14)$$

with $\mathbf{M}_L = \text{diag}\{l_1^2, \dots, l_{n_x}^2\}$ (recall n_x is the dimension of the inputs). In this case, the hyper-parameters are the real numbers $l_1, \dots, l_{n_x}$, and σ_f . The standard deviation of the noise, σ_ϵ , is usually not known in advance and can be considered as a hyper-parameter. A popular way to set the hyper-parameters is to maximize the log marginal likelihood

$$\ln(p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta})) = -\frac{1}{2} \mathbf{y}^T \mathbf{K}_y^{-1} \mathbf{y} - \frac{1}{2} \ln(\det(\mathbf{K}_y)) - \frac{N}{2} \ln(2\pi). \quad (3.15)$$

For more on GPR, refer to [116, 130] and references therein.

3.2.2 Clustering

For REWS estimation, the first step of the training process is to cluster the data. Indeed, since we expect the turbine to have different behaviors under different operating conditions, it seems reasonable to look for different estimations depending on the operating range.

Remember each input point $\mathbf{x}_i$ in our training set is a vector with dimension $3(r+1)$ containing measurements of the rotor speed, the pitch angle, and the active power. Since these measurements have different scales, we first standardize the training data. In other words, we compute the mean and the standard deviation of each component of the input points across the training set, and then from each component, we subtract the corresponding mean and then divide the result by the corresponding standard deviation.

The second step is to use Principal Component Analysis (PCA) [131] to make a change of coordinates. Note that PCA is not used for dimensional reduction here, but only to design the clustering.

Finally, we use the k-means++ algorithm [7] to split the data into $n_c \in \mathbb{N}$ clusters.

3.2.3 Finding an estimate per cluster

Kernel and mean function

To use GPR, we need to define a kernel k and mean function m that encode our prior beliefs about the relation between the inputs and the outputs. Here, we use a kernel of the form

$$k(\mathbf{x}_a, \mathbf{x}_b) = k_\omega(\boldsymbol{\omega}_{a-r:a}, \boldsymbol{\omega}_{b-r:b}) + k_\beta(\boldsymbol{\beta}_{a-r:a}, \boldsymbol{\beta}_{b-r:b}) + k_P(\mathbf{P}_{a-r:a}, \mathbf{P}_{b-r:b}), \quad (3.16)$$

where a and b are (possibly different) time instants and

$$k_\omega(\mathbf{z}_a, \mathbf{z}_b) = \mathbf{z}_a^T \mathbf{L}_1^{-1} \mathbf{z}_b + \sigma_1^2 \exp \left(-\frac{1}{2} (\mathbf{z}_a - \mathbf{z}_b)^T \mathbf{L}_2^{-1} (\mathbf{z}_a - \mathbf{z}_b) \right), \quad (3.17)$$

$$k_\beta(\mathbf{z}_a, \mathbf{z}_b) = \sigma_2^2 \exp \left(-\frac{1}{2} (\mathbf{z}_a - \mathbf{z}_b)^T \mathbf{L}_3^{-1} (\mathbf{z}_a - \mathbf{z}_b) \right), \quad (3.18)$$

$$k_P(\mathbf{z}_a, \mathbf{z}_b) = \mathbf{z}_a^T \mathbf{L}_4^{-1} \mathbf{z}_b + \sigma_3^2 \exp \left(-\frac{1}{2} (\mathbf{z}_a - \mathbf{z}_b)^T \mathbf{L}_5^{-1} (\mathbf{z}_a - \mathbf{z}_b) \right), \quad (3.19)$$

with

$$\mathbf{L}_i = \text{diag} \{ ((r+1)l_i)^2, (rl_i)^2, \dots, (2l_i)^2, l_i^2 \}, \text{ for } i = 1, \dots, 5. \quad (3.20)$$

Hence, the hyper-parameters are $l_1, \dots, l_5, \sigma_1, \sigma_2$, and σ_3 . The ideas behind the use of this form for the kernel are:

1. the linear contributions (those of the form $\mathbf{z}_a^T \mathbf{L}_i^{-1} \mathbf{z}_b$) account for the fact that we expect the wind speed to be higher when the turbine is producing more power and the rotor speed is faster;
2. the nonlinear contributions are in the form of a squared exponential kernel, which is a more generic form and should account for the more intricate dynamics of the turbine;
3. the matrices $\mathbf{L}_i$ are as presented in (3.20) to yield smaller weights for the input components containing information further in the past while maintaining a small number of hyper-parameters to be selected.

For the mean function, we use a constant function equal to the mean of the wind speed for the training points in the cluster. Also, since we consider the reference wind speed to be the ground truth, we set the standard deviation of the noise σ_ϵ to zero.

Training and computing estimates

We select $N_{GP,i}$ pairs $(\mathbf{x}_i, y_i)$ from the i th cluster to use in the following computations. Here we use the index i instead of t because these training pairs are randomly chosen from the available data, so that, e.g., y_2 does not necessarily correspond to wind speed measurement obtained in the sample time right after the one corresponding to y_1 . However, the measurements that constitute a particular $\mathbf{x}_i$ are ordered. Thus, here, and in the sequel, one should understand that the pair $(\mathbf{x}_i, y_i)$ corresponds not to time $t = i$, but to some time t_i .

After selecting the training pairs, the next step is to tune the hyper-parameters by maximizing the marginal likelihood (3.15). Notice that, for different values of the time-window r , the training pairs are different and

so might be the hyper-parameters. Hence, we can effectively have several different models for the relation between the SCADA measurements and the wind speed. However, once the training data is prepared and the hyper-parameters are set, the procedure to obtain the estimates is the same regardless of the choice of r .

Recall that, for a test point $\mathbf{x}_*$, we would normally find the estimate

$$\hat{y}_* = m(\mathbf{x}_*) + k(\mathbf{x}_*, \mathbf{X}) \mathbf{K}_y^{-1} (\mathbf{y} - m(\mathbf{X})), \quad (3.21)$$

where $\mathbf{X}$ is the matrix containing all training points (of the relevant cluster) and $\mathbf{K}_y = k(\mathbf{X}, \mathbf{X}) + \sigma_\epsilon^2 \mathbf{I}$. However, real-time computation of $\hat{y}_*$ using (3.21) leads to a numerical burden since the time to compute $k(\mathbf{x}_*, \mathbf{X})$ scales with the size of the training set. Thus, it is desired to have a large training set to learn the map well from SCADA measurements to wind speed, but a small one to perform estimations in real-time. In order to solve this dilemma, we follow the method proposed by [84], which consists in using the following approximation:

$$\hat{y}_* \approx m(\mathbf{x}_*) + k(\mathbf{x}_*, \bar{\mathbf{X}}) \bar{\boldsymbol{\alpha}}, \quad (3.22)$$

where $\bar{\boldsymbol{\alpha}}$ is found as the solution of a regularized Least Squares (LS) problem,

$$\bar{\boldsymbol{\alpha}} = \arg \min_{\boldsymbol{\alpha} \in \mathbb{R}^M} \|(\mathbf{y} - m(\mathbf{X})) - \bar{\mathbf{K}}_y \boldsymbol{\alpha}\|^2 + \mu \|\boldsymbol{\alpha}\|^2 \quad (3.23)$$

$$= (\bar{\mathbf{K}}_y^T \bar{\mathbf{K}}_y + \mu \mathbf{I})^{-1} \bar{\mathbf{K}}_y^T (\mathbf{y} - m(\mathbf{X})), \quad (3.24)$$

$\bar{\mathbf{X}}$ is obtained by deleting a number of rows from $\mathbf{X}$, $\bar{\mathbf{K}}_y$ is obtained by deleting the corresponding columns of $\mathbf{K}_y$, $\mu \geq 0$ is a regularization coefficient, and M is the number of rows of $\bar{\mathbf{X}}$. We refer to this method as GPR+LS in the sequel.

Notice $\bar{\boldsymbol{\alpha}}$ depends only on the training set, hence, we can store its value and then compute $k(\mathbf{x}_*, \bar{\mathbf{X}})$ online for each new test point. Therefore, once $\bar{\boldsymbol{\alpha}}$ is known, the time to compute $\hat{y}_*$ using (3.22) is $\mathcal{O}(M)$. Furthermore, we found in our numerical experiments that the matrix $\mathbf{K}_y$ is often poorly conditioned, which leads to numerical problems in computing the estimates using the exact GPR method.

Fig. 3.1 depicts a flowchart of the algorithm for one cluster. Notice once $\bar{\mathbf{X}}$ and $\bar{\boldsymbol{\alpha}}$ are known, we can input each test point $\mathbf{x}_*$ to the model displayed on the right-hand side of Fig. 3.1 and obtain an output $\hat{y}_*$. There is a computation of $m(\mathbf{x}_*)$ for the figure to be more general, but, in our case, $m(\mathbf{x}_*)$ is the mean wind speed for the cluster, so it is fixed.

3.2.4 Finding the final estimate

In the previous subsection, we performed regression using each cluster separately. Thus, we can find n_c output estimates $(\hat{y}_{1,*}, \dots, \hat{y}_{n_c,*})$ for the same

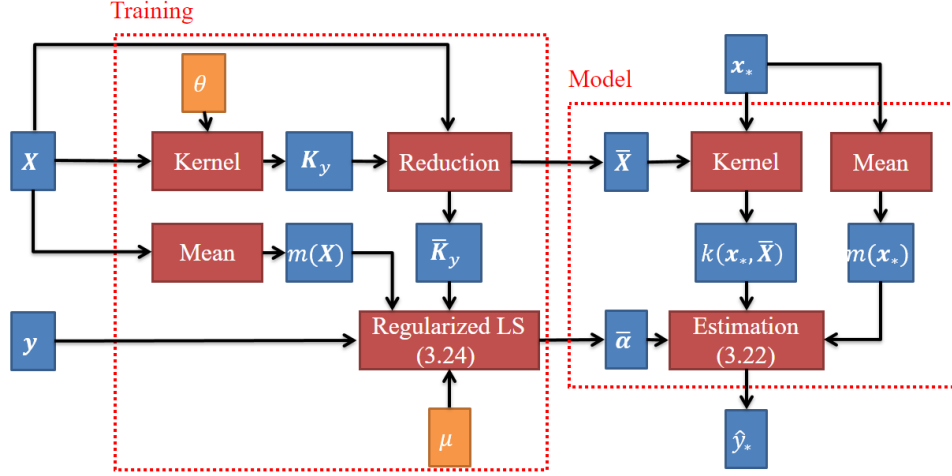


Figure 3.1: Schematic of the algorithm for one cluster.

test point $\mathbf{x}_*$, where n_c is the number of clusters. In this subsection, we discuss how to compute a final value for the estimate.

First, we apply to $\mathbf{x}_*$ the same transformations we applied to the training set in Subsection 3.2.3 (standardization with the training set statistics and PCA) to obtain $\mathbf{x}'_*$. Then, we propose two approaches to find the final estimate:

1. to use only the estimate associated with the cluster whose centroid is closest to $\mathbf{x}'_*$;
2. to compute a weighted average of all the estimates

$$\hat{y}_{\text{final},*} = \frac{\sum_{i=1}^{n_c} w_i \hat{y}_{i,*}}{\sum_{i=1}^{n_c} w_i}, \quad (3.25)$$

where the weights $w_1, \dots, w_{n_c}$ are computed as

$$w_i = \frac{1}{\|\mathbf{x}'_* - \mathbf{c}_i\|^n}, \quad (3.26)$$

with $n > 0$ and $\mathbf{c}_i$ the centroid of the i th cluster.

Notice that, when using only the nearest cluster, i.e., the first option, one can reduce the computational cost of computing the estimate by first verifying which is the nearest cluster and then computing the estimation only with the model which is associated with it (since the remaining $\hat{y}_{i,*}$'s are not necessary for the final computation in this case).

3.3 Numerical results for the REWS estimation with GPR

Recall that the data used for training and testing the previously presented method are field data provided by Engie Green and Leosphere within the framework of the project “SmartEole” of the French National Research Agency (ANR). They correspond to a 3-bladed horizontal axis wind turbine model Senvion MM82 (82 m rotor diameter and 2.05 MW rated power) located in the Ablaincourt-Pressoir wind farm, in France.

3.3.1 Performance measurement

To assess the performance of the proposed methods, we compute the following index (relative error)

$$e_* = \left| \frac{\hat{y} - y_*}{y_*} \right| \times 100\%, \quad (3.27)$$

where y_* is a reference value and $\hat{y}$ is the corresponding estimate. For each set of tests, we use boxplots of this index to depict its distribution.

3.3.2 Results

We start with a large set $\mathcal{S}_{\text{train}}$ of training points, split it into n_c clusters, and compute their centroids $\mathbf{c}_1, \dots, \mathbf{c}_{n_c}$. Then, for each cluster, we randomly choose N points to build the matrix $\mathbf{X}$ and a subset of these with $M < N$ points to build the matrix $\tilde{\mathbf{X}}$ to perform the regression. Finally, we select test inputs (SCADA measurements) from another set, $\mathcal{S}_{\text{test}}$, and estimate the corresponding outputs (i.e., the wind speed values). The sets $\mathcal{S}_{\text{train}}$ and $\mathcal{S}_{\text{test}}$ are disjoint. In this section, we present the results for different choices of n_c , M , and N . We keep $r = 20$ fixed.

Simulation of a real-time application

To simulate a real-time application, we took consecutive points from a randomly chosen day and estimated the corresponding wind speed values.

Figs. 3.2, 3.3, and 3.4 show, respectively, the rotor speed, the pitch angle, and the produced power time-series corresponding to the selected day. These are used as inputs to the estimation procedure. The points at time t are color coded according to the cluster used for the estimation of the wind speed at the corresponding time, i.e., the one whose centroid is nearest to the input point at that time. We used 5 clusters in this case, but this particular day happens not to have any input point associated with cluster 2.

The estimation results are depicted in Figs. 3.5a and 3.5b. In this case, the number of clusters is $n_c = 5$, the time window size is $r = 20$, the

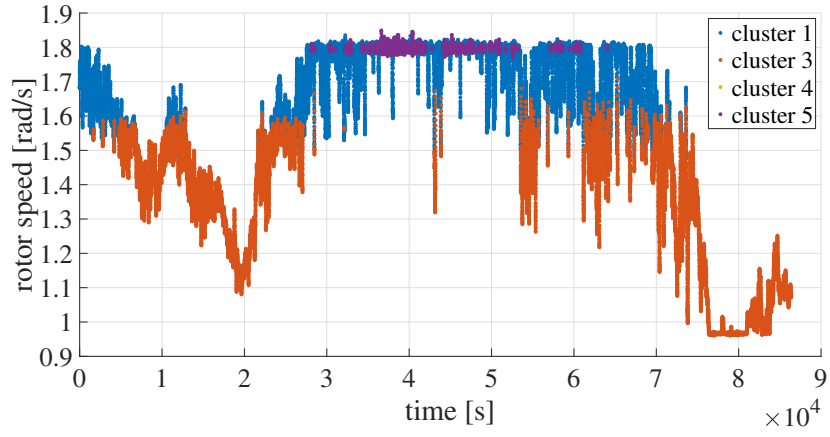


Figure 3.2: Rotor speed time-series corresponding to the selected day. The points are color-coded according to the cluster used for the estimation of the wind speed at the corresponding time.

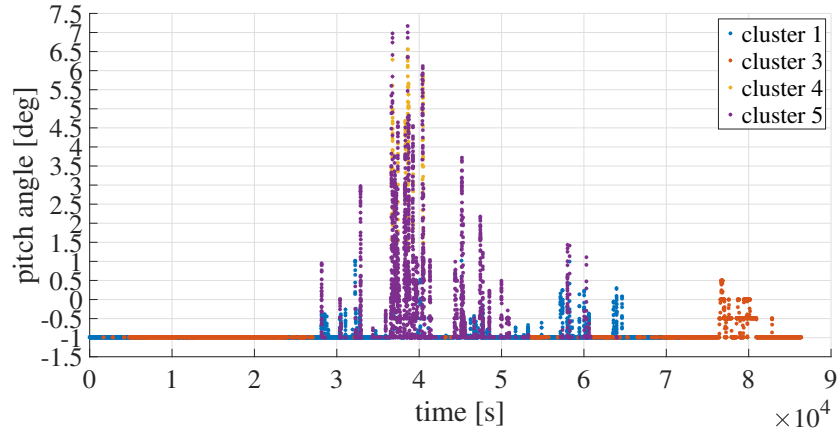


Figure 3.3: Pitch angle time-series corresponding to the selected day. The points are color-coded according to the cluster used for the estimation of the wind speed at the corresponding time.

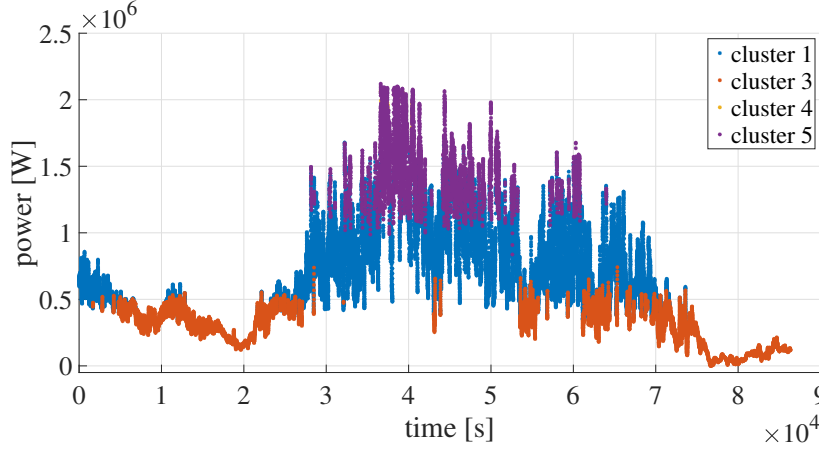


Figure 3.4: Produced power time-series corresponding to the selected day. The points are color-coded according to the cluster used for the estimation of the wind speed at the corresponding time.

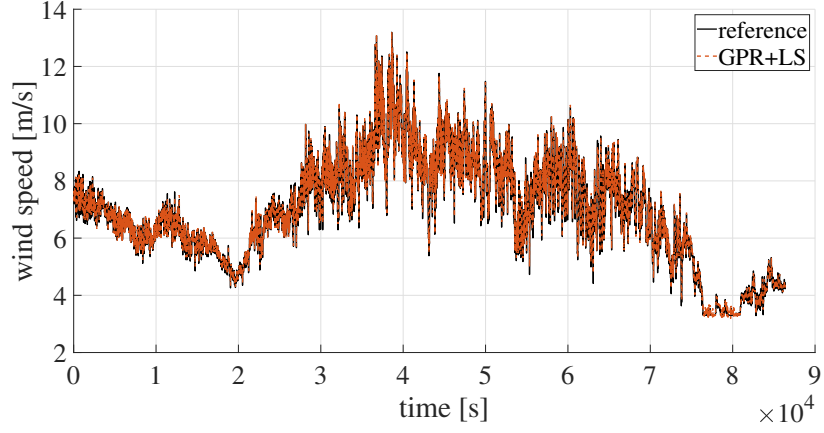
number of points per cluster (used offline) is $N = 10^4$, and the number of points per cluster used in the online computations is $M = 3 \times 10^3$. We use only the nearest cluster estimate. We observe the result, in this case, is highly satisfactory for control applications. Notice the selected day covers a relatively large range of wind speeds, from cut-in to above-rated, but very high speeds are absent. It has been verified on other days that the estimation behaves similarly well in this range. Clustering is required to obtain good behavior both in the cut-in speed and the above-rated regions.

Also, in Fig 3.5b, we included the results obtained with a more traditional model-based approach using a Kalman Filter (KF). This is similar to, e.g., what [94] used to estimate the wind speed. The error comparison is presented formally in Fig. 3.6.

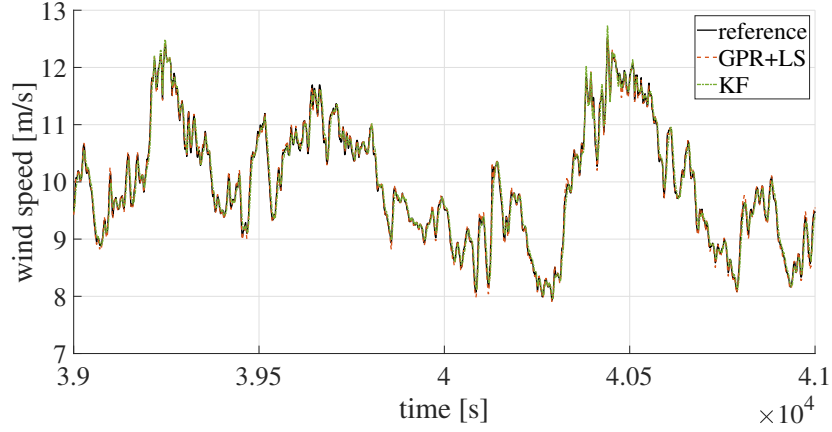
Notice our proposed approach has comparable performance to that of the KF approach and, most importantly, requires no access to the turbine model. Furthermore, the KF requires manual tuning of the parameters. In this case, we carefully tuned the KF using part of our data before testing it with the data of the day presented in Figs. 3.5a and 3.5b.

Performance assessment on data from several days

Here, we show the effects of varying the parameters n_c (number of clusters), M (number of points per cluster used in the online computation), and N (numbers of points per cluster) on the performance of our method. We do it by inputting test points collected from several different days of operation of the turbine to estimate the corresponding wind speed and then plotting the corresponding boxplots to show how the relative error distribution changes



(a) GPR+LS results for the entire day.



(b) Zoomed-in view with comparison to a Kalman Filter (KF) approach.

Figure 3.5: Estimation for consecutive points to simulate a real-time application. The parameters are $n_c = 5$, $r = 20$, $N = 10^4$, and $M = 3 \times 10^3$. We use only the nearest cluster estimate.

according to the changes in these parameters.

First, we chose $N = 10^4$, $M = 3000$, and $n_c = 5$ to present the effect of combining the estimates from each cluster using different weight functions. The best mean error was obtained using only the nearest cluster estimate. This is advantageous since the estimate can be computed much faster, as commented in Subsection 3.2.4. Considering this result, we use only the nearest cluster estimate for the other tests presented here.

In Fig. 3.7, we show the effect of varying the number of clusters n_c . The median error steadily decreases with the number of clusters. From $n_c = 5$, the mean error and the 25%-quantile error are low and steadily decreasing. Only the 5%-quantile error is still slightly irregular. Note that we need

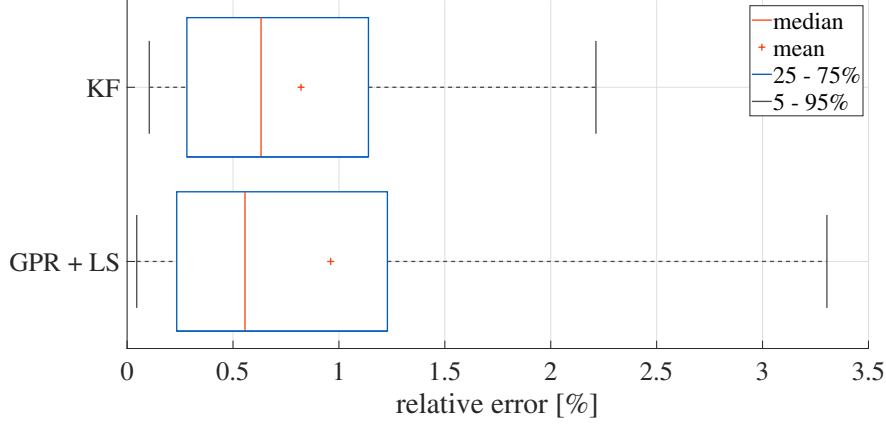


Figure 3.6: Boxplots of the estimation error for the day shown in Figs. 3.5a and 3.5b.

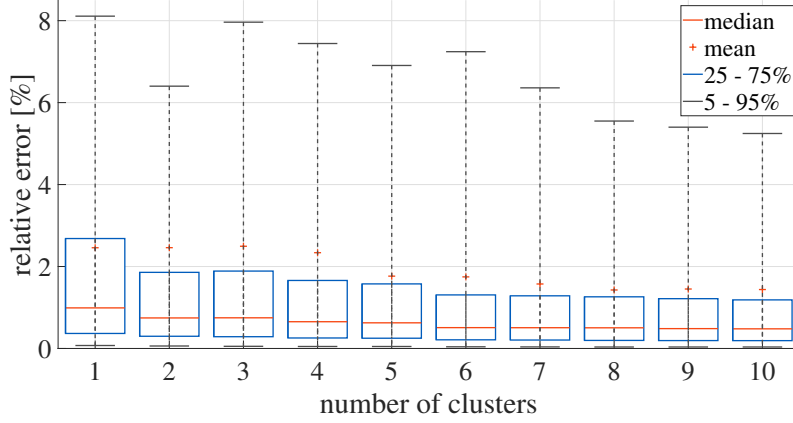


Figure 3.7: Relative error for different choices of the number of clusters. In this case $r = 20$, $N = 10^4$, $M = 3000$, and we only use the nearest cluster estimate.

not select a low value of n_c , since it does not impact the cost of online computations (when using only the nearest cluster estimate). Finally, for the results presented in this chapter, we pick up $n_c = 5$.

Also, using a very low number of clusters such as $n_c = 1$ would lead to unsatisfactory results in the vicinity of the turbine cut-in speed and/or above-rated speeds.

Fig. 3.8 depicts a comparison among the results using different values of M . We see the trade-off related to using an approximation instead of the exact GPR (in which case we would have $M = N$). We see a downward trend in the median of the error as we increase the number of points, as

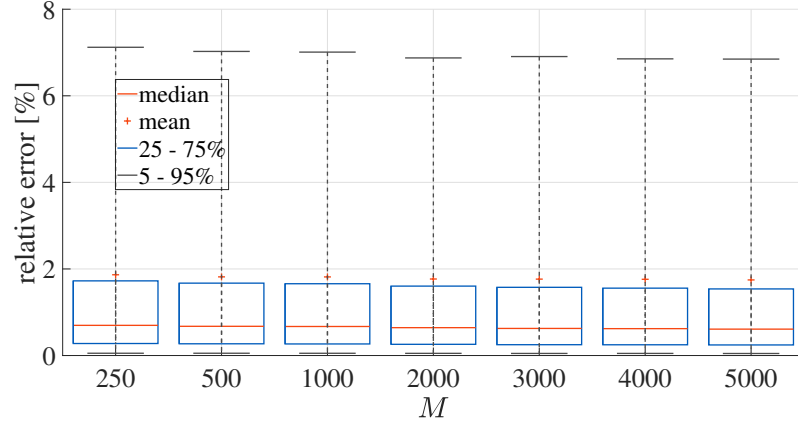


Figure 3.8: Relative error for different values of M . The remaining parameters are $n_c = 5$, $r = 20$, and $N = 10^4$. We use only the nearest cluster estimate.

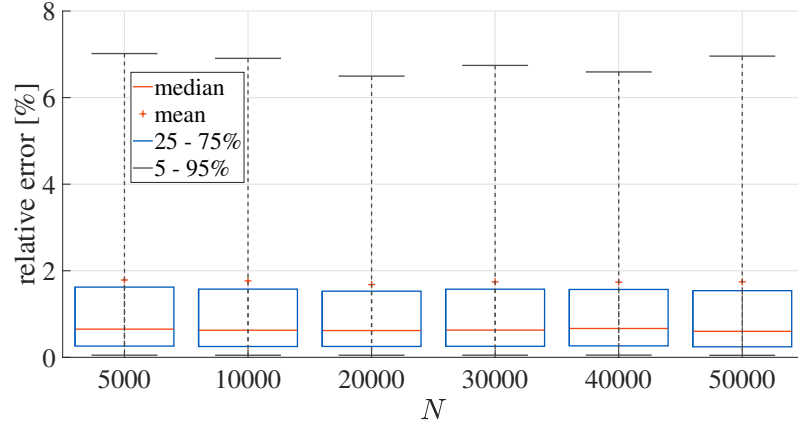


Figure 3.9: Relative error for different values of N . The remaining parameters are $n_c = 5$, $r = 20$, and $M = 3000$. We use only the nearest cluster estimate.

expected.

Finally, we show the effect of varying N in Fig. 3.9. N is related to the offline computational cost, so it needs not to be selected very carefully and can be much larger than M . Here, we work with $N = 10^4$. Again, we see an overall downward trend in the median of the error as the number of points increases.

3.4 Estimating the wind speed with a mixed data/model-based approach

In this section, we present an alternative estimation methodology which, in lieu of directly estimating the wind speed with a GPR as previously, proposes a mixed data/model-based approach to estimate the REWS. Namely, we propose here to use GPR to estimate the aerodynamic torque of the turbine and then use this estimated torque in an observer to estimate the wind speed.

By partially grounding on the physical model of the system, this approach aims at providing some indication on the accuracy of the GPR and thus potential need for re-training. Indeed, as the turbine ages, its characteristics might change and the map learned by GPR may become less accurate and less useful for estimation. This cannot be accounted for by the previous method, while the one presented in this section could, by comparing the measured rotor speed with an estimated one obtained from the estimated open-loop dynamical model.

As previously, the method proposed in this section does not require knowledge of turbine look-up tables of the torque or power coefficients, which are learned, and can be applied without the permanent use of a LiDAR sensor. It consists in two steps which are now detailed: (i) learning of the aerodynamic torque with a GPR; (ii) design of a wind speed observer with a high-gain observer, for which a bound on the asymptotic observation error is provided, involving the aerodynamic torque learning error.

Once again, the training is done using SCADA data from the turbine and REWS data obtained with a nacelle-mounted LiDAR sensor.

3.4.1 Estimating the aerodynamic torque with GPR

As mentioned in the Introduction (see Section 2.2), neglecting losses due to, e.g., friction, a simple dynamic model for the torque is

$$J\dot{\omega}(t) = T_a(\omega(t), \beta(t), v(t)) - T_g(t), \quad (3.28)$$

where J is the moment of inertia of the rotor, ω is the rotor angular speed, v is the REWS, T_a is the aerodynamic torque, β is the pitch angle (which is assumed to be the same for all blades, as previously), and T_g is the generator torque. The quantities ω , β , and T_g are measured and the inertia J is known.

In view of this dynamics, given discrete-time trajectories of rotor speed, pitch angle, wind speed, and generated torque, sampled at a rate $\Delta t > 0$, we propose to train a GPR model to estimate the mapping T_a . Namely, we perform a nonlinear regression to estimate the mapping $\mu : (\omega, \beta, v, T_g) \mapsto \omega(t + \Delta t)$ and, using a Euler differentiation of (3.28), we form an estimate

of the aerodynamic torque as

$$\hat{T}_a(\omega, \beta, v, T_g) = J \frac{\mu(\omega, \beta, v, T_g) - \omega}{\Delta t} + T_g \quad (3.29)$$

The protocol for the training is similar to the one presented in the previous section, namely the data were standardized, we used PCA to make a change of coordinates, and then split the data into clusters using the k-means++ algorithm. We trained a model for each cluster and when testing the system we used the model of the appropriate cluster for the particular input received to estimate the torque. We use a squared exponential covariance function and a zero mean function.

Notice that, in this case, we use regular GPR and not the least squares approximation we used in the previous method.

3.4.2 High-gain observer to estimate the REWS

As a second step in our methodology, we propose to estimate the REWS based on the learned aerodynamic torque with a high-gain observer. A myriad of observation techniques were proposed for this estimation problems over the years (see Subsection 2.3.1) when the model is assumed to be perfectly known, but we were more specifically inspired by [92], which proposes a control-based high-gain design, to choose this high-gain approach.

This subsection is divided into three parts. In the first one, we briefly present the concept of high-gain observer and we prove a new convergence result that will be useful later. In the second one, we show how we apply the high-gain observer together with GPR torque estimation to estimate the wind speed. Finally, the third part discusses the convergence of the observer for the application at hand.

High-gain observers

Generally speaking, high-gain observers are used for systems that can be written in a lower triangular form where (in continuous time) the derivative of the k th state variable of the system only depends on the first $k + 1$ state variables (and possibly on exogenous variables and on the systems output, but not on the remaining state variables). The idea is to choose a large enough scalar gain λ such that a correction term associated with λ^k is added to the dynamic equation of the k th state variable.

More concretely, following the design of [58], consider a system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, y, t) + \boldsymbol{\eta}(t), \quad (3.30)$$

$$\mathbf{y} = h(\mathbf{x}, t), \quad (3.31)$$

where $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbb{R}^n$ is the state vector, with $n \in \mathbb{N}$, $y \in \mathbb{R}$ is the measurable output, $t \in \mathbb{R}$ is time, $\mathbf{f}$ and h are sufficiently smooth functions,

and $\boldsymbol{\eta} \in \mathbb{R}^n$ is a vector of unknown disturbances, which can also account for modeling errors (we assume these disturbances are bounded functions). We do not include explicit dependencies on any known exogenous signal to simplify the presentation. Now, consider this system satisfies the following assumptions

$$\frac{\partial f_i(\mathbf{x}, y, t)}{\partial x_{i+1}} \neq 0 \text{ and } \frac{\partial f_i(\mathbf{x}, y, t)}{\partial x_j} = 0 \quad \forall i \in \{1, \dots, n-1\}, j > i+1, \quad (3.32)$$

$$\frac{\partial h(\mathbf{x}, t)}{\partial x_1} \neq 0 \text{ and } \frac{\partial h(\mathbf{x}, t)}{\partial x_j} = 0 \quad \forall j > 1, \quad (3.33)$$

for all $(\mathbf{x}, y, t) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}$, where f_i is the i th component of $\mathbf{f}$ (due to the second condition, we write $h(x_1, t)$ instead of $h(\mathbf{x}, t)$ in the sequel).

A system described by (3.30)–(3.31) satisfying (3.32) and (3.33) is said to be in Gauthier-Kupca’s uniform observability canonical form [58]. It is important to stress we need the dynamical system of interest to be described in this form for a high-gain observer to be applied to it.¹

The idea of high-gain observers is to use the following equation to recover an estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$:

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, y, t) + \text{diag}\{\lambda, \lambda^2, \dots, \lambda^n\} \mathbf{K}(y - h(\hat{x}_1, t)), \quad (3.34)$$

where $\lambda \in \mathbb{R}$ is the so called high-gain parameter and $\mathbf{K} \in \mathbb{R}^{n \times 1}$ is a constant matrix, both to be designed. Under certain assumptions, one can show that an appropriate $\mathbf{K}$ exists such that, with this $\mathbf{K}$, $\hat{\mathbf{x}}$ converges to $\mathbf{x}$ if λ is sufficiently large,² see, e.g., [16, Theorem 6.1]. For more in-depth discussions on the high-gain observer theory refer to, e.g., [69, 70, 71] and to the recent literature review [16, Section 6.2].

For our specific wind estimation problem, we aim at applying a high-gain observer for a dynamics which exhibits model uncertainties and is only in Gauthier-Kupca’s uniform observability canonical form (and not in the strict-feedback form). Up to our knowledge, convergence results available in the literature consider each issue separately (namely, [58, Theorem 7.1] does not consider a system affected by disturbances, while [17] does but requires f_i to be linear in x_{i+1}). This is why, grounding on the aforementioned references, we will introduce a new result tailored to our needs and that we use later in this section.

We start with the following assumption, then, we finish this part by stating and proving Theorem 3.1.

¹More generally, one can consider a system that can be transformed to this form through some change of variables $\mathbf{z} = \mathbf{T}(\mathbf{x}, t)$. Under appropriate conditions, one can recover an estimate for $\mathbf{x}$ from an estimate for $\mathbf{z}$ by inverting $\mathbf{T}$. However, this might lead to loss of some convergence properties. We do not delve into this details here as we do not need a change of variables for the application at hand.

²Or that the estimation error is ultimately bounded when accounting for the disturbances $\boldsymbol{\eta}$.

Assumption 3.1. *There exist positive scalars a_m , a_M , b_M , c_m , and c_M such that, for all $(\mathbf{x}, y, t) \in \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}$,*

$$a_m < \left| \frac{\partial f_i(\mathbf{x}, y, t)}{\partial x_{i+1}} \right| < a_M, 1 \leq i \leq n-1, \quad (3.35)$$

$$\left| \frac{\partial f_i(\mathbf{x}, y, t)}{\partial x_j} \right| < b_M, 1 \leq i \leq n, 1 \leq j \leq i, \quad (3.36)$$

$$c_m < \left| \frac{\partial h(x_1, t)}{\partial x_1} \right| < c_M. \quad (3.37)$$

Theorem 3.1. *Consider the dynamical system (3.30)–(3.31) satisfying (3.32)–(3.33) and Assumption 3.1. There exist $k_1, \dots, k_n \in \mathbb{R}$ and $\lambda^*, \alpha, \beta, \gamma > 0$ such that, if $\mathbf{K} = [k_1, \dots, k_n]^T$ and $\lambda > \lambda^*$, then the observer (3.34) satisfies*

$$\begin{aligned} |\hat{x}_i(t) - x_i(t)| &\leq \beta \lambda^{i-1} \exp(-\alpha \lambda(t - t_0)) \|\hat{x}(t_0) - x(t_0)\| \\ &\quad + \gamma \sup_{1 \leq j \leq n, s \in [t_0, t]} \lambda^{i-j-1} \eta_j^2(s) \end{aligned} \quad (3.38)$$

for all $i \in \{1, \dots, n\}$ and $t \geq t_0$.

Proof. This proof follows very closely the structure of the proof presented in [58, Section 7.3]. Define the error variables $\tilde{x}_1 = x_1 - \hat{x}_1, \dots, \tilde{x}_n = x_n - \hat{x}_n$. We have

$$\dot{\tilde{x}}_i = f_i(x_1, \dots, x_{i+1}, t) - f_i(\hat{x}_1, \dots, \hat{x}_{i+1}, t) - \lambda^i k_i (y - h(\hat{x}_1, t)) + \eta_i(t) \quad (3.39)$$

$$\begin{aligned} &= f_i(x_1, \dots, x_i, x_{i+1}, t) - f_i(x_1, \dots, x_i, \hat{x}_{i+1}, t) - \lambda^i k_i (y - h(\hat{x}_1, t)) \\ &\quad + f_i(x_1, \dots, x_i, \hat{x}_{i+1}, t) - f_i(\hat{x}_1, \dots, \hat{x}_{i+1}, t) + \eta_i(t) \end{aligned} \quad (3.40)$$

$$\begin{aligned} &= \frac{\partial f_i}{\partial x_{i+1}}(x_1, \dots, x_i, \theta_i(t), t) \tilde{x}_{i+1}(t) - \lambda^i k_i \frac{\partial h}{\partial x_1}(\theta_0(t), t) \tilde{x}_1(t) \\ &\quad + f_i(x_1, \dots, x_i, \hat{x}_{i+1}, t) - f_i(\hat{x}_1, \dots, \hat{x}_i(t), \hat{x}_{i+1}, t) + \eta_i(t), \end{aligned} \quad (3.41)$$

where $\theta_i(t) \in [\min\{x_{i+1}(t), \hat{x}_{i+1}(t)\}, \max\{x_{i+1}(t), \hat{x}_{i+1}(t)\}]$ and the last equality follows from the intermediate value theorem. Notice that we write $f_i(x_1, \dots, x_i, t)$ instead of $f_i(\mathbf{x}, y, t)$, i.e., we do not write the dependency on y explicitly to simplify the presentation.

Now consider the re-scaled variables

$$e_i = \lambda^{n-i} \tilde{x}_i, 1 \leq i \leq n. \quad (3.42)$$

These satisfy

$$\dot{e}_i = \lambda \frac{\partial f_i}{\partial x_{i+1}}(x_1, \dots, x_i(t), \theta_i(t), t) e_i(t) - \lambda k_i \frac{\partial h}{\partial x_1}(\theta_0(t), t) \tilde{x}_1(t) + \lambda^{n-i} F_i(t) \quad (3.43)$$

for $1 \leq i \leq n-1$ and

$$\dot{e}_n = -\lambda k_n \frac{\partial h}{\partial x_1}(\theta_0(t), t) \tilde{x}_1(t) + F_n(t), \quad (3.44)$$

with

$$F_i(t) = f_i(x_1, \dots, x_i, \hat{x}_{i+1}, t) - f_i(\hat{x}_1, \dots, \hat{x}_i(t), \hat{x}_{i+1}, t) + \eta_i(t), \quad 1 \leq i < n, \quad (3.45)$$

$$F_n(t) = f_n(x_1, \dots, x_n, t) - f_n(\hat{x}_1, \dots, \hat{x}_n(t), t) + \eta_n(t). \quad (3.46)$$

Thus, we have, in matrix notation,

$$\dot{\mathbf{e}} = \lambda \mathbf{A}(t) \mathbf{e}(t) + \mathbf{F}_\lambda, \quad (3.47)$$

with

$$\mathbf{A}(t) = \begin{bmatrix} -k_1 \frac{\partial h}{\partial x_1}(\theta_0(t), t) & a_1(t) & 0 & \dots & 0 \\ -k_2 \frac{\partial h}{\partial x_1}(\theta_0(t), t) & 0 & a_2(t) & & 0 \\ \vdots & & & \ddots & \\ -k_{n-1} \frac{\partial h}{\partial x_1}(\theta_0(t), t) & 0 & 0 & & a_{n-1}(t) \\ -k_n \frac{\partial h}{\partial x_1}(\theta_0(t), t) & 0 & 0 & \dots & 0 \end{bmatrix}, \quad (3.48)$$

$$a_i(t) = \frac{\partial f_i}{\partial x_{i+1}}(x_1, \dots, x_i, \theta_i(t), t), \quad (3.49)$$

and

$$\mathbf{F}_\lambda = [\lambda^{n-1} F_1(t), \lambda^{n-2} F_2(t), \dots, \lambda F_{n-1}(t), F_n(t)]^T. \quad (3.50)$$

The components of $\mathbf{A}$ satisfy Assumption 3.1, then, according to [39, Lemma 2.1 in Chapter 6], it holds that there are a symmetric positive definite matrix $\mathbf{S}$ and positive scalar q such that

$$\mathbf{S} \mathbf{A}(t) + \mathbf{A}^T(t) \mathbf{S} \leq -q \mathbf{I}. \quad (3.51)$$

Let us define the Lyapunov function candidate $\mathcal{V} = \mathbf{e}^T \mathbf{S} \mathbf{e}$. From the previous considerations, it satisfies

$$\dot{\mathcal{V}} = \mathbf{e}^T \lambda (\mathbf{S} \mathbf{A}(t) + \mathbf{A}^T(t) \mathbf{S}) \mathbf{e} + 2 \mathbf{e}^T \mathbf{S} \mathbf{F}_\lambda \quad (3.52)$$

$$\leq -\lambda q \|\mathbf{e}\|^2 + 2 \mathbf{e}^T \mathbf{S} \mathbf{F}_\lambda. \quad (3.53)$$

Using Assumption 3.1 and the mean-value inequality, we have

$$|\lambda^{n-i} F_i(t)| \leq b_M \lambda^{n-i} \sum_{j=1}^i |\tilde{x}_j(t)| + \lambda^{n-i} |\eta_i(t)| \quad (3.54)$$

$$\leq b_M \sum_{j=1}^i \lambda^{j-i} |e_j(t)| + \lambda^{n-i} |\eta_i(t)| \quad (3.55)$$

$$\leq b_M \sum_{j=1}^i |e_j(t)| + \lambda^{n-i} |\eta_i(t)| \quad (3.56)$$

for $1 \leq i \leq n$ if $\lambda > 1$. Hence,

$$\|\mathbf{F}_\lambda\| \leq n\sqrt{n}b_M\|\mathbf{e}(t)\| + \sqrt{n} \sup_{i=1,\dots,n} \lambda^{n-i}|\eta_i(t)| \quad (3.57)$$

and, thus,

$$2\mathbf{e}^T \mathbf{S} \mathbf{F}_\lambda \leq 2\|\mathbf{e}\| \|\mathbf{S}\| \|\mathbf{F}_\lambda\| \quad (3.58)$$

$$\leq 2\|\mathbf{S}\| n\sqrt{n}b_M\|\mathbf{e}\|^2 + 2\|\mathbf{S}\| \|\mathbf{e}\| \sqrt{n} \sup_{i=1,\dots,n} \lambda^{n-i}|\eta_i(t)| \quad (3.59)$$

$$\leq 2\|\mathbf{S}\| n\sqrt{n}b_M\|\mathbf{e}\|^2 + \lambda\gamma\|\mathbf{e}\|^2 + \frac{n\|\mathbf{S}\|^2}{\lambda\gamma} \sup_{i=1,\dots,n} \lambda^{2(n-i)}\eta_i^2(t) \quad (3.60)$$

for every $\gamma > 0$, where the last inequality follows from Young's inequality. Back to the Lyapunov function, we have

$$\dot{\mathcal{V}} \leq -\lambda \left(q - \frac{2n\sqrt{n}b_M}{\lambda} \|\mathbf{S}\| - \gamma \right) \|\mathbf{e}\|^2 + \frac{n\|\mathbf{S}\|^2}{\gamma\lambda} \sup_{i=1,\dots,n} \lambda^{2(n-i)}\eta_i^2(t). \quad (3.61)$$

Consequently, if

$$\lambda > \lambda^* = \max \left\{ 1, \frac{4n\sqrt{n}b_M\|\mathbf{S}\|}{q} \right\}, \quad (3.62)$$

then

$$\dot{\mathcal{V}} \leq -\lambda \left(\frac{q}{2} - \gamma \right) \frac{\lambda_{\min}(\mathbf{S})}{\lambda_{\max}(\mathbf{S})} \mathcal{V} + \frac{n}{\gamma\lambda} \sup_{i=1,\dots,n} \lambda^{2(n-i)}\eta_i^2(t), \quad (3.63)$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the smallest and largest eigenvalues of $\mathbf{S}$, respectively. Picking $\gamma = q/4$, this implies

$$\mathcal{V}(t) \leq \exp \left(-\lambda \frac{q}{4} \frac{\lambda_{\min}(\mathbf{S})}{\lambda_{\max}(\mathbf{S})} \right) \mathcal{V}(t_0) + \frac{16n\lambda_{\max}(\mathbf{S})}{q^2\lambda^2\lambda_{\min}(\mathbf{S})} \sup_{i=1,\dots,n,s \in [t_0,t]} \lambda^{2(n-i)}\eta_i^2(s). \quad (3.64)$$

From where we can derive

$$\begin{aligned} \|\mathbf{e}(t)\| &\leq \exp \left(-\lambda \frac{q}{8} \frac{\lambda_{\min}(\mathbf{S})}{\lambda_{\max}(\mathbf{S})} (t - t_0) \right) \sqrt{\frac{\lambda_{\max}(\mathbf{S})}{\lambda_{\min}(\mathbf{S})}} \|\mathbf{e}(t_0)\| \\ &\quad + \frac{4}{q} \sqrt{\frac{n\lambda_{\max}(\mathbf{S})}{\lambda_{\min}(\mathbf{S})}} \sup_{i=1,\dots,n,s \in [t_0,t]} \lambda^{n-i-1} |\eta_i(s)|. \end{aligned} \quad (3.65)$$

Finally, as $\|\mathbf{e}(t)\| \geq |e_i(t)| = \lambda^{n-i} |\tilde{x}_i(t)|$, it follows that

$$|\tilde{x}_i(t)| \leq \exp(-\alpha\lambda(t - t_0))\beta \sqrt{\sum_{j=1}^n \lambda^{2(i-j)} \tilde{x}_j^2(t_0)} + \gamma \sup_{1 \leq j \leq n, s \in [t_0,t]} \lambda^{i-j-1} \eta_j^2(s) \quad (3.66)$$

$$\leq \exp(-\alpha\lambda(t - t_0))\beta \lambda^{i-1} \|\tilde{\mathbf{x}}(t_0)\| + \gamma \sup_{1 \leq j \leq n, s \in [t_0,t]} \lambda^{i-j-1} \eta_j^2(s), \quad (3.67)$$

with $\alpha = \frac{q}{8} \frac{\lambda_{\min}(\mathbf{S})}{\lambda_{\max}(\mathbf{S})}$, $\beta = \sqrt{\frac{\lambda_{\max}(\mathbf{S})}{\lambda_{\min}(\mathbf{S})}}$ and $\gamma = \frac{4}{q} \sqrt{\frac{n\lambda_{\max}(\mathbf{S})}{\lambda_{\min}(\mathbf{S})}}$. $\square$

We now use this result as a second block in our REWS estimation methodology.

Estimating the REWS

For REWS estimation, we complete the rotor dynamics (3.28) with a wind speed dynamics as follows

$$\dot{\omega} = \frac{1}{J}T_a(\omega, v, t) - \frac{1}{J}T_g(t) + \eta_1(t), \quad (3.68)$$

$$\dot{v} = \eta_2(t), \quad (3.69)$$

in which the dependence of the aerodynamic torque with respect to the pitch angle is recast as a time-dependence. In the second state equation, we express the derivative of the REWS as an unknown disturbance, since we do not have a model for it. Notice that this is not much problematic, as the high-gain observer can mitigate the effects of η_2 , according to Theorem 3.1. We assume that η_1 and η_2 are bounded functions.

Then, the high-gain observer equations should be

$$\dot{\hat{\omega}} = \frac{1}{J}T_a(\hat{\omega}, \hat{v}, t) - \frac{1}{J}T_g(t) + k_1\lambda(\omega - \hat{\omega}), \quad (3.70)$$

$$\dot{\hat{v}} = k_2\lambda^2(\omega - \hat{\omega}). \quad (3.71)$$

for some positive scalars k_1 , k_2 , and λ . Yet, as the aerodynamic torque is unknown, in lieu of T_a , we use the approximation of it given by the GPR previously designed, denoted $\hat{T}_a$. This yields the observer

$$\dot{\hat{\omega}} = \frac{1}{J}\hat{T}_a(\hat{\omega}, \hat{v}, t) - \frac{1}{J}T_g(t) + k_1\lambda(\omega - \hat{\omega}), \quad (3.72)$$

$$\dot{\hat{v}} = k_2\lambda^2(\omega - \hat{\omega}). \quad (3.73)$$

Convergence of the REWS estimate

In our discussion on high-gain observers, we presented Theorem 3.1, which states sufficient conditions for the state estimate error to exponentially converge to a region that we can make smaller by increasing the high-gain parameter, and the norm of the error actually converges to zero in the absence of disturbances.

To satisfy the first equation in Assumption 3.1, we need the dynamics of the first state variable, that is, ω , to be monotonic with respect to the second state variable, that is, v .

With the model (3.68)–(3.69), this means we need $T_a(\omega, v, t)$ to be monotonic with respect to v . Intuitively, it makes sense that the aerodynamic torque will increase with the wind speed. And, indeed, that is the case at least in typical operational conditions, as Fig. 3.11, which pictures the

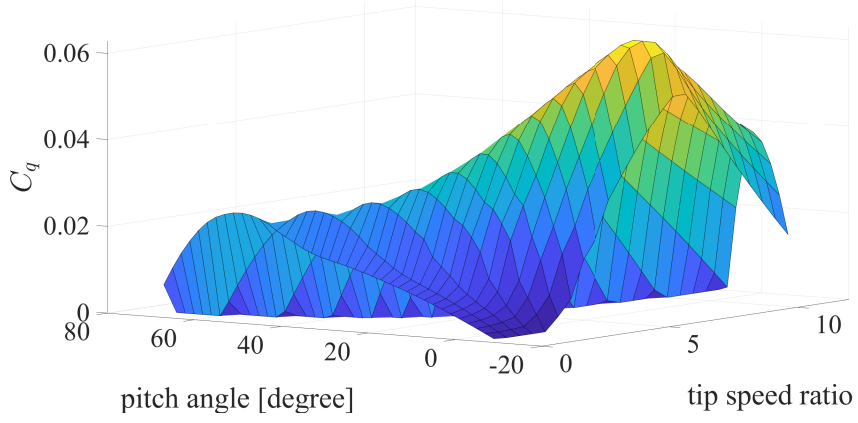


Figure 3.10: The C_q map for the turbine used in this study.

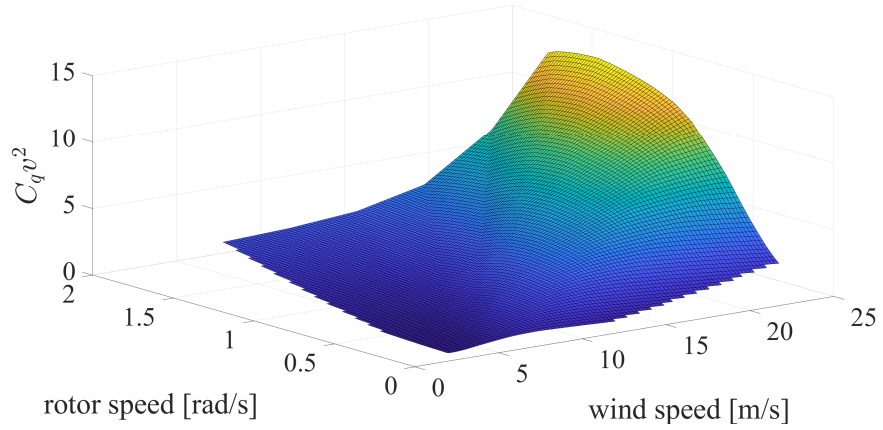


Figure 3.11: The term $C_q v^2$ as a function of v and ω with the pitch angle fixed to zero. Recall that $T_a \propto C_q v^2$.

term $C_q v^2$ (proportional to T_a) for $\beta = 0$, illustrates. Fig. 3.10 pictures the theoretical C_q map obtained from FAST simulations for the turbine under consideration in this study.

However, in our observer (3.72)–(3.73), we use the estimated aerodynamic torque obtained from a GPR. Thus, in our case, the monotonicity requirement with respect to v should hold for $\hat{T}_a$ rather than T_a . Even if T_a is monotonic with respect to v , the same property does not hold a priori for its estimate $\hat{T}_a$. Note that it is possible to enforce monotonicity of the output of a GPR model with respect to some of its inputs by choosing a grid of points and constraining the partial derivative to be positive in those points as detailed in [31, 110]. However, this complicates the implementation and can also add a substantial computational overhead to the use of the

method. Yet, it turns out that the monotonicity of the aerodynamic torque is sufficient for the conclusions of Theorem 3.1 to hold, as we prove below. We thus have the following theoretical guarantees regarding the convergence of our high-gain observer (3.72)–(3.73).

Corollary 3.1. *Consider the system (3.68)–(3.69) satisfying Assumption 3.1. There exist $k_1, k_2 \in \mathbb{R}$ and $\lambda^*, \alpha, \beta, \gamma > 0$ such that, for all $t \geq t_0$, the observer (3.72)–(3.73), with $\lambda > \lambda^*$, satisfies*

$$\begin{aligned} |\tilde{\omega}(t)| &\leq \beta \exp(-\alpha\lambda(t-t_0)) \|\tilde{\mathbf{x}}(t_0)\| \\ &\quad + \gamma \max_{s \in [t_0, t]} \left\{ \frac{1}{\lambda} \left(\eta_1(s) + \frac{T_a(\hat{\omega}(s), \hat{v}(s), s) - \hat{T}_a(\hat{\omega}(s), \hat{v}(s), s)}{J} \right)^2, \right. \\ &\quad \left. \frac{\eta_2^2(s)}{\lambda^2} \right\}, \end{aligned} \quad (3.74)$$

$$\begin{aligned} |\tilde{v}(t)| &\leq \beta\lambda \exp(-\alpha\lambda(t-t_0)) \|\tilde{\mathbf{x}}(t_0)\| \\ &\quad + \gamma \max_{s \in [t_0, t]} \left\{ \left(\eta_1(s) + \frac{T_a(\hat{\omega}(s), \hat{v}(s), s) - \hat{T}_a(\hat{\omega}(s), \hat{v}(s), s)}{J} \right)^2, \right. \\ &\quad \left. \frac{\eta_2^2(s)}{\lambda} \right\}, \end{aligned} \quad (3.75)$$

where $\tilde{\mathbf{x}}(t) = [\tilde{\omega}(t), \tilde{v}(t)]^T = [\omega(t) - \hat{\omega}(t), v(t) - \hat{v}(t)]^T$.

Proof. Let $\tilde{\mathbf{x}} = [\omega - \hat{\omega}, v - \hat{v}]^T = [\tilde{x}_1, \tilde{x}_2]^T$ and

$$\bar{\eta}_1 = \eta_1 + \frac{1}{J} \left(T_a(\hat{\omega}, \hat{v}, t) - \hat{T}_a(\hat{\omega}, \hat{v}, t) \right). \quad (3.76)$$

We have

$$\dot{\tilde{x}}_1 = \frac{1}{J} \left(T_a(\omega, v, t) - \hat{T}_a(\hat{\omega}, \hat{v}, t) \right) - k_1 \lambda \tilde{x}_1 + \eta_1 \quad (3.77)$$

$$\begin{aligned} &= \frac{1}{J} (T_a(\omega, v, t) - T_a(\hat{\omega}, \hat{v}, t)) + \frac{1}{J} \left(T_a(\hat{\omega}(t), \hat{v}(t), t) - \hat{T}_a(\hat{\omega}, \hat{v}(t), t) \right) \\ &\quad - k_1 \lambda \tilde{x}_1 + \eta_1 \end{aligned} \quad (3.78)$$

$$= \frac{1}{J} (T_a(\omega, v, t) - T_a(\hat{\omega}, \hat{v}, t)) + \bar{\eta}_1(t) - k_1 \lambda \tilde{x}_1 \quad (3.79)$$

$$\begin{aligned} &= \frac{1}{J} (T_a(\omega, v, t) - T_a(\omega, \hat{v}, t)) + \frac{1}{J} (T_a(\omega, \hat{v}, t) - T_a(\hat{\omega}, \hat{v}, t)) + \bar{\eta}_1(t) \\ &\quad - k_1 \lambda \tilde{x}_1. \end{aligned} \quad (3.80)$$

Introducing the re-scaled variables $e_i = \lambda^{2-i} \tilde{x}_i$ for $i = 1, 2$ and using the Mean-Value Theorem, one obtains

$$\dot{\mathbf{e}} = \lambda \mathbf{A}(t) \mathbf{e}(t) + \begin{bmatrix} \lambda F_1(t) \\ F_2(t) \end{bmatrix}, \quad (3.81)$$

where

$$F_1(t) = \frac{1}{J} (T_a(\omega, \hat{v}, t) - T_a(\hat{\omega}, \hat{v}, t)) + \bar{\eta}_1(t), \quad (3.82)$$

$$F_2(t) = \eta_2(t), \quad (3.83)$$

$$\mathbf{A}(t) = \begin{bmatrix} -k_1 & \frac{\partial T_a}{\partial v}(\omega, \theta(t), t) \\ -k_2 & 0 \end{bmatrix}, \quad (3.84)$$

and $\theta(t) \in [\min\{v(t), \hat{v}(t)\}, \max\{v(t), \hat{v}(t)\}]$. The rest of the proof of Theorem 3.1 then holds. $\square$

Notice that, as expected, the high-gain observer mitigates the effects of η_2 in the asymptotic error. This makes this design particularly suitable for REWS estimation, where wind speed modeling is a very challenging task.

One subtle point is that the bounds given by this corollary depend on the estimated values $\hat{v}$ and $\hat{\omega}$ through the term $T_a(\hat{\omega}(s), \hat{v}(s), s) - \hat{T}_a(\hat{\omega}(s), \hat{v}(s), s)$. If the model $\hat{T}_a$ is a good approximation for T_a globally, i.e., $|T_a(\hat{\omega}(s), \hat{v}(s), s) - \hat{T}_a(\hat{\omega}(s), \hat{v}(s), s)| < \epsilon$ for any values of the inputs for some constant ϵ , then we can replace this term by ϵ and have bounds that do not depend on the estimation. This, together with the fact that we cannot reduce the effect of the term $T_a - \hat{T}_a$ by increasing the gain λ , reinforces that we need a precise model $\hat{T}_a$ for this strategy to be effective.

3.5 Numerical results for the REWS estimation with GPR and high-gain observer

To test our methodology, we use a set-up similar to the one detailed for the pure data-based approach, presented in Section 3.3. We fixed the number of clusters to 5 and use a sampling rate 10 times that of the SCADA system for the observer, using a zero-order holder to interpolate values.

We selected some values of λ to illustrate the performance of the observer here. We need to be careful not to select a value too large because of the well-known peaking phenomenon of the high-gain observer (that leads the state estimates to be like impulse functions in the transient) as well as the fact that we use a discrete-time implementation, so the approximation error due to the discretization is large when the state variables change a lot in between time-steps.

Fig. 3.12 shows some results for different choices of λ . We see that a larger λ yields noisier results, whereas a small λ is slow to follow the reference. In Fig. 3.13, we show the box plots corresponding to the distributions of the relative error for the first 1000 points. Recall this error is defined as

$$e_* = \frac{\hat{v} - v}{v} \times 100\%. \quad (3.85)$$

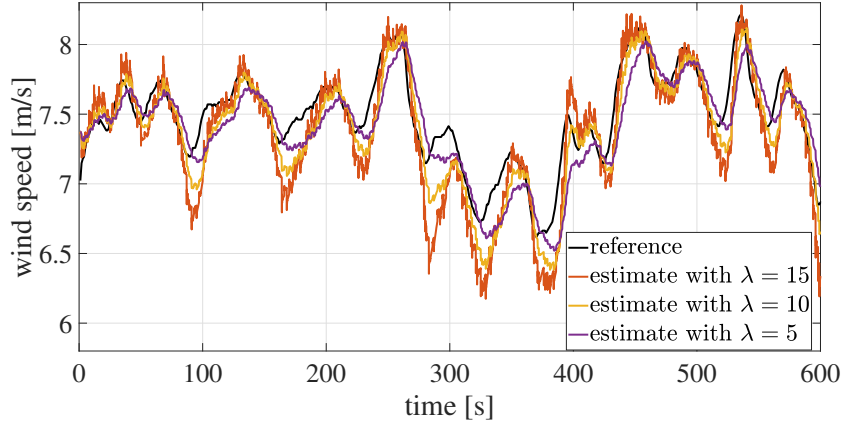


Figure 3.12: Comparison in performance among different values of λ for the same reference. The reference data is the same as in Fig. 3.5a.

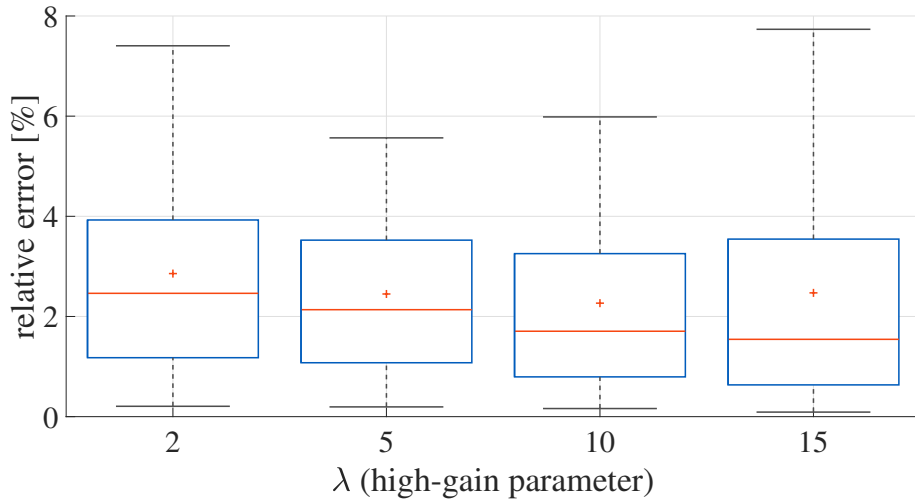


Figure 3.13: Relative error for different values of λ corresponding to estimation results shown in Fig. 3.12. The crosses represent the mean values, the red lines are the medians, the boxes are in between the 25th and 75th percentiles and the whiskers are in between the 5th and 95th percentiles.

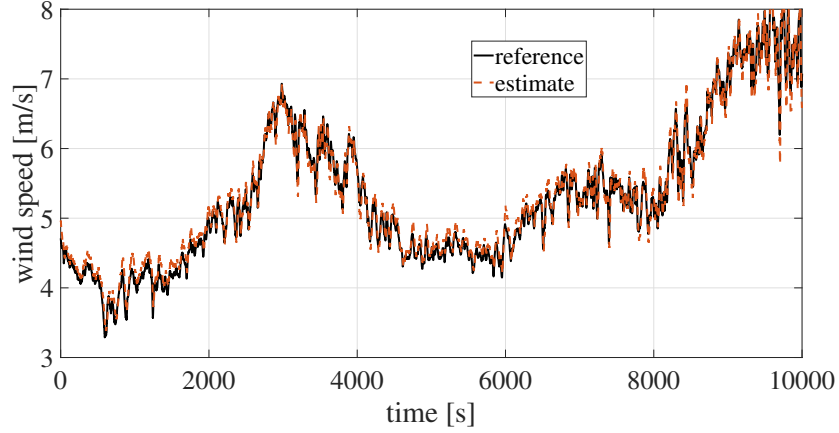


Figure 3.14: Estimation with $\lambda = 10$ for a different reference.

Finally, Fig. 3.14 presents the results for a different reference, where lower wind speed values appear. We use just $\lambda = 10$ in this case as it was the best choice previously. The error distribution is similar to that of the previous case.

Generally speaking, one can observe that the REWS estimation is slightly less accurate than the first purely data-based method, as the relative errors have a magnitude four times more important. This is in accordance with the fact that the GPR was trained with fewer inputs. Yet, the magnitude of this error still remains low. Besides, this observer-based method still has its merits. First, it uses a standard GPR with low dimensional input unlike the previous method. Second, as mentioned previously, this method provides the user with a measure of the current accuracy of the GPR (which could vary with time, due to aging of the turbine with fatigue, for instance). In detail, it is unlikely that the rotor speed estimation error $\tilde{\omega}$ could be used to provide an information on the magnitude of the error $\hat{T}_a - T_a$, due to the appearance of λ in the denominator of the bound (3.74). However, one could still perform an open-loop estimator of $\hat{\omega}$ and compare it with the actual rotor speed to provide some indication on the accuracy of the GPR.

3.6 Conclusion

This chapter presented two methods for estimating the REWS acting on a turbine. The first one is purely based on machine learning and works by learning a map between the data provided by the turbine's SCADA system and the wind speed. The second one combines a GPR approach to estimate the aerodynamic torque on the turbine with a high-gain observer that estimates the wind speed.

In both cases, the methods aim at overcoming the constraint of LiDAR

availability or knowledge of the torque coefficient of the turbine. They were trained with wind speed data obtained from (temporarily available) LiDAR sensors and use low-level data, which can be obtained from the Supervisory Control And Data Acquisition (SCADA) system of an industrial-scale wind turbine.

We presented numerical results for both cases using experimental data. For the second method, we also presented a mathematical resulting regarding the convergence of the REWS estimate, depending on the estimation error of the aerodynamic torque.

For the second method, inspired by [92] and [25], we chose a high-gain observer design because it enables us to put little effort in the modeling of the wind speed dynamics. Future research may focus on the possibility to adapt the ideas presented in [25] to estimate both parameters and states, with the hope to cope with time variations of the turbine's torque coefficient map.

Grounding on this REWS estimation (or on another approach available in the literature) at the local level of one turbine, the remaining chapters of this manuscript will address the question of estimating the free-flow wind at the level of a wind farm.

Chapter 4

Free-flow wind speed estimation in a wind farm from measurements inside the wake

Résumé. L'objectif de ce chapitre est d'estimer la vitesse du vent en écoulement libre affectant un parc éolien à partir de mesures de vitesse locales, fournies par une éolienne dans la zone de sillage par exemple. Pour ce faire, nous considérons une ferme avec une géométrie simplifiée, où toutes les éoliennes sont alignées. On considère par ailleurs que la direction du vent est connue.

Pour résoudre ce problème, nous proposons d'utiliser un modèle simplifié de sillage proposé par Shapiro [119]. Ce modèle décrit l'effet du sillage à l'aide d'équations aux dérivées partielles à une dimension, dont la vitesse de transport est la vitesse du vent en écoulement libre, c'est-à-dire la vitesse à estimer. Ces équations ont des solutions analytiques basées sur des retards de transport, ce qui permet une estimation en temps réel de la vitesse en écoulement libre du vent, basée sur une stratégie d'estimation simple. Nous fournissons une preuve théorique de la convergence de l'algorithme d'estimation et illustrons son intérêt en simulations avec des données numériques issues du simulateur FAST.Farm [66, 67].

4.1 Introduction

In this chapter, we consider the problem of estimating the free-flow wind speed affecting a wind farm from a local wind speed measurement (obtained, e.g., with the methods detailed in Chapter 3). This information could then be used to provide an estimate of the distributed wind profile inside a wind

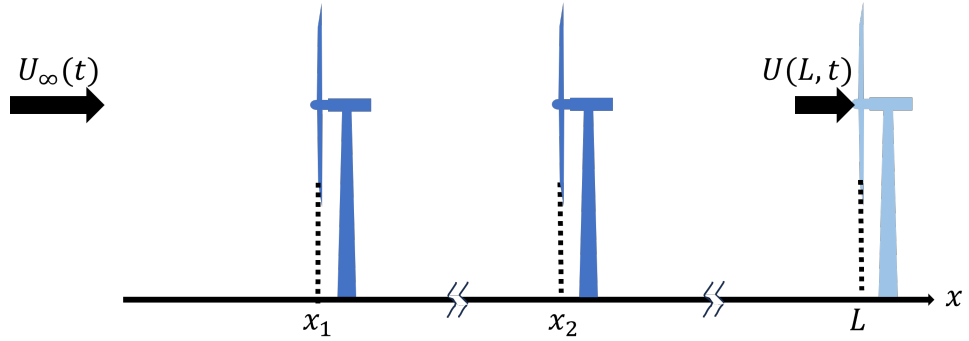


Figure 4.1: Array of wind turbines and downwind measurement device. In this case, we have an array of two wind turbines and the measurement is given by a third turbine (equipped with a REWS estimation algorithm, for instance).

farm.

More specifically, in this chapter, we consider that the wind direction is known and we wish to estimate the time-varying free-flow wind speed from local speed information provided by turbines located inside the wake zone of an array of wind turbines or by a meteorological mast located downwind of the array.

Fig. 4.1 shows a schematic view of such a situation, where an array of wind turbines is aligned, with the wind direction. Such a geometry occurs for instance in the d’Ablaincourt wind farm, as illustrated in Fig. 4.2, or in the three Horns Rev wind farms in Denmark¹, as also illustrated in Fig. 4.2. Note that the wind farm layout is supposed to ensure optimal production, which is not achieved when the wind is aligned with the turbines array and thus, somehow, makes this configuration of alignment between the wind and turbines specific. We consider this case study in this chapter as a preliminary study, for the sake of simplicity, before extending its scope in the following chapter.

In this chapter, consequently, we consider an array of wind turbines such as depicted in Fig. 4.1 and wish to estimate the free-flow wind speed from a downwind measurement inside the farm, which is thus affected by the wake of the upwind turbines.

To address this issue, we propose to model the wake with a simple control-oriented model, referred to as Shapiro’s model [119, 122] (see the Introduction, Section 2.3.3). This model, detailed in the sequel, describes the wake of a wind turbine through an 1-D transport Partial Differential Equation (PDE), the transport speed of which is the free-flow wind speed, that is, the variable to be estimated. As this equation admits an analytical

¹For the Horns Rev 1, we also showed a photo in the Introduction (Fig. 2.4), where we can see the downwind turbines inside of the wake zone of the upwind ones.

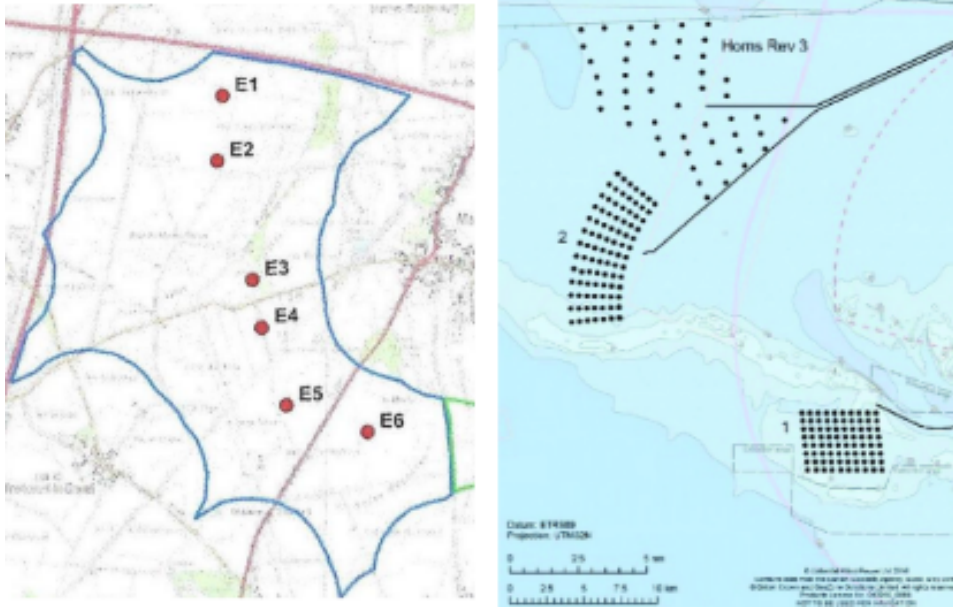


Figure 4.2: Maps of a portion of the Ablaincourt wind farm (left-hand side) [86, p. 30] and the Horns Rev wind farms (right-hand side) [123, Fig. 2].

solution expressed through transport delays, this opens the door for a simple real-time estimation technique, which we will detail in the sequel.

This chapter is organized as follows. First, Section 4.2 delves into the details of the wake model we propose to use, Shapiro’s model. Then, in Section 4.3, we present the analytical solution to this model, through transport delays. Section 4.4 then presents the estimation strategy that we propose to follow, before providing a theorem presenting the convergence properties that we were able to establish, along with its corresponding proof. We then conclude this chapter with numerical simulations obtained by interfacing our proposed estimation strategy with the simulator FAST.Farm.

4.2 Shapiro’s wake modeling and problem statement

In this section, we detail the model of the speed deficit caused by wind turbines proposed in [119], which we refer to as Shapiro’s 1-D model. This model was initially proposed for a constant free-flow speed, but it has since been applied also to the time-varying case [121].

Given an array of N aligned wind turbines, Shapiro’s 1-D model char-

acterizes the speed deficit δu_n caused by the n th turbine by²

$$\frac{\partial \delta u_n}{\partial t} + U_\infty(t) \frac{\partial \delta u_n}{\partial x} = -w_n(x) U_\infty(t) \delta u_n(x, t) + S_n(x, U_\infty(t)), \quad (4.1)$$

$$\delta u_n(0, t) = 0, \quad (4.2)$$

where $x \geq 0$ is direction of the wind and

$$w_n(x) = \frac{2}{d_n(x)} \frac{dd_n}{dx}(x), \quad (4.3)$$

$$S_n(x, U_\infty(t)) = \frac{2a_n U_\infty^2(t)}{d_n^2(x)} G_n(x), \quad (4.4)$$

$$d_n(x) = 1 + \kappa_n \ln \left(1 + \exp \left(\frac{x - x_n - D}{D/2} \right) \right), \quad (4.5)$$

$$G_n(x) = \frac{1}{(D/2)\sqrt{2\pi}} \exp \left(-\frac{(x - x_n)^2}{2(D/2)^2} \right). \quad (4.6)$$

Here, U_∞ is the free-flow wind speed (which is always positive), δu_n is the speed deficit caused by the n th turbine in the array, $\kappa_n > 0$ is a coefficient related to the wake diameter expansion, $a_n > 0$ is the induction factor of the n th turbine³, $D > 0$ is the rotor diameter (assumed the same for all turbines), and $x_n > 0$ is the x position of the n th turbine. For convenience, we label the turbines so that $x_1 < x_2 < \dots < x_N$. It should be noted that the boundary of the domain should be far upwind from the turbine array, i.e., $x_1 \gg 0$.

In this model, d_n represents the wake diameter divided by the rotor diameter, i.e., the normalized wake diameter. Notice the further downwind from the turbine, the larger the wake diameter is, which matches the discussion about wake modeling presented in the Introduction. The term $-w_n(x)U_\infty(t)\delta u_n$ introduces a decay for the speed deficit, i.e., the speed deficit becomes smaller for positions further downwind from the turbine, which is the wake recovery phenomenon. The term $S_n(x, U_\infty(t))$ models the turbine as a sink of momentum, but instead of a punctual sink, it is distributed as a Gaussian around the position of the turbine according to the function G_n to yield a smoother solution.

We consider the wind speed at a position x and time t then writes in

²The PDE is valid for all $x > 0$, but, as we will see later, we will be interested only in a domain $x \in [0, L]$, where $L > 0$ represents the position where the measurement is taken as shown in Fig. 4.1.

³We assume each induction factor to be constant throughout the time window of interest. The turbine controllers typically try to keep a constant induction factor, or, equivalently, C_p , when operating in the below rated region [62].

terms of a linear superposition of the speed deficits, i.e, as

$$U(x, t) = U_\infty(t) - \sum_{n=1}^N \delta u_n(x, t). \quad (4.7)$$

Notice that this is a simplification over the original model presented in [119]. In fact, several models are reported in the literature to superpose the wakes of multiple turbines [53]. Here, we opted for the simple linear relation (4.7) as it is more convenient to handle but is also reported in [135] as an accurate enough for control-oriented modeling.

Finally, we consider the measured output

$$y(t) = U(L, t), \quad (4.8)$$

where the constant $L > 0$ represents the position where we obtain our measurement.

The estimation problem under consideration, thus, consists in reconstructing the free-flow wind speed U_∞ from the measurement (4.8) provided at position L .

Numerous studies have explored the development of controllers and observers for systems governed by first-order hyperbolic PDEs, such as the one under consideration here. These studies include the estimation of uncertain source terms or boundary coefficients, see, e.g., [18, 32, 48]. Yet, in all of these cited references, the transport speed of the PDEs is assumed to be known. In fact, the problem of estimating transport speed is well-known to be a complex task, even more in the case of a time-varying speed. Up to our knowledge, one of the only works in the literature addressing speed estimation for hyperbolic PDEs is [4]. However, their proposed method necessitates boundary sensing at both extremities of the domain, a requirement which is not met here.

Besides, notice that the problem under consideration differs from those usually considered in the literature of estimation for systems described by hyperbolic PDEs since we do not directly measure the boundary value of the quantity governed by the transport PDE but the difference between the free-flow wind speed and the speed deficit governed by the PDE. In some sense, this output carries out more information about the transport speed, allowing us to estimate it.

To tackle this problem, we propose to adopt an observer-like strategy. Given a free-flow wind speed estimate, we build on the previous wake model to generate an artificial output that is supposed to estimate the real measurement. The difference between these two measurements (real and artificial) then serves as the basis for updating our estimate of the free-flow wind speed. This approach results in a straightforward and easily implementable algorithm.

To build the speed deficit estimate, rather than relying on the equations (4.1)-(4.2) directly, we propose to use their analytical time-delay solution, that we now present.

Remark. It is important to note that the estimation method proposed in the following sections holds for generic continuous functions $w_n : [0, L] \mapsto \mathbb{R}$ and $S_n : [0, L] \times \mathbb{R}_+ \mapsto \mathbb{R}$. The specific definitions of the functions w_n and S_n provided in (4.3)–(4.6) of Shapiro’s model are only used later in Section 4.6 for simulation purposes.

4.3 Analytical expression of the speed deficit with transport delays

It is well-known that first-order hyperbolic PDEs such as the one under consideration here have strong connections with time-delay systems and can be recast as retarded ones, see, e.g., [10, 38, 109], through the method of characteristic lines. Here, we propose to follow such an approach to formulate an explicit solution of (4.1)-(4.2) which can thus be efficiently computed and used in our estimation strategy.

As the transport speed in (4.1) is time-varying, this reformulation thus involves a specific class of structured delays, that we refer to as transport delays [24] (but are also called hydraulic delays in the literature [29]). Such delays are implicitly defined as the solution of an integral relation depending on past values of the input [133]. They are the exact solution of a plug-flow transport equation [105], hence their designation as hydraulic or transport delays. We now detail this reformulation.

Consider (4.1) with BC (4.2) and assume⁴ there is a positive constant U_m such that $U_\infty(t) \geq U_m$ for all t . We append to the problem (4.1)–(4.2) the initial condition

$$\delta u_n(x, -T) = b(x), \quad (4.9)$$

where $T > 0$ is a sufficiently large number⁵, such that the initial condition does not affect the solution for $t \geq 0$, $x \in [0, L]$, and b is some function that will play no role in the solution in the domain of interest $(x, t) \in [0, L] \times [0, \infty)$.

Let us start by introducing the transport delay τ , which is defined by

$$\int_{t-\tau(x,t)}^t U_\infty(r) dr = x. \quad (4.10)$$

⁴This assumption must hold for a PDE of the form (4.1) to have a well-defined solution for all x and t . In our case, this assumption means that the wind does not fully stop nor reverse direction in the time frame of interest.

⁵ T needs to be larger than the largest possible transport delay inside the relevant domain, which is L/U_m . This delay will be explained in the sequel.

Note that, since U_∞ is always positive, this equation uniquely defines τ (see [24]). Also, this integral relation corresponds to the integration along the characteristic lines of (4.1). We will also need

$$W_n(x) = \exp \left(\int_0^x w_n(\xi) d\xi \right), \quad (4.11)$$

$$\Lambda(s, t) = \int_s^t U_\infty(r) dr. \quad (4.12)$$

We can establish the following lemma.

Lemma 4.1. *Assume there is $U_m > 0$ such that $U_\infty(t) \geq U_m$ for all t . Consider δu_n satisfying the PDE (4.1) with boundary condition (4.2) and initial condition (4.9). Then,*

$$\begin{aligned} \delta u_n(x, t) &= \frac{1}{W_n(x)} \int_0^x \frac{W_n(\xi) S_n(\xi, U_\infty(t - \tau(x - \xi, t)))}{U_\infty(t - \tau(x - \xi, t))} d\xi \\ &= \frac{1}{W_n(x)} \int_{t-\tau(x, t)}^t W_n(x - \Lambda(s, t)) S_n(x - \Lambda(s, t), U_\infty(t)) ds, \end{aligned} \quad (4.13)$$

for $t > 0$ and $x \in [0, L]$, and in which τ , W_n , and Λ have been defined in (4.10), (4.11), and (4.12), respectively.

Proof. We can obtain the analytical expression for δu_n through the following procedure. First, introduce the new quantity

$$\vartheta(x, t) = W_n(x) \delta u_n(x, t), \quad (4.15)$$

with W_n as in (4.11). It is easy to show that ϑ satisfies

$$\frac{\partial \vartheta}{\partial t} + U_\infty(t) \frac{\partial \vartheta}{\partial x} = h_1(x, t), \quad (4.16)$$

$$\vartheta(0, t) = 0, \quad (4.17)$$

$$\vartheta(x, -T) = h_2(x), \quad (4.18)$$

where $h_1(x, t) = W_n(x) S_n(x, U_\infty(t))$ and $h_2(x) = W_n(x) b(x)$. Then, we can apply the Laplace transform to change from the variable x to a variable χ , which yields

$$\frac{\partial \vartheta^*}{\partial t} + \chi U_\infty(t) \vartheta^*(\chi, t) = h_1^*(\chi, t), \quad (4.19)$$

$$\vartheta^*(\chi, -T) = h_2^*(\chi), \quad (4.20)$$

where the superscript $*$ denotes the transformed functions. It is well-known that the solution for an initial value problem of this form is

$$\vartheta^*(\chi, t) = e^{-\int_{-T}^t \chi U_\infty(r) dr} \left[h_2^*(\chi) + \int_{-T}^t e^{\int_{-T}^s \chi U_\infty(r) dr} h_1^*(\chi, s) ds \right] \quad (4.21)$$

$$= e^{-\int_{-T}^t \chi U_\infty(r) dr} h_2^*(\chi) + \int_{-T}^t e^{-\int_s^t \chi U_\infty(r) dr} h_1^*(\chi, s) ds. \quad (4.22)$$

Then, we have

$$\vartheta^*(\chi, t) = e^{-\chi\Lambda(-T, t)}h_2^*(\chi) + \int_{-T}^t e^{-\chi\Lambda(s, t)}h_1^*(\chi, s)ds, \quad (4.23)$$

with Λ as in (4.12). Applying the inverse Laplace transform, we have

$$\begin{aligned} \vartheta(x, t) &= h_2(x - \Lambda(-T, t))\mathbb{1}(x - \Lambda(-T, t)) \\ &\quad + \int_{-T}^t h_1(x - \Lambda(s, t), s)\mathbb{1}(x - \Lambda(s, t))ds, \end{aligned} \quad (4.24)$$

where $\mathbb{1}(s) = 1$ if $s \geq 0$ and $\mathbb{1}(s) = 0$ otherwise.

Recall we are interested in the solution in the domain $(x, t) \in [0, L] \times [0, \infty)$, $U_\infty(t) \geq U_m > 0$ for all t , and T is sufficiently large so that the initial condition does not affect the solution if $(x, t) \in [0, L] \times [0, \infty)$. If we set $T > L/U_m$, then $x - \Lambda(-T, t) < 0$ and $t - \tau(x, t) > -T$ in the domain of interest. Hence, we have

$$\vartheta(x, t) = \int_{t-\tau(x, t)}^t h_1(x - \Lambda(s, t))ds \quad (4.25)$$

$$= \int_{t-\tau(x, t)}^t W_n(x - \Lambda(s, t))S_n(x - \Lambda(s, t), U_\infty(t))ds. \quad (4.26)$$

Therefore, the solution is

$$\begin{aligned} \delta u_n(x, t) &= \frac{\vartheta(x, t)}{W_n(x)} \\ &= \frac{1}{W_n(x)} \int_{t-\tau(x, t)}^t W_n(x - \Lambda(s, t))S_n(x - \Lambda(s, t), U_\infty(t))ds. \end{aligned} \quad (4.27)$$

(4.28)

We can also express δu_n with an integral over space instead of an integral over time. Define $\xi = x - \Lambda(s, t)$, then $d\xi = U_\infty(s)ds$ and $s = t - \tau(x - \xi, t)$, which can be seen by noticing that $\Lambda(t - \tau(x - \xi, t), t) = x - \xi$. We also have that $\xi = 0$ when $s = t - \tau(x, t)$ and $\xi = x$ when $s = t$. Hence,

$$\delta u_n(x, t) = \frac{1}{W_n(x)} \int_0^x \frac{W_n(\xi)S_n(\xi, U_\infty(t - \tau(x - \xi, t)))}{U_\infty(t - \tau(x - \xi, t))}d\xi. \quad (4.29)$$

□

Thus, Lemma 4.1 provides an explicit solution to (4.1)–(4.2), which can be efficiently computed. We propose to take advantage of this solution in our estimation methodology.

4.4 Proposed estimation method

Our transport speed estimation method grounds on the explicit solution provided by Lemma 4.1.

Let $\hat{U}_\infty(t)$ represent the (sufficiently smooth) estimate of $U_\infty(t)$. Similarly, from (4.8), define the estimate of the measured output at time t , $\hat{y}(t)$, as

$$\hat{y}(t) = \hat{U}_\infty(t) - \sum_{n=1}^N \delta \hat{u}_n(t), \quad (4.30)$$

where

$$\delta \hat{u}_n(t) = \frac{1}{W_n(L)} \int_0^L \frac{W_n(\xi) S_n(\xi, \hat{U}_\infty(t - \hat{\tau}(L - \xi, t)))}{\hat{U}_\infty(t - \hat{\tau}(x - \xi, t))} d\xi, \quad (4.31)$$

and $\hat{\tau}$ is defined by

$$\int_{t-\hat{\tau}(x,t)}^t \hat{U}_\infty(r) dr = x. \quad (4.32)$$

Notice (4.31) and (4.32) correspond to (4.13) and (4.10), respectively, but they use the estimate $\hat{U}_\infty$ instead of the actual speed U_∞ . So, $\delta \hat{u}_n(t)$ is the estimated n th turbine speed deficit at time t and at position $L > 0$, which is the position where the measurement is taken, and $\hat{\tau}$ is the estimate of the transport delay. Recall that N is the number of turbines upwind of the position where the measurement is taken.

We propose to update $\hat{U}_\infty$ as follows

$$\dot{\hat{U}}_\infty(t) = k[y(t) - \hat{y}(t)], \quad (4.33)$$

where $k > 0$ is a gain to be chosen by the user of the method and $y(t)$ is the measurement obtained at time t as in (4.8).

Yet, for the transport delay (4.32) to be well-defined, it is necessary to guarantee that $\hat{U}_\infty$ remains positive. With this in view, we consider $\hat{U}_m > 0$ and saturate the speed estimate $\hat{U}_\infty$ on the interval $[\hat{U}_m, \infty)$, that is, we modify (4.33) as follows

$$\dot{\hat{U}}_\infty(t) = \text{Proj}_{[\hat{U}_m, \infty)} \{k[y(t) - \hat{y}(t)]\} \quad (4.34)$$

$$= \begin{cases} 0, & \text{if } y(t) \leq \hat{y}(t) \text{ and } \hat{U}_\infty(t) = \hat{U}_m, \\ k[y(t) - \hat{y}(t)], & \text{otherwise.} \end{cases} \quad (4.35)$$

4.4.1 Convergence result

We start by stating some assumptions. about the problem and then show that, under those assumptions, we have a convergence result.

Assumption 4.1. For every $x \in [0, L]$ and every $n \in \{1, \dots, N\}$, there exists $M_n(x)$ such that, for all positive λ_1, λ_2 ,

$$\left| \frac{S_n(x, \lambda_1)}{\lambda_1} - \frac{S_n(x, \lambda_2)}{\lambda_2} \right| \leq M_n(x) |\lambda_1 - \lambda_2|, \quad (4.36)$$

i.e., the function $F_n : (x, \lambda) \in [0, L] \times (0, \infty) \mapsto S_n(x, \lambda)/\lambda$ is globally Lipschitz with respect to its second argument.⁶

Assumption 4.2. There exist constants $U_m > 0$ and $\zeta \geq 0$ such that $U_\infty(t) \geq U_m$ and $|\dot{U}_\infty(t)| \leq \zeta$ for all $t \in \mathbb{R}$.

Assumption 4.3. Let

$$\alpha_n \triangleq \frac{1}{W_n(L)} \int_0^L W_n(\xi) M_n(\xi) d\xi, \quad (4.37)$$

$$\beta_n \triangleq \frac{1}{W_n(L)} \int_0^L W_n(\xi) (L - \xi) M_n(\xi) d\xi, \quad (4.38)$$

$$Z \triangleq \sum_{i=1}^N \left(\alpha_n + \frac{\beta_n \zeta}{U_m^2} \right). \quad (4.39)$$

It holds

$$Z < 1. \quad (4.40)$$

Assumption 4.2 requires the free-flow to be lower-bounded by a positive constant, which, in turn, ensures that the transport delay defined in (4.10) is upper-bounded (we had already introduced this bound U_m in Section 4.3 but formalize it here for the sake of self-containedness of the following theorem). This and the fact that the variations of the free-flow are bounded are reasonable requirements from a practical point of view.

Assumption 4.3 is the most demanding and technical one. It originates from the use of Halanay's inequality in our stability proof, as will be detailed in the sequel, and somehow restricts the magnitude of the speed variations and requires the speed to be large enough. This requirement did not seem restrictive in practice, as we will detail in the simulations section.

Let us define the error variable

$$\tilde{U}_\infty(t) = U_\infty(t) - \hat{U}_\infty(t). \quad (4.41)$$

The following result then holds.

Theorem 4.1. Consider the system (4.1)–(4.2) satisfying Assumptions 4.1, 4.2, and 4.3 and the free-flow wind speed estimate $\hat{U}_\infty$ defined according to

⁶In the particular case of Shapiro's model, i.e., with S_n given by (4.4), Assumption 4.1 is trivially satisfied with $M_n(x) = 2a_n G_n(x)/d_n^2(x)$.

(4.30)–(4.33). Furthermore, consider there exists $\hat{U}_m \in \left(\sqrt{\frac{\zeta \sum_{n=1}^N \beta_n}{1 - \sum_{n=1}^N \alpha_n}}, U_m \right)$ such that $\sup_{s \in [-L/\hat{U}_m, 0]} |\tilde{U}_\infty(s)| < U_m - \hat{U}_m$. Then, there exists $k_* > 0$ dependent on $\hat{U}_m$ and $\sup_{s \in [-L/\hat{U}_m, 0]} |\tilde{U}_\infty(s)|$ such that, if $k > k_*$, then, for all $t \geq 0$, $\hat{U}_\infty(t) \geq \hat{U}_m$ and

$$|\tilde{U}_\infty(t)| \leq \sup_{s \in [-L/\hat{U}_m, 0]} |\tilde{U}_\infty(s)| e^{-\sigma t} + \frac{\zeta}{k \left[1 - \sum_{i=1}^N \left(\alpha_n + \frac{\beta_n \zeta}{\hat{U}_m^2} \right) \right]}, \quad (4.42)$$

for some $\sigma > 0$ that depends on k and $\hat{U}_m$.

Notice Theorem 4.1 states the estimation error is bounded by the sum of two terms. The first one is related to the initial condition and goes to zero exponentially. The second one is related to ζ and one can make it arbitrarily small by choosing a large gain k . According to this result, one should thus adapt the update gain k according to the magnitude of the free-flow variations: the more important they are, the larger the gain should be, to faster counteract them. Furthermore, the values of k_* and σ also depend on ζ , as detailed in the proof we give in the sequel.

It is also worth underlining that, to hold, Theorem 4.1 requires the initial estimation error to be sufficiently small. This requirement aims at avoid reaching the lower bound $\hat{U}_m$ and thus to saturate the update law. This technical assumption does not seem to be so restrictive in practice as the simulations detailed later illustrate.

4.4.2 Proof of Theorem 4.1

Let

$$\delta \tilde{u}_n(t) \triangleq \delta u_n(L, t) - \delta \hat{u}_n(t), \quad (4.43)$$

$$\delta \tilde{u}(t) \triangleq \sum_{n=1}^N \delta \tilde{u}_n(t). \quad (4.44)$$

Consider the Lyapunov function candidate $\mathcal{V}(t) \triangleq \tilde{U}_\infty^2(t)/2$. We have

$$\dot{\mathcal{V}}(t) = \tilde{U}_\infty(t) \dot{\tilde{U}}_\infty(t) = \tilde{U}_\infty(t) (\dot{U}_\infty(t) - \dot{\hat{U}}_\infty(t)). \quad (4.45)$$

Notice that

$$\dot{\hat{U}}_\infty(t) = k(y(t) - \hat{y}(t)) = k(\tilde{U}_\infty(t) - \delta \tilde{u}(t)). \quad (4.46)$$

Using this and Assumption 4.2, we have

$$\dot{\mathcal{V}}(t) \leq |\tilde{U}_\infty(t)| \zeta - k \tilde{U}_\infty^2(t) + k |\tilde{U}_\infty(t)| |\delta \tilde{u}(t)|. \quad (4.47)$$

Next, we will find an upper-bound for $|\delta \tilde{u}(t)|$.

Notice that, from (4.13) and (4.31),

$$\begin{aligned}\delta\tilde{u}_n(t) &= \frac{1}{W_n(L)} \int_0^L \frac{W_n(\xi) S_n(\xi, U_\infty(t - \tau(L - \xi, t)))}{U_\infty(t - \tau(L - \xi, t))} d\xi \\ &\quad - \frac{1}{W_n(L)} \int_0^L \frac{W_n(\xi) S_n(\xi, \hat{U}_\infty(t - \hat{\tau}(L - \xi, t)))}{\hat{U}_\infty(t - \hat{\tau}(L - \xi, t))} d\xi\end{aligned}\quad (4.48)$$

$$\begin{aligned}&= \frac{1}{W_n(L)} \int_0^L W_n(\xi) [F_n(\xi, U_\infty(t - \tau(L - \xi, t))) \\ &\quad - F_n(\xi, \hat{U}_\infty(t - \hat{\tau}(L - \xi, t)))] d\xi,\end{aligned}\quad (4.49)$$

where $F_n(x, U_\infty(t)) = S_n(x, U_\infty(t))/U_\infty(t)$. Now, notice that we can add and subtract $W_n(\xi)F_n(\xi, U_\infty(t - \hat{\tau}(L - \xi, t)))$ under the integral sign to yield

$$\begin{aligned}\delta\tilde{u}_n(t) &= \left\{ \frac{1}{W_n(L)} \int_0^L W_n(\xi) [F_n(\xi, U_\infty(t - \hat{\tau}(L - \xi, t))) \right. \\ &\quad \left. - F_n(\xi, \hat{U}_\infty(t - \hat{\tau}(L - \xi, t)))] d\xi \right\} \\ &\quad + \left\{ \frac{1}{W_n(L)} \int_0^L W_n(\xi) [F_n(\xi, U_\infty(t - \tau(L - \xi, t))) \right. \\ &\quad \left. - F_n(\xi, U_\infty(t - \hat{\tau}(L - \xi, t)))] d\xi \right\}\end{aligned}\quad (4.50)$$

$$\triangleq I_1 + I_2, \quad (4.51)$$

with I_1 the first pair of curly brackets on the right-hand side of (4.50) and I_2 the second one.

Assume for now that $\hat{U}_\infty(t) \geq \hat{U}_m \in (0, U_m)$ for $t \geq 0$. Notice this, together with $U_\infty(t) \geq U_m > \hat{U}_m$, implies that both τ and $\hat{\tau}$ are upper bounded by $\bar{\tau} = L/\hat{U}_m$. We can upper bound I_1 as:

$$I_1 \leq \frac{1}{W_n(L)} \int_0^L W_n(\xi) M_n(\xi) |\tilde{U}_\infty(t - \hat{\tau}(L - \xi, t))| d\xi \quad (4.52)$$

$$\leq \alpha_n \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(t + s)|, \quad (4.53)$$

where α_n is as in (4.37).

For I_2 , we can first use the Lipschitz property of F_n (Assumption 4.1) to write

$$\begin{aligned}&|F_n(\xi, U_\infty(t - \tau(L - \xi, t))) - F_n(\xi, U_\infty(t - \hat{\tau}(L - \xi, t)))| \\ &\leq M_n(\xi) |U_\infty(t - \tau(L - \xi, t)) - U_\infty(t - \hat{\tau}(L - \xi, t))|.\end{aligned}\quad (4.54)$$

According to the Mean Value Theorem, there exists $\tau_*(L - \xi, t)$ such that

$$\begin{aligned}&|U_\infty(t - \tau(L - \xi, t)) - U_\infty(t - \hat{\tau}(L - \xi, t))| \\ &= |(\tau(L - \xi, t) - \hat{\tau}(L - \xi, t)) \times \dot{U}_\infty(t - \tau_*(L - \xi, t))| \\ &\leq |(\tau(L - \xi, t) - \hat{\tau}(L - \xi, t))| \zeta.\end{aligned}\quad (4.55)$$

Furthermore, we can bound $|\tau - \hat{\tau}|$ noticing that, from the definitions of τ and $\hat{\tau}$ in (4.10) and (4.32), respectively, we have

$$\int_{t-\hat{\tau}(x,t)}^t \hat{U}_\infty(r) dr = x = \int_{t-\tau(x,t)}^t U_\infty(r) dr. \quad (4.56)$$

Then

$$-\int_{t-\hat{\tau}(x,t)}^t \tilde{U}_\infty(r) dr = \int_{t-\tau(x,t)}^t U_\infty(r) dr - \int_{t-\hat{\tau}(x,t)}^t U_\infty(r) dr, \quad (4.57)$$

which implies

$$-\int_{t-\hat{\tau}(x,t)}^t \tilde{U}_\infty(r) dr = \int_{t-\tau(x,t)}^{t-\hat{\tau}(x,t)} U_\infty(r) dr, \quad (4.58)$$

and thus

$$\left| \int_{t-\hat{\tau}(x,t)}^t \tilde{U}_\infty(r) dr \right| = \left| \int_{t-\tau(x,t)}^{t-\hat{\tau}(x,t)} U_\infty(r) dr \right|. \quad (4.59)$$

It is clear that

$$\left| \int_{t-\hat{\tau}(x,t)}^t \tilde{U}_\infty(r) dr \right| \leq \hat{\tau}(x,t) \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(t+s)|, \quad (4.60)$$

$$\left| \int_{t-\tau(x,t)}^{t-\hat{\tau}(x,t)} U_\infty(r) dr \right| \geq |\tau(x,t) - \hat{\tau}(x,t)| \hat{U}_m. \quad (4.61)$$

Therefore, we have

$$|\tau(x,t) - \hat{\tau}(x,t)| \leq \frac{\hat{\tau}(x,t)}{\hat{U}_m} \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(t+s)|. \quad (4.62)$$

Hence, combining (4.54), (4.55), and (4.62), we can bound I_2 as:

$$I_2 \leq \frac{1}{W_n(L)} \int_0^L W_n(\xi) |\tau(L-\xi, t) - \hat{\tau}(L-\xi, t)| M_n(\xi) \zeta d\xi \quad (4.63)$$

$$\leq \frac{\zeta}{W_n(L) \hat{U}_m} \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(t+s)| \int_0^L W_n(\xi) \hat{\tau}(L-\xi, t) M_n(\xi) d\xi. \quad (4.64)$$

Finally, using the fact that $\hat{U}_\infty(t) \geq \hat{U}_m$ for all t implies $\hat{\tau}(L-\xi, t) \leq (L-\xi)/\hat{U}_m$ for all t , we have

$$\begin{aligned} I_2 &\leq \frac{\zeta}{W_n(L) \hat{U}_m^2} \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(t+s)| \int_0^L W_n(\xi) (L-\xi) M_n(\xi) d\xi \\ &= \frac{\beta_n \zeta}{\hat{U}_m^2} \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(t+s)|, \end{aligned} \quad (4.65)$$

with β_n defined in (4.38).

Gathering (4.51), (4.53), and (4.65), we have

$$|\delta \tilde{u}_n(t)| \leq \left(\alpha_n + \frac{\beta_n \zeta}{\hat{U}_m^2} \right) \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(t+s)|, \quad (4.66)$$

which implies

$$|\delta \tilde{u}(t)| \leq \hat{Z} \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(t+s)|, \quad (4.67)$$

with

$$\hat{Z} \triangleq \sum_{n=1}^N \left(\alpha_n + \frac{\zeta \beta_n}{\hat{U}_m^2} \right). \quad (4.68)$$

Notice that, if Z defined in (4.39) is less than 1, as assumed, there is some choice of $\hat{U}_m \in (0, U_m)$ such that $\hat{Z} < 1$. In fact, we need, and assume,

$$\hat{U}_m \in \left(\sqrt{\frac{\zeta \sum_{n=1}^N \beta_n}{1 - \sum_{n=1}^N \alpha_n}}, U_m \right). \quad (4.69)$$

Combining (4.47) and (4.67), we see that

$$\dot{\mathcal{V}}(t) \leq |\tilde{U}_\infty(t)| \zeta - k \tilde{U}_\infty^2(t) + k \hat{Z} \sup_{s \in [-\bar{\tau}, 0]} \tilde{U}_\infty^2(t+s) \quad (4.70)$$

$$\leq \sqrt{2\mathcal{V}(t)} \zeta - 2k \mathcal{V}(t) + 2k \hat{Z} \sup_{s \in [-\bar{\tau}, 0]} \mathcal{V}(t+s). \quad (4.71)$$

From Young's inequality, we have

$$\sqrt{2\mathcal{V}(t)} \zeta \leq \frac{\epsilon \zeta^2}{2} + \frac{\mathcal{V}(t)}{\epsilon} \quad (4.72)$$

for any $\epsilon > 0$. Defining $\rho = 2k\epsilon > 0$, we have

$$\dot{\mathcal{V}}(t) \leq \frac{\rho \zeta^2}{4k} - 2k \left(1 - \frac{1}{\rho} \right) \mathcal{V}(t) + 2k \hat{Z} \sup_{s \in [-\bar{\tau}, 0]} \mathcal{V}(t+s). \quad (4.73)$$

Let us choose a value of $\rho > 0$ such that $\hat{Z} < 1 - 1/\rho$.⁷ Then, we can apply a generalization of Halanay's inequality [129] to show that

$$\mathcal{V}(t) \leq \sup_{s \in [-\bar{\tau}, 0]} \mathcal{V}(s) e^{-\sigma_1 t} + \frac{\rho \zeta^2}{8k^2 \left(1 - \frac{1}{\rho} - \hat{Z} \right)}, \quad (4.74)$$

⁷Such ρ exists because (4.69) holds.

with $\sigma_1 > 0$ satisfying

$$\sigma_1 - 2k \left(1 - \frac{1}{\rho}\right) + 2k\hat{Z}e^{\sigma_1\bar{\tau}} = 0. \quad (4.75)$$

Finally, using the fact that $\sqrt{a^2 + b^2} \leq |a| + |b|$, we have

$$|\tilde{U}_\infty(t)| < \sqrt{2 \sup_{s \in [-\bar{\tau}, 0]} \mathcal{V}(s)e^{-\sigma_1 t} + \frac{\rho\zeta^2}{4k^2 \left(1 - \frac{1}{\rho} - \hat{Z}\right)}} \quad (4.76)$$

$$< \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)|e^{-\sigma t} + \frac{\sqrt{\rho}\zeta}{2k\sqrt{1 - \frac{1}{\rho} - \hat{Z}}}, \quad (4.77)$$

where $\sigma = \sigma_1/2$. We minimize the constant term of this upper bound by picking $\rho = 2/(1 - \hat{Z})$, which yields (4.42).

To prove that $\hat{U}_\infty(t)$ remains lower-bounded, consider

$$k_* \triangleq \frac{\zeta/(1 - \hat{Z})}{U_m - \hat{U}_m - \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)|}. \quad (4.78)$$

Notice $k_* > 0$ because $\hat{U}_m$ satisfies (4.69), so $\hat{Z} < 1$, and we assume in the statement of the theorem that $\sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)| < U_m - \hat{U}_m$. Notice

$$\hat{U}_\infty(0) = U_\infty(0) - \tilde{U}_\infty(0) \geq U_m - \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)| > \hat{U}_m. \quad (4.79)$$

By way of contradiction, suppose there exists $t_1 > 0$ such that $\hat{U}_\infty(t_1) < \hat{U}_m$. Since $\hat{U}_\infty$ is continuous, there must be a $t_2 \in (0, t_1)$ such that $\hat{U}_\infty(t) > \hat{U}_m$ for all $t \in [0, t_2)$ and $\hat{U}_\infty(t_2) = \hat{U}_m$. Using (4.42) with $t = t_2$ and $k > k_*$, we have⁸

$$|\tilde{U}_\infty(t_2)| < \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)|e^{-\sigma t_2} + \frac{\zeta}{k_* \left(1 - \hat{Z}\right)} \quad (4.80)$$

$$\leq \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)|e^{-\sigma t_2} + U_m - \hat{U}_m - \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)| \quad (4.81)$$

$$\leq U_m - \hat{U}_m - \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)| (1 - e^{-\sigma t_2}). \quad (4.82)$$

⁸Notice (4.42) holds because we know a priori that $\hat{U}_\infty(t)$ is lower-bounded by $\hat{U}_m$ for $t \leq t_2$.

Then,

$$\hat{U}_\infty(t_2) = U_\infty(t_2) - \tilde{U}_\infty(t_2) \quad (4.83)$$

$$\geq U_m - |\tilde{U}_\infty(t_2)| \quad (4.84)$$

$$> U_m - \left[U_m - \hat{U}_m - \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)| (1 - e^{-\sigma t_2}) \right] \quad (4.85)$$

$$> \hat{U}_m + \sup_{s \in [-\bar{\tau}, 0]} |\tilde{U}_\infty(s)| (1 - e^{-\sigma t_2}) \quad (4.86)$$

$$> \hat{U}_m, \quad (4.87)$$

i.e., $\hat{U}_\infty(t_2) > \hat{U}_m$. However, t_2 was defined so that $\hat{U}_\infty(t_2) = \hat{U}_m$, so this is a contradiction and, therefore, we cannot have a $t_1 > 0$ such that $\hat{U}_\infty(t_1) < \hat{U}_m$. This concludes the proof.

4.5 Free-flow speed estimation

4.5.1 Application of Theorem 4.1 to Shapiro's model

We now detail how the previous generic result applies to Shapiro's model. When we replace S_n and w_n with the expressions given in (4.4) and (4.3), respectively, we have that Assumption 4.1 is trivially satisfied with

$$M_n(x) = \frac{2a_n}{d_n^2(x)} G_n(x). \quad (4.88)$$

Furthermore, we have, from (4.11),

$$W_n(x) = \exp \left(\int_0^x w_n(\xi) d\xi \right) = \exp \left(\int_0^x \frac{2d'_n(\xi)}{d_n(\xi)} d\xi \right) \quad (4.89)$$

$$= \exp \left(2 \ln \left(\frac{d_n(x)}{d_n(0)} \right) \right) = \frac{d_n^2(x)}{d_n^2(0)}. \quad (4.90)$$

Hence, the expressions for δu_n in (4.13)–(4.14) reduce to

$$\delta u_n(x, t) = \frac{2a_n}{d_n^2(x)} \int_0^x G_n(\xi) U_\infty(t - \tau(x - \xi, t)) d\xi \quad (4.91)$$

$$= \frac{2a_n}{d_n^2(x)} \int_{t-\tau(x, t)}^t G_n(x - \Lambda(s, t)) U_\infty^2(s) ds \quad (4.92)$$

and the expressions for α_n and β_n in Assumption 4.3 become

$$\alpha_n = \frac{2a_n}{d_n^2(L)} \int_0^L G_n(\xi) d\xi, \quad (4.93)$$

$$\beta_n = \frac{2a_n}{d_n^2(L)} \int_0^L (L - \xi) G_n(\xi) d\xi. \quad (4.94)$$

Notice α_n and β_n depend on the position of the n th turbine and its induction factor, the position where the measurement is taken, and on κ_n , which is a coefficient that in reality depends, e.g., on the terrain where the wind farm is. We will give numerical values of α_n and β_n in Section 4.6, when presenting numerical results. For a typical case, such as the ones considered later in Section 4.6, one has $\alpha_n \approx 0.4$. Hence, the first requirement of Assumption 4.2 can be satisfied for one or two turbines, typically, for small enough wind variations. This point is discussed further in Section 4.6.

4.5.2 Estimation strategy implementation

Here, we detail the implementation of the proposed method. We propose to use a first-order discretization with a fixed time-step of the time-integral formulation of the solution to (4.1)–(4.2), which is (4.14). We use this form of the solution rather than the space-integral formulation (4.13) as it is easier to implement since computing the speed deficit with (4.13) requires computing the transport delay for each iteration of the integration loop. This is avoided in this case as explained below.

Algorithm 1 shows the pseudo-code for the implementation. We use square brackets to denote the arguments in discrete time, so, for instance, $y[i] = y(i\Delta t)$, where Δt is the time-step. Recall that we assume the free-flow wind speed estimate is constant before the estimation procedure starts running, i.e., if $j \leq 0$, we use $\hat{U}_\infty[j] = \hat{U}_\infty(0)$.

Notice that the estimation procedure might use a time step smaller than the time step used to obtain the measurements. This may be a feature of interest to mitigate the steady-state error as, according to Theorem 4.1, this may require using a high value for the gain k , which, in general, requires using a small Δt . In this case, we simply interpolate the measurements using a zero-order holder. Also, we include a saturation in the update law implementation which corresponds to the Proj operator used in (4.34).

Moreover, note that the “while loop” is used to perform the integral operation in (4.14). The condition $\hat{\Lambda} < L$ is used because $\Lambda(t - \tau(L, t), t) = L$, thus, we can stop the integration at the correct bound without computing the value of transport delay in advance.

Most of the computational cost comes from the computation of the speed deficits. If the free-flow wind speed estimate is nearly constant, we can roughly say that the computational time for each iteration of the outermost loop is $O(N/\Delta t)$.

In terms of memory usage, we need to store the values $\hat{U}_\infty$ in between iterations of this outermost loop. Since $\hat{U}_\infty \geq \hat{U}_m$, the number of iterations of the “while loop” is limited and we need at most $\lfloor L/(\hat{U}_m \Delta t) \rfloor + 1$ past values of $\hat{U}_\infty$.

Algorithm 1 Implementation of the proposed method

```

for  $i = 0, 1, 2, \dots$  do
   $j \leftarrow i, \hat{\Lambda} \leftarrow 0$ 
  for  $n = 1, \dots, N$  do
     $\delta \hat{u}_n[i] \leftarrow 0$ 
  end for
  while  $\hat{\Lambda} < L$  do
    for  $n = 1, \dots, N$  do
       $\delta \hat{u}_n[i] \leftarrow \delta \hat{u}_n[i] + \frac{1}{W_n(L)} W_n(L - \hat{\Lambda}) S_n(L - \hat{\Lambda}, \hat{U}_\infty[j]) \Delta t$ 
    end for
     $\hat{\Lambda} \leftarrow \hat{\Lambda} + \hat{U}_\infty[j] \Delta t$ 
     $j \leftarrow j - 1$ 
  end while
   $\hat{y}[i] \leftarrow \hat{U}_\infty[i] - \delta \hat{u}_1[i] - \dots - \delta \hat{u}_N[i]$ 
   $\hat{U}_\infty[i + 1] \leftarrow \max \left\{ \hat{U}_\infty[i] + k(y[i] - \hat{y}[i]) \Delta t, \hat{U}_m \right\}$ 
end for

```

4.6 Numerical results

4.6.1 Cases with no modeling error and no noise

First, to better illustrate the essence and merits of the proposed methodology, we present simulations performed with Shapiro's model as a reference model, so there is no modeling error. We consider an example of one turbine creating the wake, that is we have $N = 1$, with the physical parameters $a_1 = 0.25$, $D = 126$ m, $\kappa_1 = 0.05$, $x_1 = 5D$, and $L = 7D$.

Recall that the upper-bound for the error error in Theorem 4.1 consists of a term that decays to zero exponentially plus a term that is a increasing function of ζ (the upper-bound of $\dot{U}_\infty$) and inversely proportional to the gain k . Thus, we expect to have worst steady-state errors with large $\dot{U}_\infty$ and small k . To test this, we used free-flow wind speeds given by $U_\infty(t) = 10 + \sin(2\pi ft)$ for different frequencies f and different gains k . Notice we can consider $\zeta = 2\pi f$ in these cases. The results are shown in Figs. 4.3 and 4.4.

Indeed, we see that the steady-state error remains confined to a smaller region for smaller values of f and larger values of k . In particular, the error converges to zero when $f = 0$ Hz (and, therefore, $\zeta = 0$), which also agrees with Theorem 4.1. We also see in Fig. 4.4 that larger values of k yield a faster transient response, we expect that from the proof of the theorem; notice that the term multiplying $\mathcal{V}$ in (4.73) is proportional to k , so the decay rate is increasing with k .

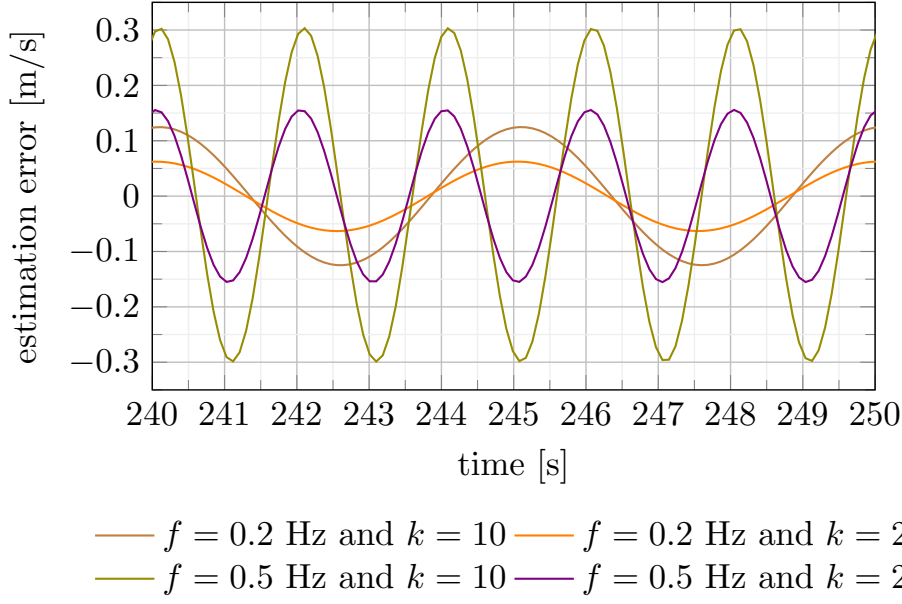


Figure 4.3: Steady-state behavior of the estimation error $U_\infty(t) - \hat{U}_\infty(t)$ for $U_\infty(t) = 10 + \sin(2\pi ft)$ for different frequencies f and different gains k . The initial guess is $\hat{U}_\infty(0) = 15 \text{ m/s}$ in these cases.

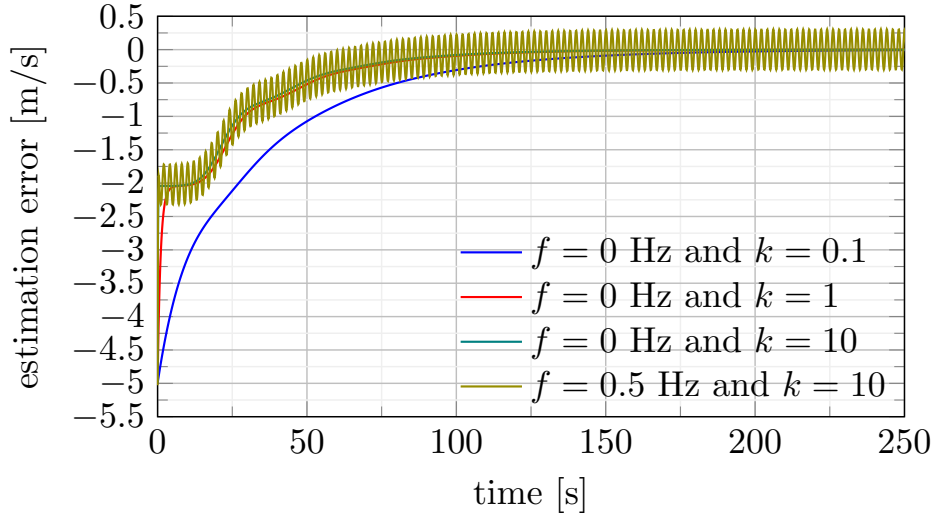


Figure 4.4: Time-evolution of the estimation error $U_\infty(t) - \hat{U}_\infty(t)$ for $U_\infty(t) = 10 + \sin(2\pi ft)$ for different frequencies f and different gains k . The initial guess is $\hat{U}_\infty(0) = 15 \text{ m/s}$ in these cases.

Fig. 4.5 presents a case with the same parameters as before, but with more realistic free-flow condition, namely, with a free-flow wind speed obtained experimentally with a LiDAR sensor.⁹ One can observe that the free flow wind speed is then well-estimated, after a certain transient which could be tuned with a different gain values, but at the expense of potential noise amplification.

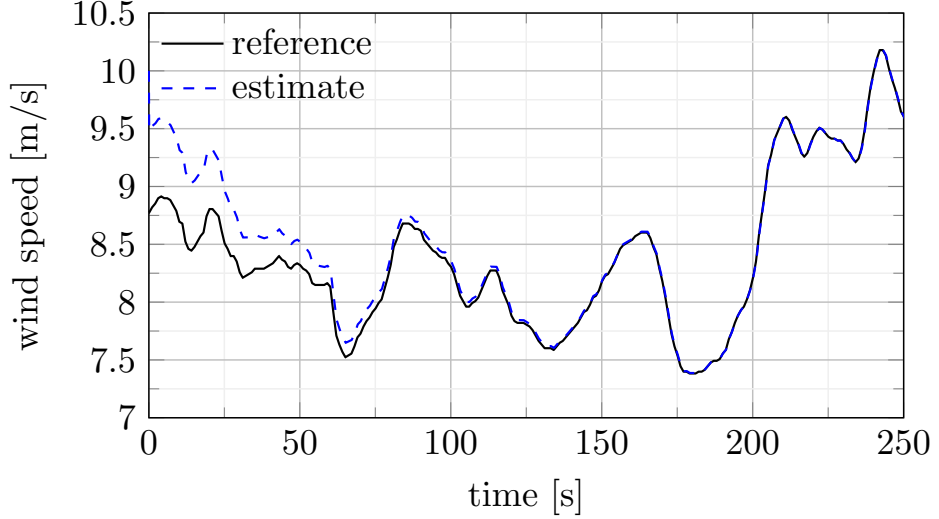


Figure 4.5: Free-flow wind speed estimation for the case $N = 1$ with no modeling error and no measurement noise for an experimental free-flow condition. We used $k = 10$ and $\hat{U}_\infty(0) = 10$ m/s.

The parameters used here yield $\alpha_1 = 0.4085$ and $\beta_1 = 102.9412$ m. In this case, we need the values of ζ and U_m to be in the shaded region depicted in Fig. 4.6 so that $Z < 1$ and all the assumptions of the theorem can be satisfied. For the case in Fig. 4.5, we can use $U_m = 7$ m/s and $\zeta = 0.25$ m/s², which fall within the area depicted in Fig. 4.6. In some of the previous cases, however, we do not meet this condition and the estimation still works, which shows that Theorem 4.1 is conservative.

4.6.2 Cases with modeling error and measurement noise

To assess the performance of the method in a more realistic scenario, we tested it using NREL’s FAST.Farm simulator [66, 67] to play the role of the actual wind farm and provide us with the speed measurement.

We use a configuration where we have an array of three turbines evenly spaced $5D$ apart and the first turbine is at $x_1 = 5D$. We tested two cases.

⁹The reference wind speed was obtained from LiDAR data provided by Leosphere within the framework of the project “SmartEole” of the French National Research Agency (ANR). These are the same data we used in the previous chapter.

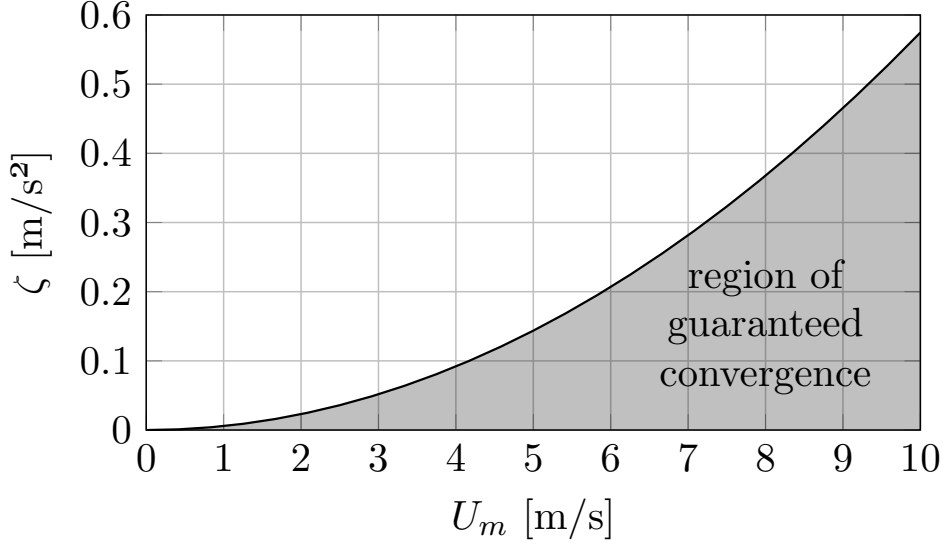


Figure 4.6: Region of guaranteed convergence for the choice of parameters used in Subsection 4.6.1.

In the first one, we consider we can measure the effective wind speed for the second turbine, so $N = 1$ and $L = 10D$. In the second case, we measure the effective wind speed of the third turbine, so $N = 2$ and $L = 15D$. In all simulations, the turbines used are NREL's 5 MW turbine model with a rotor diameter of $D = 126$ m [65]. The values of the coefficients κ_n and the induction factors were tuned on preliminary tests with known constant free-flow wind speed to calibrate Shapiro's wake model with the one provided by FAST.Farm for this specific wind farm. The parameters selected were $a_1 = 0.27$, $a_2 = 0.32$, $\kappa_1 = 0.03$, and $\kappa_2 = 0.15$.

We fed to FAST.Farm the free-flow wind speed obtained from experimental field data used in the previous subsection in Fig. 4.5 and in the previous chapter.

Fig. 4.7 displays the estimation results in both the $N = 1$ and $N = 2$ cases. We used $k = 10$ and $\hat{U}_\infty(0) = 10$ m/s in both cases. The errors $U_\infty - \hat{U}_\infty$ for these same cases are depicted in Fig. 4.8.

Fig. 4.9 depicts the actual local measurement provided in the $N = 2$ case compared to one estimated by the model. We see that the measurement estimate converges very fast to the actual measurement. Also Fig. 4.10 depicts the speed deficit for this same case (which is the difference between the free-flow wind speed and the measurement). We can observe that there is a high-frequency component in the deficit. The big jump in the deficit shortly after time 100 seconds is due to the propagation of the wake, at this time, the wake from the first turbine hits the turbine that is measuring the wind speed.

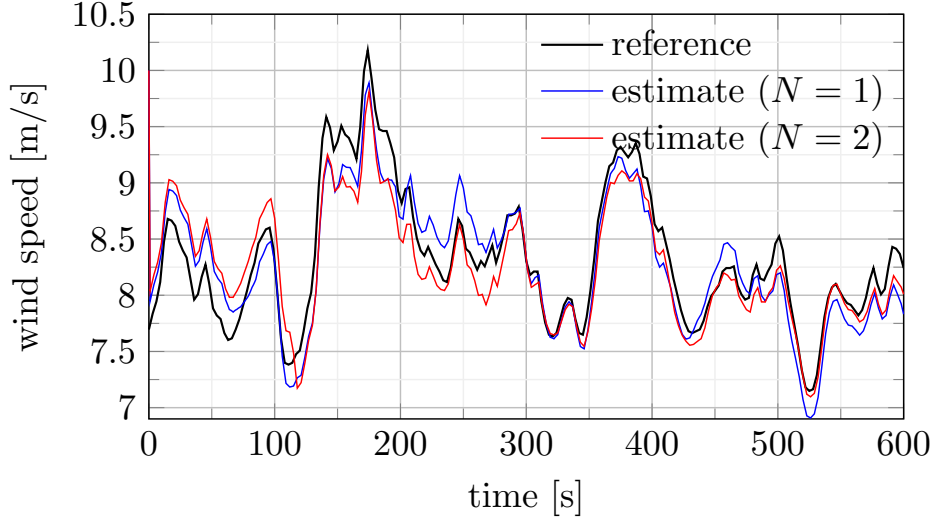


Figure 4.7: Free-flow wind speed estimation results using the local measurement of the second ($N = 1$ case) and, respectively, third ($N = 2$ case) turbine for a window of 10 minutes. The reference for the free-flow wind speed comes from experimental data. The gain is $k = 10$ and $\hat{U}_{\infty}(0) = 10$ m/s.

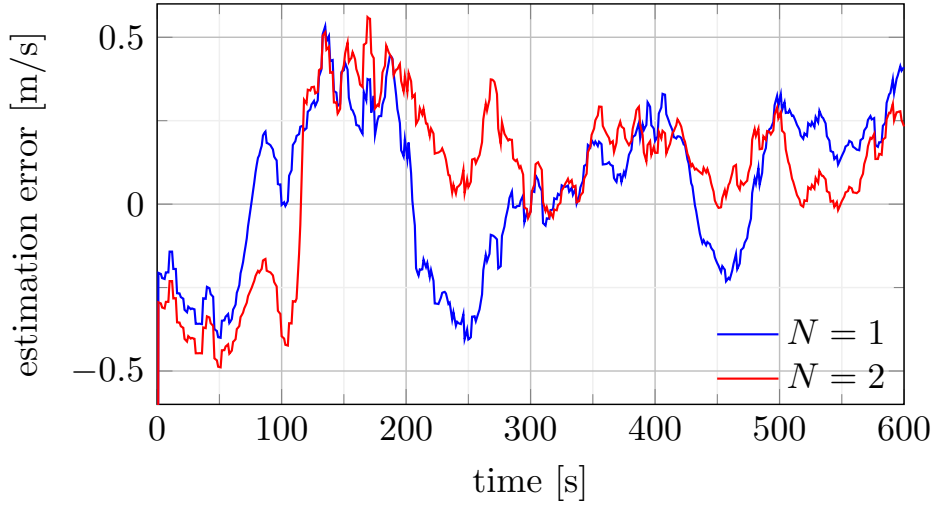


Figure 4.8: Free-flow wind speed estimation error for the cases presented in Fig. 4.7.

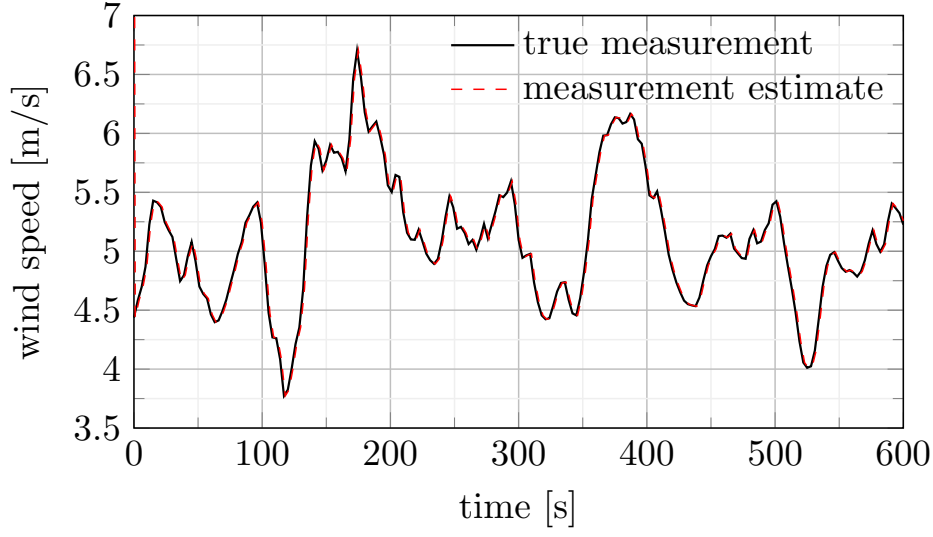


Figure 4.9: Estimation of the local speed measurement in the $N = 2$ case corresponding to Fig. 4.7.

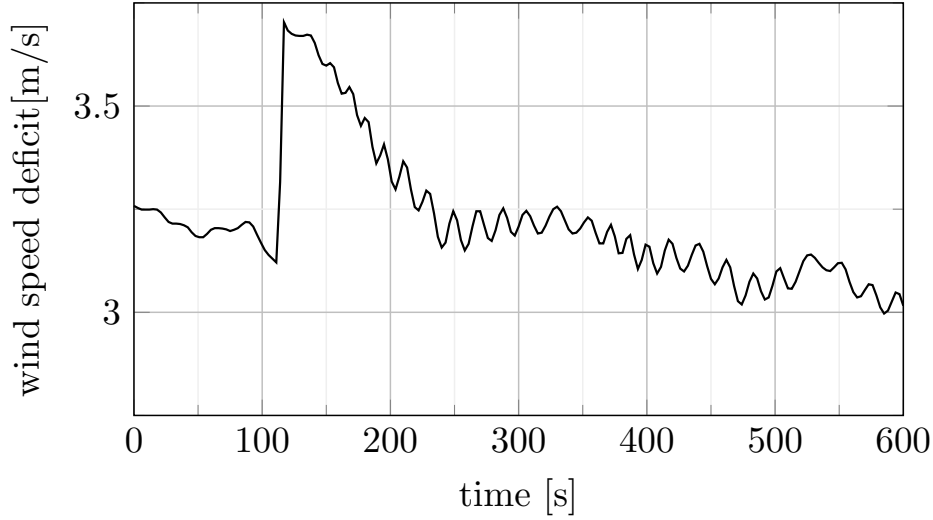


Figure 4.10: Speed deficit in the $N = 2$ case corresponding to Fig. 4.7.

In Fig. 4.11, we use the same data in the $N = 2$ case, but we add noise to the measurements and show the effect of using different values for the gain k . We used $\hat{U}_\infty(0) = 10$ m/s as before. At each measurement, the added noise is sampled from a zero-mean Gaussian distribution with a standard deviation equal to 2% of the true value of the measurement. One can observe some degradation of the performance, as one would expect, but the estimates still exhibit a similar trend to the actual wind speed. We

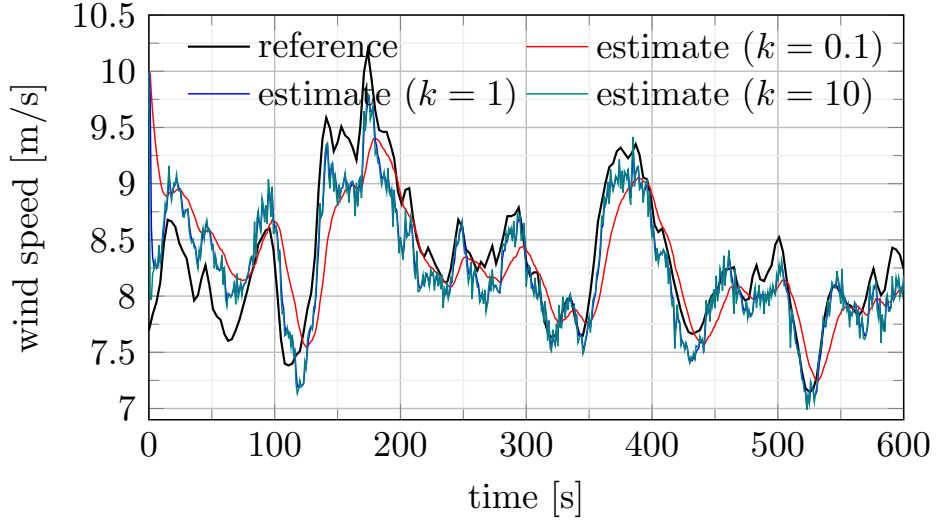


Figure 4.11: Free-flow wind speed estimation results using noisy local measurements of the third turbine ($N = 2$) for different gains. We used $\hat{U}_\infty(0) = 10$ m/s.

also notice that the effect of the noise is less pronounced when a small gain ($k = 0.1$) is used, as one would expect for a technique that relies on output feedback, but at the expense of the time response.

Notice that, as shown in Fig. 4.9, the estimated local measurements are the same as the actual ones, which indicates the difference between the estimated free-flow speed and the actual one is due to the modeling error between Shapiro's model and FAST.Farm.

4.7 Conclusion

This chapter presented a strategy to estimate the free-flow wind speed acting on a array of turbines aligned with the wind direction, when the only speed measurement available is inside the wake zone. This information could then be used to provide an estimate of the distributed wind profile inside the wind farm, and, more specifically, locally for each turbine, from the analytical solution to Shapiro's model.

We managed to prove a powerful convergence theorem and the results obtained were considered satisfactory given the simplicity of both the model used and the applied technique, with a reduced computational cost making it real-time compliant.

From a theoretical perspective, directions of future research include the relaxation of the assumptions for the convergence result we obtained to hold, as our simulations underline they may not be necessary, or to extend this

approach with a time-varying gain, for instance.

Yet, the geometry of the array of wind turbines under consideration here is quite specific. However, considering the wind direction as known, it seems highly likely that the proposed estimation strategy could be generalized to handle unaligned turbines, grounding on an extension of Shapiro's model to two dimensions, such as the one presented in the following chapter. A similar convergence result could then hold. This constitutes a potential direction of future works.

Nevertheless, in all cases, this technique should rely on knowing the wind direction, which could be quite uncertain as well. For this reason, the next chapter considers the problem of combined free-flow wind speed and wind direction estimation.

Chapter 5

Free-flow wind speed and direction estimation in a wind farm from local wind speed measurements

Résumé. Dans ce chapitre, nous présentons une méthode pour estimer conjointement la vitesse et la direction du vent en écoulement libre à partir de mesures de vitesse locale des turbines d'un parc éolien dont la géométrie peut être arbitraire a priori. Cette méthode s'appuie sur un modèle simplifié du sillage en deux dimensions [120], basé sur des cascades d'équations différentielles partielles hyperboliques du premier ordre, dont la vitesse de transport est la vitesse en écoulement libre du vent. Nous formulons le problème d'identification correspondant et proposons une méthode de résolution adaptée. Nous illustrons l'intérêt de cette méthode à l'aide de données numériques issues du simulateur FAST.Farm [66, 67] et dressons des perspectives d'utilisation et d'amélioration.

5.1 Introduction

This chapter investigates the combined estimation of free-flow wind speed and wind direction in a wind farm, given local wind speed measurements of each individual turbine. As in the previous chapter, these local REWS values could be obtained from an estimation technique such as the ones presented in Chapter 3 or one that uses a Light Distance And Ranging (LiDAR) sensor. Contrary to Chapter 4, we do not make specific assumptions on the wind farm geometry. However, we assume that, for a given farm geometry, we are able a priori to consider an array of turbines which are only affected by each other wakes. This can be achieved from a very rough estimate of the

wind direction.

We assume the wind direction to be constant during the considered time horizon, while the wind speed is assumed to be time-varying. This assumption originates from the fact that the wind direction tends to vary much more slowly than the wind speed. One can refer to, e.g., [125], where experimental data show the significant free-flow wind direction changes happen in a scale of minutes.

Compared to Chapter 4, the method we propose here to solve this problem relies on a more standard identification methodology. Indeed, based on estimates of the wind speed and wind direction, we propose to build estimates of the outputs given by each turbine, based on a wake model, and to minimize the cost function arising from the differences between the actual REWS values and the estimated ones. Less standard approaches, for instance, based on integrators as in the previous chapter, were investigated but did not yield satisfactory results. To take into account the effect of possible misalignment with the wind direction, we propose to use a two-dimensional extension [120] of Shapiro’s wake model presented in Chapter 4. Similarly to its one-dimensional counterpart, this simple control-oriented model relies on cascaded transport Partial Differential Equations (PDEs), the transport speed of which is the free-flow wind speed, that is, the variable to be estimated. The structure of the cascades involved in this extension is slightly more complex than the original one-dimensional model [119], but can still be analytically solved, using transport delays. This simplifies the resolution of the identification problem at stake. To solve the optimization problem under consideration, we propose to use a direct optimal control approach. Yet, this optimization problem reveals to be particularly challenging as, due to the appearance of transport delays, the size of the decision variable depends on the decision variable values. To circumvent this issue, we propose to simplify the problem and estimate an average value of the free-flow wind speed history before a certain fixed-size time-window, while estimating its time-varying history inside that same time-window.

The rest of this chapter is organized as follows. First, Section 5.2 presents the extension of Shapiro’s wake model in a two-dimensional context along with its analytical solution, through transport delays. Section 5.3 then formulates the identification problem at stake, which is then recast as a direct optimal control problem in Section 5.4, the gradient of which is provided. We then conclude this chapter with numerical simulations obtained with the simulator FAST.Farm.

5.2 Shapiro’s two-dimensional wake model

Here, we use a two-dimensional wake model initially proposed by [120] to describe the wake of a yawed turbine and we refer to it as Shapiro’s 2-

D model. This model extends the one-dimensional wake model used in Chapter 4.

Shapiro’s 2-D model was initially proposed in [120] and has since been applied in studies such as [14, 54, 82]. This model is based on Prandtl’s lifting line theory [90] and assumes the turbines are operating on the region where the produced power is below rated power, the free-flow wind speed is uniform in space, and the wind direction is constant during the time-horizon of interest. It provides a set of simple enough Partial Differential Equations (PDEs), namely, transport PDEs, that can ultimately be used to compute the 2-D wind velocity field at hub-height (the height of the center of the turbines’ rotors) in a wind farm.

5.2.1 Wind farm modeling

Consider a wind farm with N wind turbines of location (X_n, Y_n) ($1 \leq n \leq N$). We compute the wake of each turbine using the model proposed in [120] and some considerations of our own as explained in the following. We assume all turbines are of the same model and have the same hub-height, as the model considers the 2-D velocity field on a horizontal plane at hub-height.

The model consists of a system of three scalar PDEs for each turbine:

$$\begin{aligned} \frac{\partial \delta u_n}{\partial t} + U_\infty(t) \frac{\partial \delta u_n}{\partial x} = & -w_n(x, x_n) U_\infty(t) \delta u_n(x, t) \\ & + 2a_n(\tilde{\gamma}_n) U_\infty^2(t) G(x - x_n), \end{aligned} \quad (5.1)$$

$$\begin{aligned} \frac{\partial \delta v_n}{\partial t} + U_\infty(t) \frac{\partial \delta v_n}{\partial x} = & -w_n(x, x_n) U_\infty(t) \delta v_n(x, t) \\ & + \frac{1}{4} b(\tilde{\gamma}_n) U_\infty^2(t) C_{T,n} G(x - x_n), \end{aligned} \quad (5.2)$$

$$\frac{\partial y_{c,n}}{\partial t} + U_\infty(t) \frac{\partial y_{c,n}}{\partial x} = -\delta v_n(x, t), \quad (5.3)$$

where the index n represents the n th turbine¹, $x \geq 0$ is the streamwise direction, U_∞ is the free-flow speed, which is assumed to be a continuous function of time, $\tilde{\gamma}_n = \gamma_n - \gamma_\infty$ is the misalignment angle between the n th turbine yaw angle γ_n and the streamwise direction γ_∞ , x_n is the n th turbine’s x position (i.e., the position of the center of the turbine’s rotor),

¹Notice that this numbering depends a priori on the geometry of the farm and the wind direction. This point is further discussed later.

and the functions a_n , b , G , and w_n are defined as

$$a_n(\gamma) = \frac{1}{2} \left(1 - \sqrt{1 - C_{T,n} \cos^2(\gamma)} \right), \quad (5.4)$$

$$b(\gamma) = \cos^2(\gamma) \sin(\gamma), \quad (5.5)$$

$$G(x) = \frac{1}{\frac{D}{2} \sqrt{2\pi}} \exp \left(-\frac{x^2}{2(D/2)^2} \right), \quad (5.6)$$

$$w_n(x, x_n) = \frac{2}{d_n(x, x_n)} \frac{\partial d_n}{\partial x}(x, x_n), \quad (5.7)$$

$$d_n(x, x_n) = 1 + \kappa_n \ln \left(1 + \exp \left(\frac{x - x_n - D}{D/2} \right) \right), \quad (5.8)$$

where $C_{T,n} > 0$ is the n th turbine's thrust coefficient, assumed constant, D is the rotor diameter (assumed to be the same for all turbines), and the constant positive parameter κ_n is related to the n th turbine's wake expansion as in Chapter 4.² The Boundary Conditions (BCs) corresponding to (5.1)–(5.3) are

$$\delta u_n(0, t) = 0, \quad (5.9)$$

$$\delta v_n(0, t) = 0, \quad (5.10)$$

$$y_{c,n}(0, t) = y_n, \quad (5.11)$$

where y_n is the y position of the n th turbine (y being a horizontal direction perpendicular to x). These BCs essentially state that the turbine has no influence over the boundary $x = 0$ of the domain, thus, there is an implicit assumption that the x position of each turbine is much larger than zero, i.e., it is far away from the boundary of the domain³.

Since the x direction corresponds to the free-flow direction, we define two coordinate systems as shown in Fig. 5.1. The coordinate system (X, Y) is a known reference frame, while the coordinate system (x, y) corresponds to the wind frame. One can convert between these two systems of coordinates using the relations

$$x = X \cos(\gamma_\infty) + Y \sin(\gamma_\infty), \quad (5.12)$$

$$y = -X \sin(\gamma_\infty) + Y \cos(\gamma_\infty), \quad (5.13)$$

which follow from rotating the (X, Y) frame counterclockwise around the origin by an angle γ_∞ . Notice that the yaw angles of the turbines shown in Fig. 5.1 are then $\gamma_n = \tilde{\gamma}_n + \gamma_\infty$.

²Here, the notation is a bit different from Chapter 4 in that we make x_n an explicit argument of d_n and w_n and we use G instead of G_n ($G_n(x) = G(x - x_n)$). We made these modifications because the x positions of the turbines in the wind frame are unknown in this case and each x_n needs to be estimated, this will be discussed further in the sequel.

³Notice that, originally, in [120], the boundary conditions are at $-\infty$ and the turbine is placed at $x = 0$. We modified this convention to unify this modeling with the one used in the previous chapter.

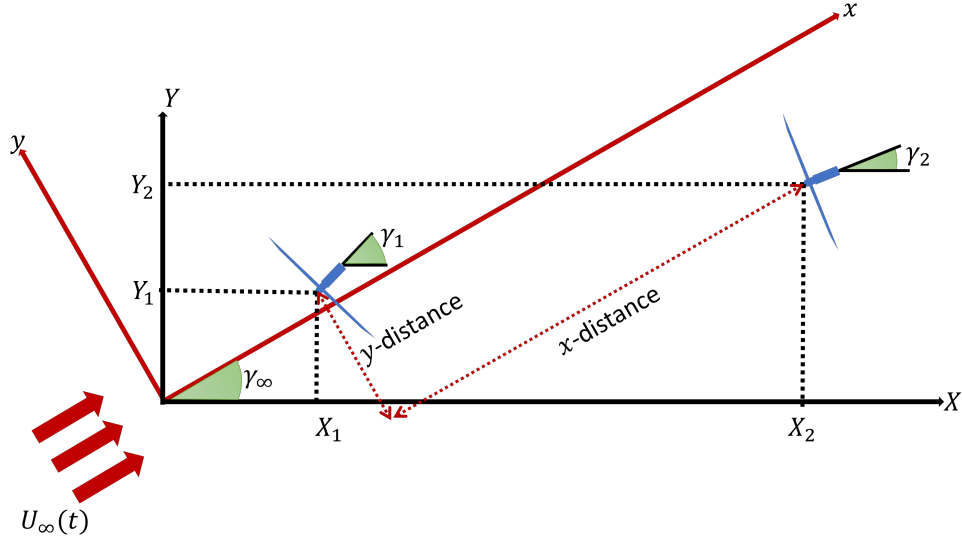


Figure 5.1: Top-view of a wind farm with $N = 2$ turbines. The system of coordinates (X, Y) is the reference frame, (x, y) is the wind frame, the yaw angles γ_1 and γ_2 and wind direction γ_∞ are measured with respect to the direction X . We also show the x - and y -distances between the turbines, which are important to quantify how much the second turbine is affected by the wake of the first.

Notice the functions G , w_n , and d_n appear as well in the 1-D model described in Chapter 4.⁴ The term $2a_n(\tilde{\gamma}_n)U_\infty^2(t)G(x-x_n)$ models the presence of the turbine as a sink of momentum, and depends on the misalignment angle: when there is no misalignment the turbine extracts maximum momentum from the wind, and when its rotor plane is parallel to the wind direction it extracts none. The term $-w_n(x, x_n)U_\infty(t)\delta u(x, t)$ is related to the wake expansion and the decay of the speed deficit downwind from the turbine, notice that w_n is related to the normalized wake diameter d_n and its spatial derivative. The terms on the right-hand side of the second equation of the model play similar roles. The equation for y_c expresses how much the wind is deflected by the turbine, giving the y position of the centerline, which is an integral of the velocity on the y direction, but with a delay related to the propagation speed.

As in Chapter 4, we assume the speed deficits are added linearly. Furthermore, following [120], the streamwise velocity deficit is distributed using a Gaussian profile. Namely, the streamwise velocity at a point (x, y) at time

⁴Yet, notice the source term $2a_n(\tilde{\gamma}_n)U_\infty^2(t)G(x-x_n)$ is not divided by d_n^2 as in the 1-D model (4.1)–(4.6). This is due to a slight difference in the derivation of the sink terms representing the turbines between references [120] and [119]. As this difference is not significant from a numerical viewpoint, we decided conform our models with, respectively, [120] in this chapter and [119] in Chapter 4. See also Footnote 2.

t is

$$U(x, y, t) = U_\infty(t) - \sum_{n \in T(x, y)} \delta u_n(x, t) \frac{D^2}{8\sigma_n^2} \exp\left(-\frac{(y - y_{c,n}(x, t))^2}{2\sigma_n^2 d_n^2(x, x_n)}\right), \quad (5.14)$$

where $\sigma_n > 0$ is a parameter related to the width of the wake and $T(x, y) \subset \{1, \dots, N\}$ is the set of turbines whose wakes affect the location (x, y) (namely, upwind the location). Regarding the deficit in the y direction, we consider it uniformly distributed according to

$$V(x, y, t) = - \sum_{n \in T(x, y)} \delta v_n(x, t), \quad (5.15)$$

where V is the wind speed in the transverse (y) direction.

Notice Shapiro's paper [120] does not discuss the distribution of the deficit in the y direction. We opted for this simple modeling to ease the implementation, and because the transversal speed deficit is supposed to be less significant than the streamwise one.

Fig. 5.2 shows a simulation result for a single turbine at a certain time instant. The speed shown is $\sqrt{(U_\infty(t) - \delta u_1(x, y, t))^2 + \delta v_1^2(x, y, t)}$. The axis are the x and y directions, recall that the free-flow wind direction is along x . We can clearly see the effect of the turbine in reducing the wind speed downwind itself and deflecting the wake centerline. Notice that is also a deflection of the centerline and a speed deficit upwind the turbine (due to the Gaussian present in the source terms of the PDEs of the model).

We define U_n and V_n to be, respectively, the streamwise and transverse velocities at the position of the n th turbine. We consider, for a given wind direction γ_∞ , that the turbines under consideration are ordered such that turbine n is downwind of turbine j for all $j \in \{1, \dots, n-1\}$, so that $T(x_n, y_n) = \{1, \dots, n-1\}$ in (5.14)–(5.15). Consequently, U_n and V_n are computed using only the wakes of the upwind turbines as

$$U_1(t) = U_\infty(t), \quad (5.16)$$

$$V_1(t) = 0, \quad (5.17)$$

and, for $n \in \{2, \dots, N\}$,

$$U_n(t) = U_\infty(t) - \sum_{j=1}^{n-1} \delta u_j(x_n, t) \frac{D^2}{8\sigma_j^2} \exp\left(-\frac{(y_n - y_{c,j}(x_n, t))^2}{2\sigma_j^2 d_j^2(x_n, x_j)}\right), \quad (5.18)$$

$$V_n(t) = - \sum_{j=1}^{n-1} \delta v_j(x_n, t). \quad (5.19)$$

The REWS measurement for each turbine is considered to be the component of the velocity perpendicular to the turbine plane at the position

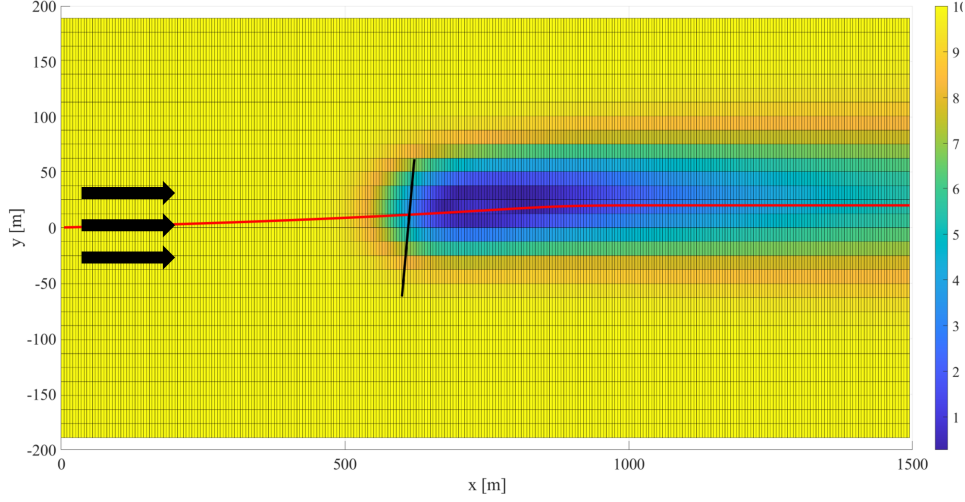


Figure 5.2: Simulation of Shapiro's 2-D model. The wake centerline is shown in red and the turbine is shown in black. The speed at each point in space is represented in meters per second. The arrows show the wind direction.

of the turbine not accounting for induction effects. Let Υ_n be the REWS measurement of turbine n , then

$$\Upsilon_1(t) = U_\infty(t) \cos(\gamma_1 - \gamma_\infty), \quad (5.20)$$

$$\Upsilon_n(t) = \sqrt{U_n^2(t) + V_n^2(t)} \cos\left(\gamma_n - \gamma_\infty - \text{atan}\left(\frac{V_n(t)}{U_n(t)}\right)\right), \quad n > 1, \quad (5.21)$$

These measurements were also not discussed in the reference for the model. We define them in this way so that the wind speed for a certain turbine is characterized by the wind speed at the center of the rotor of that turbine and the arctangent term is used to account for the wind deflection caused by upwind turbines. This model gives simple relations and is a sensible choice according to the obtained results detailed later.

We propose to use this model in the sequel to estimate the free-flow wind speed in a wind farm, based on local wind measurements at each turbine. With this aim in view, we first detail analytical expressions for the solutions of these equations.

5.2.2 Analytical solutions of the PDEs (5.1)–(5.3) with BCs (5.9)–(5.11)

Using the same technique we used to solve the PDE for the 1-D model in Chapter 4 and defining

$$H_n(x, x_n) = d_n^2(x, x_n)G(x - x_n), \quad (5.22)$$

we have, for $n = 1, \dots, N$,

$$\delta u_n(x, t) = \frac{2a_n(\gamma_n - \gamma_\infty)}{d_n^2(x, x_n)} \int_{t-\tau(x, t)}^t H_n(x - \Lambda(s, t), x_n) U_\infty^2(s) ds, \quad (5.23)$$

$$\delta v_n(x, t) = \frac{b(\gamma_n - \gamma_\infty)C_{T, n}}{4d_n^2(x, x_n)} \int_{t-\tau(x, t)}^t H_n(x - \Lambda(s, t), x_n) U_\infty^2(s) ds, \quad (5.24)$$

$$y_{c, n}(x, t) = y_n - \int_{t-\tau(x, t)}^t \delta v_n(x - \Lambda(s, t), s) ds, \quad (5.25)$$

with

$$\Lambda(s, t) = \int_s^t U_\infty(r) dr \quad (5.26)$$

and the transport delay τ defined by

$$\int_{t-\tau(x, t)}^t U_\infty(r) dr = x. \quad (5.27)$$

Observe that, as previously, in Chapter 4, one has $\tau(x, t) \leq x/U_m$ if U_∞ is assumed to be lower-bounded by $U_m > 0$ (we make this assumption here as we did in Chapter 4). Hence, for any $x \in [0, \bar{L}]$ (for a certain upper length $\bar{L}$, which is assumed to be known in the sequel), one has $\tau(x, t) \leq \bar{L}/U_m \triangleq \bar{\tau}$. We will use this upper-bound for the delay in the remaining of this chapter.

Also, notice that the speed deficits and the centerline (5.23)–(5.25) all depend on the past values of the free-flow wind speed U_∞ . In details, for a fixed $(x, t) \in [0, \bar{L}] \times \mathbb{R}_+$, both the streamwise (δu_n) and transverse (δv_n) speed deficits in (5.23) and (5.24) involve values of U_∞ over the interval $[t - \tau(x, t), t]$, which is contained in $[t - \bar{\tau}, t]$. Hence, the centerline y position ($y_{c, n}$) in (5.25) depends on values of U_∞ over the interval $[\min_{s \in [t-\tau(x, t), t]} \{s - \tau(x - \Lambda(s, t), s)\}, t]$, which is contained in $[t - 2\bar{\tau}, t]$.

In the following, we will take advantage of these analytical expressions for the solutions to the PDEs to develop an estimation methodology.

5.3 Estimation through nonlinear least squares

Recall our objective is to estimate the free-flow wind speed and direction from local speed measurements given by the turbines in the wind farm. To do so, we propose to use a standard identification strategy. The first step in this strategy is to provide estimates of the wake with certain guesses for the wind speed U_∞ and the wind direction γ_∞ and compute estimated values for the measurements with those. Then, we can compare these virtual measurements to the actual ones and update our guess of the decision variable

accordingly. We refer to this optimization strategy as a Nonlinear Least Squares (NLS) problem, due to its discrete-time formulation detailed later.

In this section, we present the formulation of the estimation problem at stake, before detailing its corresponding discrete-time counterpart, which can be seen as a direct optimal control problem, in Section 5.4.

5.3.1 Formulation of the estimation problem as a continuous-time optimization problem

As mentioned before, we assume, even though the wind direction is unknown, we are able to order the N turbines under consideration such that, for every $n \in \{2, \dots, N\}$, turbine n is downwind of turbine j for all $j \in \{1, \dots, n-1\}$.

Let $t \geq 0$. With the aim of parameters estimation, consider the following cost function

$$J(t) = \frac{1}{2} \int_{t-\ell\Delta t}^t \sum_{n=1}^N (\Upsilon_n(s) - \hat{\Upsilon}_n(s))^2 ds, \quad (5.28)$$

where Υ_n , defined in (5.20) for $n = 1$ and (5.21) otherwise, is the measurement given by the n th turbine (the REWS), $\ell\Delta t$ is the time-window length ($\ell \in \mathbb{N}$ and $\Delta t > 0$), and $\hat{\Upsilon}_n$ is a measurement estimate⁵, which we detail in the sequel.

As observed in Subsection 5.2.2, for a fixed $t \geq 0$, the analytical solutions to the speed deficits and wake centerline depend on past values of U_∞ over the time interval $[t-2\bar{\tau}, t]$. As the cost function (5.28) involves measurements over the time window $[t-\ell\Delta t, t]$, this implies that $J(t)$ depends on past values of U_∞ over the time horizon $[t-\ell\Delta t-2\bar{\tau}, t]$. To account for these two different intervals in the following analysis, we introduce the following notations.

Notations. In the rest of this chapter, we denote

$$\mathcal{C} = \mathcal{C}([-\ell\Delta t-2\bar{\tau}, 0], [U_m, \infty)), \quad (5.29)$$

$$\bar{\mathcal{C}} = \mathcal{C}([-2\bar{\tau}, 0], [U_m, \infty)) \quad (5.30)$$

the spaces of continuous functions from $[-\ell\Delta t-2\bar{\tau}, 0]$ into $[U_m, \infty)$ and from $[-2\bar{\tau}, 0]$ into $[U_m, \infty)$, respectively. Besides, for a given function $f : \mathbb{R} \mapsto \mathbb{R}$ and $t \in \mathbb{R}$, we denote the corresponding segment functions as

$$f_{[t]} : s \in [-\ell\Delta t-2\bar{\tau}, 0] \mapsto f(t+s), \quad (5.31)$$

$$f_{[\bar{t}]} : s \in [-2\bar{\tau}, 0] \mapsto f(t+s). \quad (5.32)$$

⁵For the sake of simplicity, we denoted $\hat{\Upsilon}_n$ as a function of s in (5.28). Yet, it actually depends on estimates of the wind speed and wind direction, as we will see in the sequel.

With these notations, for $t \geq 0$ and $s \in [t - \ell\Delta t, t]$, $\Upsilon_n(s)$ depends on $U_{\infty, [s]}$ and γ_∞ and, accordingly, $J(t)$ depends on $U_{\infty, [t]}$ and γ_∞ . We are now ready to define our measurement estimate $\hat{\Upsilon}_n$.

Given estimates $\hat{U}_{\infty, [t]} \in \mathcal{C}$ and $\hat{\gamma}_\infty \in \mathbb{R}$ for the segment function $U_{\infty, [t]}$ and the scalar γ_∞ , respectively, we compute the estimates for the measurements as, for $s \in [t - \ell\Delta t, t]$,

$$\hat{\Upsilon}_1(\hat{U}_{\infty, [s]}, \hat{\gamma}_\infty) = \hat{U}_\infty(s) \cos(\gamma_1 - \hat{\gamma}_\infty), \quad (5.33)$$

$$\begin{aligned} \hat{\Upsilon}_n(\hat{U}_{\infty, [s]}, \hat{\gamma}_\infty) &= \sqrt{\hat{U}_n^2(\hat{U}_{\infty, [s]}, \hat{\gamma}_\infty) + \hat{V}_n^2(\hat{U}_{\infty, [s]}, \hat{\gamma}_\infty)} \\ &\times \cos\left(\gamma_n - \hat{\gamma}_\infty - \text{atan}\left(\frac{\hat{V}_n(\hat{U}_{\infty, [s]}, \hat{\gamma}_\infty)}{\hat{U}_n(\hat{U}_{\infty, [s]}, \hat{\gamma}_\infty)}\right)\right), \quad n > 1, \end{aligned} \quad (5.34)$$

where $\hat{U}_n$ and $\hat{V}_n$ are estimates for U_n and V_n computed using (5.18)–(5.19) but with $\hat{U}_\infty$ and $\hat{\gamma}_\infty$ in lieu of the actual parameters U_∞ and γ_∞ , i.e.,

$$\begin{aligned} \hat{U}_n(\hat{U}_{\infty, [s]}, \hat{\gamma}_\infty) &= \hat{U}_\infty(s) - \sum_{j=1}^{n-1} \delta \hat{u}_j(\hat{x}_n(\hat{\gamma}_\infty), \hat{U}_{\infty, [s]}, \hat{\gamma}_\infty) \\ &\times G_a(\hat{y}_n(\hat{\gamma}_\infty), \hat{y}_{c,j}(\hat{x}_n(\hat{\gamma}_\infty)), d_j^2(\hat{x}_n(\hat{\gamma}_\infty), \hat{x}_j(\hat{\gamma}_\infty))), \end{aligned} \quad (5.35)$$

$$\hat{V}_n(\hat{U}_{\infty, [s]}, \hat{\gamma}_\infty) = - \sum_{j=1}^{n-1} \delta \hat{v}_j(\hat{x}_n(\hat{\gamma}_\infty), \hat{U}_{\infty, [s]}, \hat{\gamma}_\infty), \quad (5.36)$$

where

$$G_a(s_1, s_2, s_3) = \frac{D^2}{8\sigma^2} \exp\left(-\frac{(s_1 - s_2)^2}{2\sigma^2 s_3^2}\right), \quad (5.37)$$

$\hat{x}_n$ and $\hat{y}_n$ are the estimates of the position of the n th turbine on the (x, y) coordinates system based on the estimate for the wind direction,

$$\hat{x}_n(\hat{\gamma}_\infty) = X_n \cos(\hat{\gamma}_\infty) + Y_n \sin(\hat{\gamma}_\infty), \quad (5.38)$$

$$\hat{y}_n(\hat{\gamma}_\infty) = -X_n \sin(\hat{\gamma}_\infty) + Y_n \cos(\hat{\gamma}_\infty), \quad (5.39)$$

and $\delta \hat{u}_n$, $\delta \hat{v}_n$, and $\hat{y}_{c,n}$ are obtained⁶ by plugging $\hat{U}_\infty$ and $\hat{\gamma}_\infty$ in place of U_∞

⁶In details, from (5.23)–(5.26), one would first obtain integrals of $\hat{U}_\infty(r)$ for $r \in [s - \tau, s]$. Performing the change of variable $h = r - s$, one can rewrite these integrals to emphasize the dependence with respect to $U_{\infty, [s]}$.

and γ_∞ in (5.23)–(5.26):

$$\begin{aligned}\delta\hat{u}_n(x, \hat{U}_{\infty, [s]}, \hat{\gamma}_\infty) &= \frac{2a_n(\gamma_n - \hat{\gamma}_\infty)}{d_n^2(x, \hat{x}_n(\hat{\gamma}_\infty))} \\ &\times \int_{-\hat{\tau}(x, \hat{U}_{\infty, [s]})}^0 H_n(x - \hat{\Lambda}(h, \hat{U}_{\infty, [s]}), \hat{x}_n(\hat{\gamma}_\infty)) \hat{U}_\infty^2(s+h) dh,\end{aligned}\quad (5.40)$$

$$\begin{aligned}\delta\hat{v}_n(x, \hat{U}_{\infty, [s]}, \hat{\gamma}_\infty) &= \frac{b(\gamma_n - \hat{\gamma}_\infty)C_{T,n}}{4d_n^2(x, \hat{x}_n(\hat{\gamma}_\infty))} \\ &\times \int_{-\hat{\tau}(x, \hat{U}_{\infty, [s]})}^0 H_n(x - \hat{\Lambda}(h, \hat{U}_{\infty, [s]}), \hat{x}_n(\hat{\gamma}_\infty)) \hat{U}_\infty^2(s+h) dh,\end{aligned}\quad (5.41)$$

$$\hat{y}_{c,n}(x, \hat{U}_{\infty, [t]}, \hat{\gamma}_\infty) = \hat{y}_n(\hat{\gamma}_\infty) - \int_{-\hat{\tau}(x, \hat{U}_{\infty, [s]})}^0 \delta\hat{v}_n(x - \hat{\Lambda}(h, \hat{U}_{\infty, [s]}), \hat{U}_{\infty, [s]}, \hat{\gamma}_\infty) dh,\quad (5.42)$$

with, expliciting the dependence with respect to $\hat{U}_{\infty, [s]}$,

$$\hat{\Lambda}(h, \hat{U}_{\infty, [s]}) = \int_{h+s}^s \hat{U}_\infty(r) dr = \int_h^0 \hat{U}_{\infty, [s]}(r) dr,\quad (5.43)$$

and the delay $\hat{\tau}$ is the solution to

$$\int_{s-\hat{\tau}(x, \hat{U}_{\infty, [s]})}^s \hat{U}_\infty(r) dr = x \quad \text{or, equivalently,} \quad \int_{-\hat{\tau}(x, \hat{U}_{\infty, [s]})}^0 \hat{U}_{\infty, [s]}(r) dr = x.\quad (5.44)$$

Finally, for these estimates to be well-defined (more specifically (5.34) which involves a division), we will make the following assumption on the parameters of the system.

Assumption 5.1. *There exists a known non-empty set $\Gamma = [\gamma_m, \gamma_M] \subset \cap_{n=1}^N (\gamma_n - \frac{\pi}{2}, \gamma_n + \frac{\pi}{2})$ such that the function $\hat{U}_n : \bar{\mathcal{C}} \times \Gamma \mapsto \mathbb{R}$ defined in (5.35) takes values into $(0, \infty)$.*

Assuming Γ is a non-empty set is not limiting, in the sense that the turbines will not be perpendicular to the wind direction or backward to it in normal operating conditions and the yaw angles of nearby turbines in a wind farm are generally similar. Hence, even in the case where the wind direction is uncertain, it is reasonable to consider that the yaw angles of the turbines have a similar magnitude and that an interval of length π around each γ_n is thus an interval of interest for $\hat{\gamma}_\infty$. Thence, Γ can be considered as non-empty.

Besides, $\hat{U}_n$ corresponds to the estimate of the streamwise wind speed at the n th turbine, a quantity which is positive. It thus makes sense to assume that it takes positive values of the domain of definition of interest⁷.

The estimation problem under consideration can then be formulated as follows.

Problem 5.1. *Let $t \geq 0$ and $\Gamma = [\gamma_m, \gamma_M]$ given by Assumption 5.1. Solve the optimization problem*

$$\arg \min_{(\hat{U}_{\infty, [t]}, \hat{\gamma}_{\infty}) \in \mathcal{C} \times \Gamma} \left[J(t) = \frac{1}{2} \int_{t-\ell\Delta t}^t \sum_{n=1}^N \left(\Upsilon_n(s) - \hat{\Upsilon}_n(\rho_{s-t}(\hat{U}_{\infty, [t]}), \hat{\gamma}_{\infty}) \right)^2 ds \right], \quad (5.45)$$

where $\hat{\Upsilon}_n$ is defined through (5.33)–(5.44) and $\rho_{s-t}(\hat{U}_{\infty, [t]}) = \hat{U}_{\infty, [t]}$ is the restriction of $\hat{U}_{\infty, [t]}$ to the interval $[s-t-2\bar{\tau}, s-t]$.

Notice this estimation problem involves the infinite-dimensional variable $\hat{U}_{\infty, [t]}$. Identification of an infinite-dimensional variable is known to be a challenging problem. In the context of PDEs, adjoint methods [78] have been widely used to design observers [41, 95] or estimators of distributed parameters [96]. Yet, up to our knowledge, these methods are not tailored to handle time-varying parameters, such as the free-flow wind speed here.

On the other hand, a large number of studies dealing with function estimation, such as [42, 85], reduce the problem to a finite-dimensional one by projecting the target variable on a (finite or truncated) known basis through, e.g., Fourier transforms or wavelets [56, 80, 99] grounding on a certain property of the universal approximation theorem (see [52], in the context of neural networks). The problem then becomes the search of the (finite many) coefficients involved in the projection onto the basis. We will somehow follow the spirit of these finite-dimensional approaches in Section 5.4, by considering the discrete-time equivalent problem and looking for a piecewise constant signal. From an optimization viewpoint, this could be seen as recasting the optimization problem as a direct optimal control problem.

Yet, before delving into numerically solving this problem, one should initially consider whether it is well-posed. Observe that, depending on the value of $U_{\infty, [t]}$, some of its history up to time $t - \ell\Delta t - 2\bar{\tau}$ may not impact the measurements. Namely, for instance, it may be possible that $\min_s U_{\infty, [t]}(s) > U_m$ for all time s , in which case the eldest values of $U_{\infty, [t]}$ do not influence $J(t)$. However, since the values of the delay depend on U_{∞} , which is to be estimated, we do not know a priori how far back we need to go in the history of U_{∞} and we are left with formulating the problem with

⁷Otherwise, one can still further restrict the set of definition of $\hat{\gamma}_{\infty}$ to satisfy this property.

this maximal time range. Thus, in general, there will be values of $\hat{U}_{\infty,[t]}$ that are, therefore, unidentifiable, which leads to multiple solutions for the problem.

We will go back to this discussion in the next section, where we move to the discrete-time setup and this identifiability problem becomes more apparent.

5.4 Discrete-time formulation of the optimization problem

As mentioned before, we move to the discrete-time setup to solve the estimation problem at stake. Assuming a constant time-step size $\Delta t > 0$, we sample the continuous-time variable $\hat{U}_{\infty,[t]}$ as

$$\hat{U}_{\infty} \triangleq [\hat{U}_{\infty}[i - N_U + 1], \dots, \hat{U}_{\infty}[i - 1], \hat{U}_{\infty}[i]]^T \in \mathbb{R}^{N_U}, \quad (5.46)$$

where $N_U = \ell + 2N_{\bar{\tau}} + 1$ and $N_{\bar{\tau}} = \lceil \bar{\tau}/\Delta t \rceil$. Note that $\hat{U}_{\infty}$ depends on i .

One can then discretize the cost function presented in (5.28) and obtain

$$J[i] = J(i\Delta t) = \frac{\Delta t}{2} \sum_{k=i-\ell}^i \sum_{n=1}^N (\Upsilon_n[k] - \hat{\Upsilon}_n(k, \hat{U}_{\infty}, \hat{\gamma}_{\infty}))^2, \quad (5.47)$$

with $i \in \mathbb{N}$ and where $\hat{\Upsilon}_1, \dots, \hat{\Upsilon}_N$ are the discrete-time measurement estimates. These analogs to the continuous-time measurement estimates (5.33)–(5.34) are, for $k \in \{i - \ell, \dots, i\}$,

$$\hat{\Upsilon}_1(k, \hat{U}_{\infty}, \hat{\gamma}_{\infty}) = \hat{U}_{\infty}[k] \cos(\gamma_1 - \hat{\gamma}_{\infty}), \quad (5.48)$$

$$\begin{aligned} \hat{\Upsilon}_n(k, \hat{U}_{\infty}, \hat{\gamma}_{\infty}) &= \sqrt{\hat{U}_n^2(k, \hat{U}_{\infty}, \hat{\gamma}_{\infty}) + \hat{V}_n^2(k, \hat{U}_{\infty}, \hat{\gamma}_{\infty})} \\ &\times \cos \left(\gamma_n - \hat{\gamma}_{\infty} - \text{atan} \left(\frac{\hat{V}_n(k, \hat{U}_{\infty}, \hat{\gamma}_{\infty})}{\hat{U}_n(k, \hat{U}_{\infty}, \hat{\gamma}_{\infty})} \right) \right), \quad n > 1, \end{aligned} \quad (5.49)$$

with the discrete-time velocity estimates $\hat{U}_n$ and $\hat{V}_n$ corresponding to the continuous ones (5.35)–(5.36):

$$\hat{U}_n(k, \hat{U}_{\infty}, \hat{\gamma}_{\infty}) = \hat{U}_{\infty}[k] - \sum_{j=1}^{n-1} \delta \hat{u}_j(\hat{x}_n(\hat{\gamma}_{\infty}), k, \hat{U}_{\infty}, \hat{\gamma}_{\infty}) \quad (5.50)$$

$$\begin{aligned} &\times G_a(\hat{y}_n, \hat{y}_{c,j}(\hat{x}_n(\hat{\gamma}_{\infty}), k, \hat{U}_{\infty}, \hat{\gamma}_{\infty}), d_j^2(\hat{x}_n(\hat{\gamma}_{\infty}), \hat{x}_j(\hat{\gamma}_{\infty}))), \\ \hat{V}_n(k, \hat{U}_{\infty}, \hat{\gamma}_{\infty}) &= - \sum_{j=1}^{n-1} \delta \hat{v}_j(\hat{x}_n(\hat{\gamma}_{\infty}), k, \hat{U}_{\infty}, \hat{\gamma}_{\infty}), \end{aligned} \quad (5.51)$$

in which d_n and G_a are defined in (5.8) and (5.37), respectively, $\hat{x}_n$ and $\hat{y}_n$ are defined in (5.38)–(5.39), and the discrete-time estimates for the speed deficits and the centerline y position corresponding to (5.40)–(5.42) are

$$\begin{aligned} \delta \hat{u}_n(x, k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) &= \frac{2a_n(\gamma_n - \hat{\gamma}_\infty)}{d_n^2(x, \hat{x}_n(\hat{\gamma}_\infty))} \\ &\times \sum_{j=k-n_{\hat{\tau}}(x, k, \hat{\mathbf{U}}_\infty)}^k H_n(x - \hat{\Lambda}_{j,k}(\hat{\mathbf{U}}_\infty), \hat{x}_n(\hat{\gamma}_\infty)) \hat{U}_\infty^2[j] \Delta t, \end{aligned} \quad (5.52)$$

$$\begin{aligned} \delta \hat{v}_n(x, k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) &= \frac{b(\gamma_n - \hat{\gamma}_\infty) C_{T,n}}{4d_n^2(x, \hat{x}_n(\hat{\gamma}_\infty))} \\ &\times \sum_{j=k-n_{\hat{\tau}}(x, k, \hat{\mathbf{U}}_\infty)}^k H_n(x - \hat{\Lambda}_{j,k}(\hat{\mathbf{U}}_\infty), \hat{x}_n(\hat{\gamma}_\infty)) \hat{U}_\infty^2[j] \Delta t, \end{aligned} \quad (5.53)$$

$$\hat{y}_{c,n}(x, k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) = \hat{y}_n(\hat{\gamma}_\infty) - \sum_{j=k-n_{\hat{\tau}}(x, k, \hat{\mathbf{U}}_\infty)}^k \delta \hat{v}_n(x - \hat{\Lambda}_{j,k}(\hat{\mathbf{U}}_\infty), k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \Delta t, \quad (5.54)$$

where a_n , b , and H_n are as defined in (5.4), (5.5), and (5.22), respectively, and

$$\hat{\Lambda}_{j,k}(\hat{\mathbf{U}}_\infty) = \sum_{p=j+1}^k \hat{U}_\infty[p] \Delta t, \quad (5.55)$$

$$n_{\hat{\tau}}(x, k, \hat{\mathbf{U}}_\infty) = \min \left\{ j \in \mathbb{N} \mid \sum_{p=k-j}^k \hat{U}_\infty[p] \Delta t \geq x \right\}. \quad (5.56)$$

Notice $\hat{\Lambda}_{j,k}(\hat{\mathbf{U}}_\infty)$ and $n_{\hat{\tau}}(x, k, \hat{\mathbf{U}}_\infty)$ are the discrete-time analogs of the functions $\hat{\Lambda}(h, \hat{U}_{\infty, [s]})$ and $\hat{\tau}(x, \hat{U}_{\infty, [s]})$ presented in the continuous-time setting (see (5.43) and (5.44)).

It is easier to see the identifiability issue mentioned before for Problem (5.1) in this discrete-time formulation. Indeed, observe that some of the coordinates of $\hat{\mathbf{U}}_\infty$ do not influence $\hat{\mathbf{Y}}_n(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)$, and thus the cost J , if the delays are shorter than $\bar{\tau}$, that is, if $\hat{\mathbf{U}}_\infty$ exhibit values larger than its lower bound U_m on the time window of interest. This makes the problem unidentifiable and thus ill-posed, in the sense that it admits an infinite number of solutions.

To render the problem solvable, we propose to impose an addition condition on the wind speed history, which is

$$\hat{\mathbf{U}}_\infty[k] = \hat{U}_\infty[i - N_U + k] = \hat{U}_0, \quad 1 \leq k \leq 2N_{\bar{\tau}}, \quad (5.57)$$

for some $\hat{U}_0$ to be identified. This $\hat{U}_0$ can be interpreted as an average value of the speed history over the time interval $[t - 2\bar{\tau} - \ell\Delta t, t - \ell\Delta t]$. This presents the additional advantage of reducing the number of variables under consideration. Notice that, intuitively, this trick is a priori only possible when considering a large enough number of measurements ℓ . Indeed, the influence of the free-flow wind speed history only vanishes after a certain time. Thus, ℓ should be considered large enough in the following.

Finally, we also add bounds for the speed and the direction estimates, so we have the constraints

$$\hat{U}_\infty \in [U_m, U_M]^{N_U}, \quad (5.58)$$

$$\hat{\gamma}_\infty \in [\gamma_m, \gamma_M], \quad (5.59)$$

where U_m , U_M , γ_m , and γ_M are constant scalar parameters chosen by the user that should represent physically reasonable values for the bounds of the wind speed and the wind direction; these bounds for the wind direction could also be obtained from wind vane sensors. We assume they satisfy Assumption 5.1.

Now, we can rewrite the optimization problem in discrete-time as follows.

Problem 5.2. *Consider $i \in \mathbb{N}$. Find the solution to*

$$\arg \min_{(\hat{U}_\infty, \hat{\gamma}_\infty, \hat{U}_0) \in \mathbb{R}^{N_U+2}} \left[J[i] = \frac{\Delta t}{2} \sum_{k=i-\ell}^i \sum_{n=1}^N \left(\Upsilon_n[k] - \hat{\Upsilon}_n(k, \hat{U}_\infty, \hat{\gamma}_\infty) \right)^2 \right] \quad (5.60)$$

subject to (5.48)–(5.59).

To numerically implement the solution of the optimization problem at stake, it is more convenient, for the sake of numerical efficiency, to have the analytical expression of the gradient of the cost function under consideration. We provide it for the specific case $N = 2$ next.

Gradient of the cost function for $N = 2$ Here, we present the following lemma with the analytical expression of the gradient of the cost function J in (5.47) in the case where the number of turbines is $N = 2$. These expressions can be straightforwardly extended to any N .

Lemma 5.1. *Denote by $\mathbf{e}_i$ the canonical basis of $\mathbb{R}^{N_U}$ and by $\mathbf{0}_{N_U}$ its zero vector. Consider the cost function J in (5.47) defined in terms of the estimates $\hat{\Upsilon}_1, \hat{\Upsilon}_2, \hat{U}_2, \hat{V}_2, \delta \hat{u}_1, \delta \hat{v}_1, \hat{y}_{c,1}, \hat{x}_1, \hat{x}_2, \hat{y}_1$, and $\hat{y}_2$ defined in (5.48)–(5.54) and (5.38)–(5.39) and the functions $\hat{\Lambda}$ and $n_{\hat{\tau}}$ defined in (5.55)–(5.56).*

Let $k \in \{i - \ell, \dots, i\}$ and introduce the sets

$$S_\tau^k = \left\{ (x, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \in [0, \bar{L}] \times [U_m, U_M]^{N_U} \times \Gamma \mid \sum_{j=k-n_\tau(x,k,\hat{\mathbf{U}}_\infty)}^k \hat{U}_\infty[j] = x \right\}, \quad (5.61)$$

$$S^k = \left\{ (\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \in [U_m, U_M]^{N_U} \times \Gamma \mid \exists j \in [i - n_\tau(\hat{x}_2(\hat{\gamma}_\infty), k, \hat{\mathbf{U}}_\infty), i] \right. \\ \left. (\hat{x}_2(\hat{\gamma}_\infty) - \hat{\Lambda}_{j,i}(\hat{\mathbf{U}}_\infty), \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \in S_\tau^k \right\}. \quad (5.62)$$

Then, for any fixed $k \in \{i - \ell, \dots, i\}$, $\hat{\Upsilon}_1, \hat{\Upsilon}_2, \hat{U}_2$, and $\hat{V}_2$ are almost everywhere continuously differentiable with respect to $(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)$, namely, outside of the set S^k . Besides, $\delta \hat{u}_1, \delta \hat{v}_1$, and $\hat{y}_{c,1}$ are also almost everywhere continuously differentiable with respect to $(x, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)$, namely, outside S_τ^k .

Furthermore, the gradient of the cost function writes, almost everywhere, as

$$\nabla J = -\Delta t \sum_{k=i-\ell}^i \left((\Upsilon_1[k] - \hat{\Upsilon}_1(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)) \nabla_{(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \hat{\Upsilon}_1(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \right. \\ \left. + (\Upsilon_2[k] - \hat{\Upsilon}_2(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)) \nabla_{(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \hat{\Upsilon}_2(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \right). \quad (5.63)$$

in which the gradients of the measurement estimates can be formulated, almost everywhere, as

$$\nabla_{(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \hat{\Upsilon}_1(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) = \left[\cos(\gamma_1 - \hat{\gamma}_\infty) \mathbf{e}_{N_U-i+k}^T, \hat{U}_\infty[k] \sin(\gamma_1 - \hat{\gamma}_\infty) \right]^T, \quad (5.64)$$

$$\nabla_{(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \hat{\Upsilon}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) = \frac{\cos(\gamma_2 - \hat{\gamma}_\infty - \text{atan}\left(\frac{\hat{V}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)}{\hat{U}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)}\right))}{\sqrt{\hat{U}_2^2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) + \hat{V}_2^2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)}} \\ \times \left(\hat{U}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \nabla_{(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \hat{U}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) + \hat{V}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \nabla_{(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \hat{V}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \right) \\ - \sqrt{\hat{U}_2^2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) + \hat{V}_2^2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \sin \left(\gamma_2 - \hat{\gamma}_\infty - \text{atan}\left(\frac{\hat{V}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)}{\hat{U}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)}\right) \right) \\ \times \left(- \begin{bmatrix} \mathbf{0}_{N_U} \\ 1 \end{bmatrix} - \frac{2\hat{U}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \nabla_{(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \hat{V}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)}{\hat{U}_2^2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) + \hat{V}_2^2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \right. \\ \left. + \frac{2\hat{V}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) \nabla_{(\hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \hat{U}_2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)}{\hat{U}_2^2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty) + \hat{V}_2^2(k, \hat{\mathbf{U}}_\infty, \hat{\gamma}_\infty)} \right), \quad (5.65)$$

with, almost everywhere, the partial gradients of the speed components

$$\begin{aligned}\nabla_{\hat{U}_\infty} \hat{U}_2(k, \hat{U}_\infty, \hat{\gamma}_\infty) &= \mathbf{e}_{N_U - i + k} - \nabla_{\hat{U}_\infty} \delta \hat{u}_1(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty) \\ &\quad \times G_a(\hat{y}_2, \hat{y}_c(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty), d_1^2(\hat{x}_2, \hat{x}_1)) \\ &\quad - \delta \hat{u}_1(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty) \frac{\partial G_a}{\partial s_2}(\hat{y}_2, \hat{y}_{c,1}(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty), d_1^2(\hat{x}_2, \hat{x}_1)) \\ &\quad \times \nabla_{\hat{U}_\infty} \hat{y}_{c,1}(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty),\end{aligned}\quad (5.66)$$

$$\begin{aligned}\nabla_{\hat{\gamma}_\infty} \hat{U}_2(k, \hat{U}_\infty, \hat{\gamma}_\infty) &= - \left(\nabla_x \delta \hat{u}(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty) \frac{d\hat{x}_2}{d\hat{\gamma}_\infty} + \nabla_{\hat{\gamma}_\infty} \delta \hat{u}(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty) \right) \\ &\quad \times G_a(\hat{y}_2, \hat{y}_{c,1}(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty), d_1^2(\hat{x}_2, \hat{x}_1)) \\ &\quad - \delta \hat{u}_1(\hat{x}_2(\hat{\gamma}_\infty), k, \hat{U}_\infty, \hat{\gamma}_\infty) \nabla G_a(\hat{y}_2, \hat{y}_{c,1}(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty), d_1^2(\hat{x}_2, \hat{x}_1))^T \\ &\quad \times \begin{bmatrix} \nabla_x \hat{y}_{c,1}(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty) \frac{d\hat{x}_2}{d\hat{\gamma}_\infty} + \nabla_{\hat{\gamma}_\infty} \hat{y}_{c,1}(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty) \\ \nabla[d_1^2](\hat{x}_2, \hat{x}_1)^T \begin{bmatrix} \frac{d\hat{x}_2}{d\hat{\gamma}_\infty} \\ \frac{d\hat{x}_1}{d\hat{\gamma}_\infty} \end{bmatrix} \end{bmatrix},\end{aligned}\quad (5.67)$$

$$\nabla_{\hat{U}_\infty} \hat{V}_2(k, \hat{U}_\infty, \hat{\gamma}_\infty) = - \nabla_{\hat{U}_\infty} \delta \hat{v}_1(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty), \quad (5.68)$$

$$\nabla_{\hat{\gamma}_\infty} \hat{V}_2(k, \hat{U}_\infty, \hat{\gamma}_\infty) = - \left(\nabla_x \delta \hat{v}_1(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty) \frac{d\hat{x}_2}{d\hat{\gamma}_\infty} + \nabla_{\hat{\gamma}_\infty} \delta \hat{v}_1(\hat{x}_2, k, \hat{U}_\infty, \hat{\gamma}_\infty) \right), \quad (5.69)$$

and, almost everywhere, the partial gradients of the streamwise speed deficit

$$\begin{aligned}\nabla_x \delta \hat{u}_1(x, k, \hat{U}_\infty, \hat{\gamma}_\infty) &= \frac{2a_1(\gamma_1 - \hat{\gamma}_\infty)}{d_1^2(x, \hat{x}_1)} \sum_{j=k-n_{\hat{\tau}}(x, k, \hat{U}_\infty)}^k \hat{U}_\infty^2[j] \Delta t \left(\frac{\partial H_1}{\partial x}(x - \hat{\Lambda}_{j,k}, \hat{x}_1) \right. \\ &\quad \left. - \frac{2}{d_1(x, \hat{x}_1)} \frac{\partial d_1}{\partial x}(x, \hat{x}_1) H_1(x - \hat{\Lambda}_{j,k}, \hat{x}_1) \right),\end{aligned}\quad (5.70)$$

$$\begin{aligned}\nabla_{\hat{U}_\infty} \delta \hat{u}_1(x, k, \hat{U}_\infty, \hat{\gamma}_\infty) &= \frac{2a_1(\gamma_1 - \hat{\gamma}_\infty)}{d_1^2(x, \hat{x}_1)} \Delta t \sum_{j=k-n_{\hat{\tau}}(x, k, \hat{U}_\infty)}^k \left(2H_1(x - \hat{\Lambda}_{j,k}, \hat{x}_1) \right. \\ &\quad \left. \times \hat{U}_\infty[j] \mathbf{e}_{N_U - i + j} - \frac{\partial H_1}{\partial x}(x - \hat{\Lambda}_{j,k}, \hat{x}_1) \hat{U}_\infty^2[j] \sum_{p=j+1}^k \mathbf{e}_{N_U - i + p} \right),\end{aligned}\quad (5.71)$$

$$\begin{aligned}\nabla_{\hat{\gamma}_\infty} \delta \hat{u}_1(x, k, \hat{U}_\infty, \hat{\gamma}_\infty) &= - \frac{a'_1(\gamma_1 - \hat{\gamma}_\infty)}{a_1(\gamma_1 - \hat{\gamma}_\infty)} \delta \hat{u}_1(x, k, \hat{U}_\infty, \hat{\gamma}_\infty) + \frac{2a_1(\gamma_1 - \hat{\gamma}_\infty)}{d_1^2(x, \hat{x}_1)} \frac{d\hat{x}_1}{d\hat{\gamma}_\infty} \Delta t \\ &\quad \times \sum_{j=i-n_{\hat{\tau}}(x, \hat{U}_\infty)}^i \left(\frac{\partial H_1}{\partial x_1}(x - \hat{\Lambda}_{j,k}, \hat{x}_1) - \frac{2H_1(x - \hat{\Lambda}_{j,k}, \hat{x}_1)}{d_1(x, \hat{x}_1)} \frac{\partial d_1}{\partial x_1}(x, \hat{x}_1) \right) U_\infty^2[j],\end{aligned}\quad (5.72)$$

and, similarly, almost everywhere, of the one in the transverse direction

$$\begin{aligned} \nabla_x \delta \hat{v}_1(x, k, \hat{U}_\infty, \hat{\gamma}_\infty) &= \frac{2b(\gamma_1 - \hat{\gamma}_\infty)}{4d_1^2(x, \hat{x}_1)} \sum_{j=k-n_{\hat{\tau}}(x, k, \hat{U}_\infty)}^k \hat{U}_\infty^2[j] \Delta t \left(\frac{\partial H_1}{\partial x}(x - \hat{\Lambda}_{j,k}, \hat{x}_1) \right. \\ &\quad \left. - \frac{2}{d_1(x, \hat{x}_1)} \frac{\partial d_1}{\partial x}(x, \hat{x}_1) H_1(x - \hat{\Lambda}_{j,k}, \hat{x}_1) \right), \end{aligned} \quad (5.73)$$

$$\begin{aligned} \nabla_{\hat{U}_\infty} \delta \hat{v}_1(x, k, \hat{U}_\infty, \hat{\gamma}_\infty) &= \frac{b(\gamma_1 - \hat{\gamma}_\infty)}{4d_1^2(x, \hat{x}_1)} \Delta t \sum_{j=k-n_{\hat{\tau}}(x, k, \hat{U}_\infty)}^k \left(2H_1(x - \hat{\Lambda}_{j,k}, \hat{x}_1) \right. \\ &\quad \left. \hat{U}_\infty[j] e_{N_U-i+j} - \frac{\partial H_1}{\partial x}(x - \hat{\Lambda}_{j,k}, \hat{x}_1) \hat{U}_\infty^2[j] \sum_{p=j+1}^k e_{N_U-i+p} \right), \quad (5.74) \\ \nabla_{\hat{\gamma}_\infty} \delta \hat{v}_1(x, k, \hat{U}_\infty, \hat{\gamma}_\infty) &= -\frac{b'(\gamma_1 - \hat{\gamma}_\infty)}{b(\gamma_1 - \hat{\gamma}_\infty)} \delta \hat{v}_1(x, k, \hat{U}_\infty, \hat{\gamma}_\infty) + \frac{b(\gamma_1 - \hat{\gamma}_\infty)}{4d_1^2(x, \hat{x}_1)} \frac{d\hat{x}_1}{d\hat{\gamma}_\infty} \Delta t \\ &\quad \times \sum_{j=k-n_{\hat{\tau}}(x, k, \hat{U}_\infty)}^k \left(\frac{\partial H_1}{\partial x_1}(x - \hat{\Lambda}_{j,k}, \hat{x}_1) - \frac{2H(x - \hat{\Lambda}_{j,k}, \hat{x}_1)}{d_1(x, \hat{x}_1)} \frac{\partial d_1}{\partial x_1}(x, \hat{x}_1) \right) U_\infty^2[j], \end{aligned} \quad (5.75)$$

and, almost everywhere, for the wake centerline

$$\begin{aligned} \nabla \hat{y}_{c,1}(x, k, \hat{U}_\infty, \hat{\gamma}_\infty) &= \begin{bmatrix} 0 \\ \mathbf{0}_{N_U} \\ \frac{d\hat{y}_1}{d\hat{\gamma}_\infty} \end{bmatrix} - \sum_{j=k-n_{\hat{\tau}}(x, k, \hat{U}_\infty)}^k \Delta t \\ &\quad \times \begin{bmatrix} \nabla_x \delta \hat{v}_1(x - \hat{\Lambda}_{j,k}(\hat{U}_\infty), k, \hat{U}_\infty, \hat{\gamma}_\infty) \\ -\nabla_x \delta \hat{v}_1(x - \hat{\Lambda}_{j,k}(\hat{U}_\infty), \hat{U}_\infty, \hat{\gamma}_\infty) \sum_{k=j+1}^k e_{N_U-i+j} + \nabla_{\hat{U}_\infty} \delta \hat{v}(x - \hat{\Lambda}_{j,k}(\hat{U}_\infty), \hat{U}_\infty, \hat{\gamma}_\infty) \\ \nabla_{\hat{\gamma}_\infty} \delta \hat{v}(x - \hat{\Lambda}_{j,k}(\hat{U}_\infty), k, \hat{U}_\infty, \hat{\gamma}_\infty) \end{bmatrix}, \end{aligned} \quad (5.76)$$

in which, finally, for $n = 1, 2$,

$$\frac{d\hat{x}_n}{d\hat{\gamma}_\infty} = -X_n \sin(\hat{\gamma}_\infty) + Y_n \cos(\hat{\gamma}_\infty) \quad (5.77)$$

$$\frac{d\hat{y}_n}{d\hat{\gamma}_\infty} = -X_n \cos(\hat{\gamma}_\infty) - Y_n \sin(\hat{\gamma}_\infty) \quad (5.78)$$

Proof. Let us observe that S_τ^k and S^k are hypersurfaces and, thus, sets of measure zero. Besides, from (5.50)–(5.51), $\delta \hat{u}_1$ and $\delta \hat{v}_1$ are continuously differentiable functions apart from the points $(x, \hat{U}_\infty, \hat{\gamma}_\infty)$ such that $\sum_{j=k-n_{\hat{\tau}}(x, k, \hat{U}_\infty)}^k \hat{U}_\infty[j] = x$ where they encounter a discontinuity, due to a jump of $n_{\hat{\tau}}(x, k, \hat{U}_\infty)$. As a result, $(\hat{U}_\infty, \hat{\gamma}_\infty) \mapsto y_{c,1}(\hat{x}_2(\hat{\gamma}_\infty), k, \hat{U}_\infty, \hat{\gamma}_\infty)$ jumps when $(\hat{U}_\infty, \hat{\gamma}_\infty) \in S^k$ and is continuously differentiable otherwise. Thus, the same holds for $\hat{U}_2, \hat{V}_2, \hat{Y}_1$, and $\hat{Y}_2$. The rest of the proof follows from lengthy but straightforward computations. $\square$

Observe that, even if introduced for the specific case $N = 2$, these expressions can be straightforwardly extended to any N . Yet, as these formulas are pretty lengthy, we do not formulate them here. The expression of this gradient can then be provided to a constrained-optimization solver, to solve Problem 5.2.

5.5 Implementation and numerical results

To solve Problem 5.2, we proposed to use Matlab’s constrained optimization function ‘fmincon’ [33] and, more specifically, its interior-point solver.

Preliminary studies (conducted on a set of parameters detailed in the following subsection and for two turbines) resulted into a solution corresponding to a local minimum where the speed estimation is quite accurately estimated, but the direction estimation is not. Overall, these preliminary studies indicated that the sensitivity of the problem with respect to the wind direction is reduced in the vicinity of the correct wind direction.

Thus, on top of the gradient-based solver, a line search was implemented to look for a possibly better estimate of the wind direction after the solver had already converged. Hence, after a gradient-based optimization step, we propose to keep the estimation of the speed fixed and try different values for the wind direction, we then keep the one that yields the smallest cost. This second step has a computational cost which is much smaller than the optimization one, as it only requires n calls to the objective function.

This procedure is summarized in Algorithm 2 which presents a pseudo-code of the proposed methodology to solve Problem 5.2. n is the number of points checked by the line search, with $\hat{\gamma}_{\infty,1}, \dots, \hat{\gamma}_{\infty,n}$ potential wind direction values, which could a priori be adapted from one time-step i to another but are here kept as constants.

Algorithm 2 Implementation of the proposed method

```

Define initial guesses  $(\hat{U}_{\infty}^0, \hat{\gamma}_{\infty}^0, \hat{U}_0^0)$ 
Find a local minimum for  $J[i]$  (using ‘fmincon’), denoted  $J^*$ , and an
optimal decision variable  $(\hat{U}_{\infty}^*, \hat{\gamma}_{\infty}^*, \hat{U}_0^*)$ 
for  $j = 1, \dots, n$  do
  Compute  $J[i]$  using  $(\hat{U}_{\infty}^*, \hat{U}_0^*)$  and  $\hat{\gamma}_{\infty,j}$ 
  if  $J[i] < J^*$  then
     $J^* \leftarrow J[i]$ 
     $\hat{\gamma}_{\infty}^* \leftarrow \hat{\gamma}_{\infty,j}$ 
  end if
end for
return  $(\hat{U}_{\infty}^*, \hat{\gamma}_{\infty}^*, \hat{U}_0^*)$ 

```

To compute $J[i]$, which requires computing $\hat{\Upsilon}_1[i - \ell], \dots, \hat{\Upsilon}_1[i]$ and $\hat{\Upsilon}_2[i -$

$\ell], \dots, \hat{\Upsilon}_2[i]$, we use the formulas presented previously for the discrete-time case. Notice we only need to compute $\hat{y}_{c,1}$ and $\delta\hat{u}_1$ at $\hat{x}_2$, but we need $\delta\hat{v}_1$ at several locations to compute $\hat{y}_{c,1}$ at $\hat{x}_2$ (cf. (5.54)). Thus, we compute the values of $\delta\hat{v}_1$ in a grid and then we interpolate to obtain the corresponding values at the necessary locations.

5.5.1 Numerical results with Shapiro's model as a reference

Here, we present some numerical results obtained applying the technique described in this chapter on a handful of cases. We start with a simplified situation, where the wind speed is constant. Then, we move to a case where the wind speed is a sinusoidal wave, and finally to a case where we use an experimentally obtained wind speed⁸ as input to the simulation, so we can assess the performance of our method in a more realistic setting. In the cases presented in this section, the measurements of the turbines are obtained by simulating Shapiro's model with the same parameters we use for the optimization, so there is no modeling error.

In every case, we use a turbine with a diameter $D = 126$ m, this corresponds to the NREL 5 MW wind turbine model [65]. The value for $C_{T,1}$ is fixed at 0.79, which is a typical value for C_T we obtain for the NREL 5 MW wind turbine model in simulations with FAST [64]. We also use $\kappa_1 = 0.05$ and $\sigma_1 = 0.235D$ which were chosen based on results obtained with the simulator FAST.Farm [66, 67]. The time-window for the optimization is $\ell = 100$ time-steps and $\Delta t = 1$ second, which is the sampling rate we had for the experimental data in Chapter 3. The bounds for the wind speed are $U_m = 1$ m/s and $U_M = 30$ m/s and the bounds for the wind direction are $\gamma_m = 0$ and $\gamma_M = 15$ degrees. The initial guess for the wind speed is constant equal to 12 m/s and for the direction it is 5 degrees in every case.

In all the following three cases, the distance between the turbines in the x direction is $5D$, which is a typical spacing encountered in a wind farm. Notice that with this spacing and the bounds for the wind speed considered, the transport delay from one turbine to the other is between 21 and 630 seconds.

In the first case, constant wind speed, the optimization converges to the correct wind speed and direction, we do not show plots for this case since they are just constant curves. In this case, the yaw angle of both turbines is 10 degrees, so their rotor planes are perfectly perpendicular to the wind direction. The distance between the turbines in the y direction is zero.

The second case is the one with the sinusoidal wind speed. Here, the yaw angles are zero, and the distance between the turbines in the y direction is $0.1D$. The angle converges to the correct result, as before. For the speed, we see in Fig. 5.3 that it converges almost perfectly to the correct result.

⁸Which comes from the LiDAR data set provided by Leosphere, as in the two previous chapters.

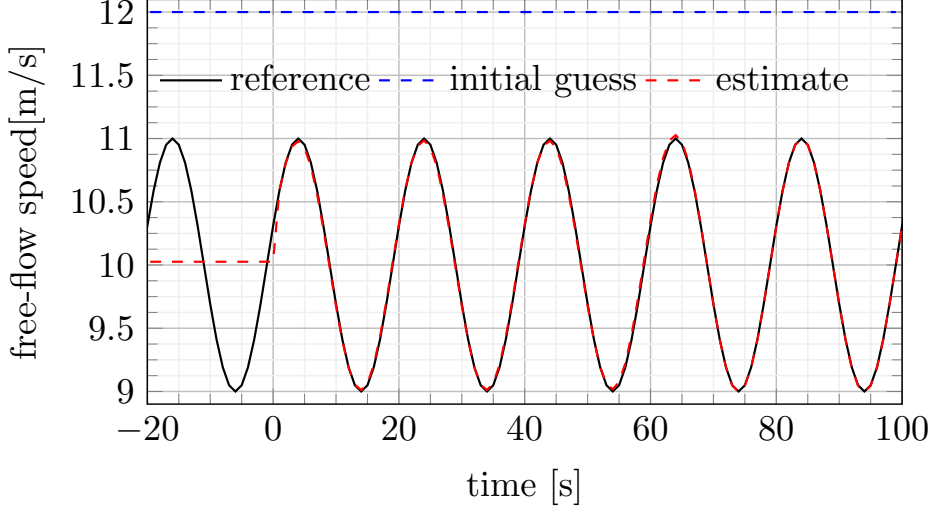


Figure 5.3: Case with sinusoidal wind speed with the estimation computed for time $t = 100$ s so that $t - \ell\Delta t = 0$. The negative time values correspond to a portion of the time window $[t - 2\bar{\tau} - \ell\Delta t, t - \ell\Delta t] = [-2\bar{\tau}, 0]$, where (5.57) holds, that is, the speed estimate is constant. We show a few seconds of this interval here to illustrate this point.

The most interesting curve appears in the case with experimental wind speed. Here, the yaw angles of both turbines are 10 degrees and the distance in the y direction between the turbines is $0.5D$. The wind direction again converges to the correct value. The curve for the speed estimate in Fig. 5.4 show that we also have good agreement with the reference in this case.

To evaluate the robustness of this estimation procedure, next, we present cases where the measurements come from FAST.Farm simulations, so we can test our method in a more realistic setup.

5.5.2 Numerical results with FAST.Farm as a reference model

Here, we obtain the measurements by simulating the wind farm using the software FAST.Farm [66, 67]. This yields a modeling error since the optimization procedure is carried out using Shapiro's 2-D model as explained before.

We present three cases using the same free-flow as in the experimental wind speed case of the previous section with a free-flow direction of 10 degrees and two wind turbines with the same yaw angle. The x -distance between the turbines is $5D$ in all three cases and we use a time-window of $\ell = 100$ time-steps and $\Delta t = 1$ second, as before. The bounds for the wind speed are $U_m = 1$ m/s and $U_M = 30$ m/s and the bounds for the wind direction are $\gamma_m = 0$ and $\gamma_M = 15$ degrees, the same as before. The initial

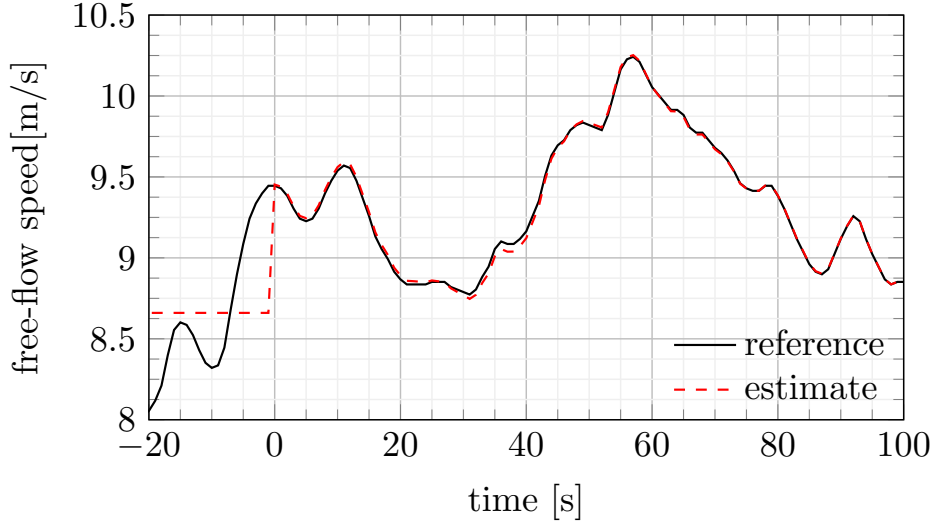


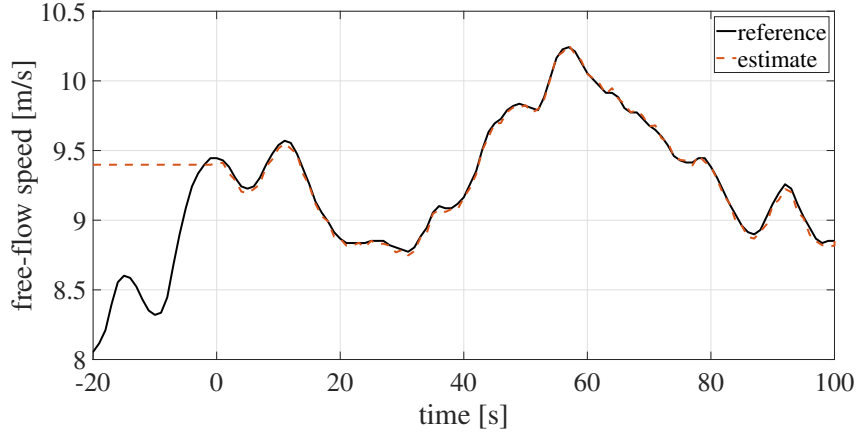
Figure 5.4: Case with experimental wind speed. The estimation is computed at time $t = 100$ s so that $t - \ell\Delta t = 0$. The negative time values correspond to a portion of the time window $[t - 2\bar{\tau} - \ell\Delta t, t - \ell\Delta t] = [-2\bar{\tau}, 0]$, where (5.57) holds, that is, the speed estimate is constant. The initial guess for the wind is constant equal to 12 m/s.

guess for the wind speed is constant equal to 12 m/s and for the direction it is 5 degrees in every case, except when otherwise specified.

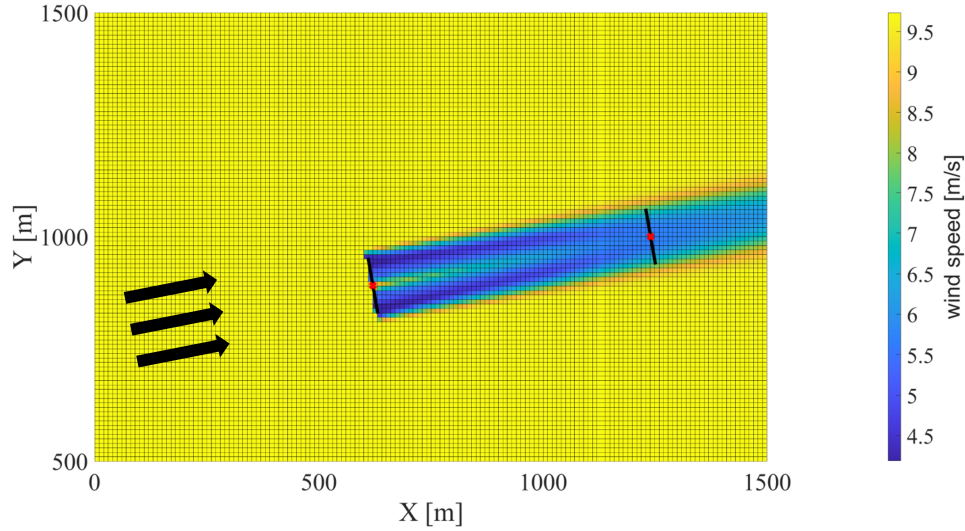
Fig. 5.5a shows the case in which the yaw of both turbines is 10 degrees and the y -distance between them is zero. We see that the agreement between the speed estimate and the reference is excellent. The direction estimation is less than 4 degrees larger than the reference, which is the worst error among the cases we show, but still a reasonable value given that the measurements depend on the cosine between the yaw angle and the direction, so a 4 degrees change should not be so relevant.

Fig. 5.6a shows the case in which the yaw of both turbines is 0 degree and the y -distance between them is $0.5D$. In this case, there is a misalignment between the turbines and the wind direction and we see that the estimations are not as good as in the previous case, but the direction estimation is still better than in the first case, it converges to 8.5 degrees.

Fig. 5.7a shows the case in which the yaw of both turbines is 10 degrees and the y -distance between them is $0.5D$. In this case, we see the effect of the initial guess on the estimate. If we start with the initial guess 5 degrees for the wind direction, it converges to 8 degrees and the speed estimate converges to the green curve in Fig. 5.7a. If we start with the initial guess 8 degrees for the direction, it converges almost perfectly to the correct value and the speed estimate is the one in red. For both cases, the initial guess



(a) Speed estimation. The estimation is computed at time $t = 100$ s so that $t - \ell\Delta t = 0$. The negative time values correspond to a portion of the time window $[t - 2\bar{\tau} - \ell\Delta t, t - \ell\Delta t] = [-2\bar{\tau}, 0]$, where (5.57) holds, that is, the speed estimate is constant.

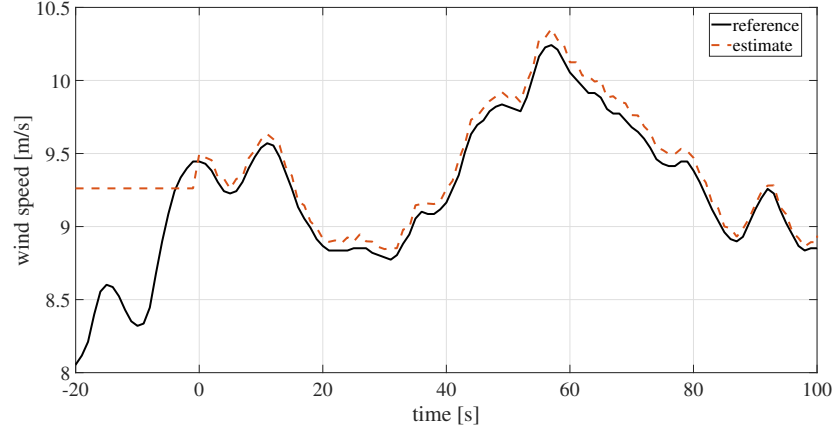


(b) Wake of the upwind turbine. The speed displayed is $\sqrt{U^2 + V^2}$ computed using only the deficit caused by the first turbine. The black lines represent the turbines' rotors and the red crosses their centers. The arrows show the wind direction.

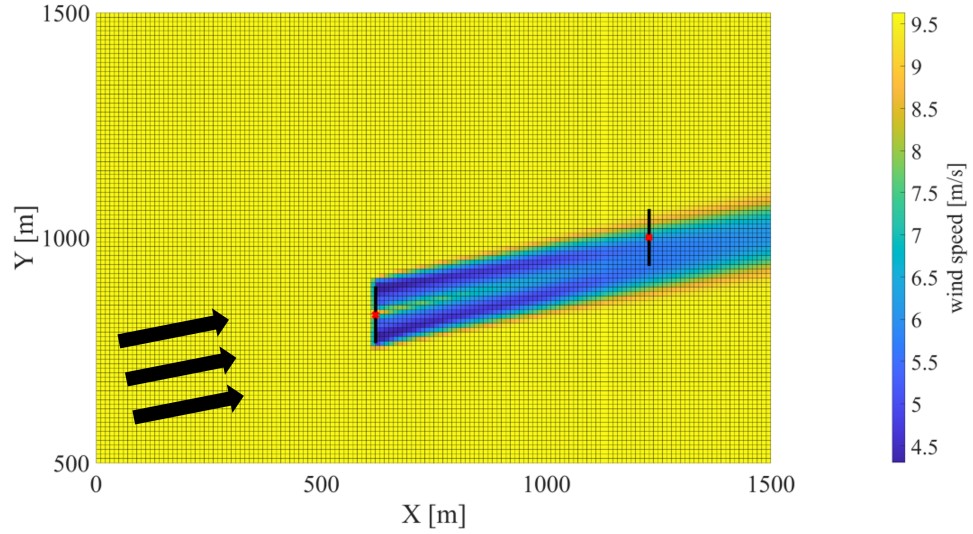
Figure 5.5: Case with yaw angle 10 degrees and y -distance 0.

for the wind speed is constant equal to 12 m/s.

Notice that we do not get perfect estimation in these cases, but considering the modeling error involved, since Shapiro's model is much simpler than FAST.Farm, we consider the results satisfactory. It is interesting to note that the case where the y -distance between the turbines is $0.5D$ and the yaw is the same as the wind direction is the one with the best result in

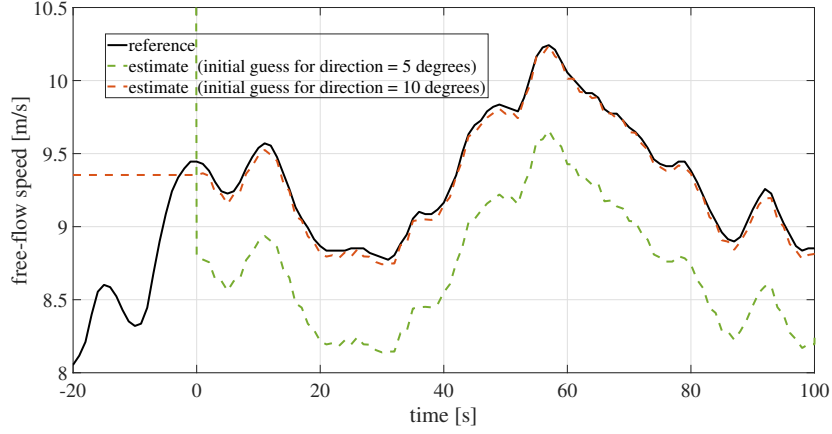


(a) Speed estimation. The estimation is computed at time $t = 100$ s so that $t - \ell\Delta t = 0$. The negative time values correspond to a portion of the time window $[t - 2\bar{\tau} - \ell\Delta t, t - \ell\Delta t] = [-2\bar{\tau}, 0]$, where (5.57) holds, that is, the speed estimate is constant.

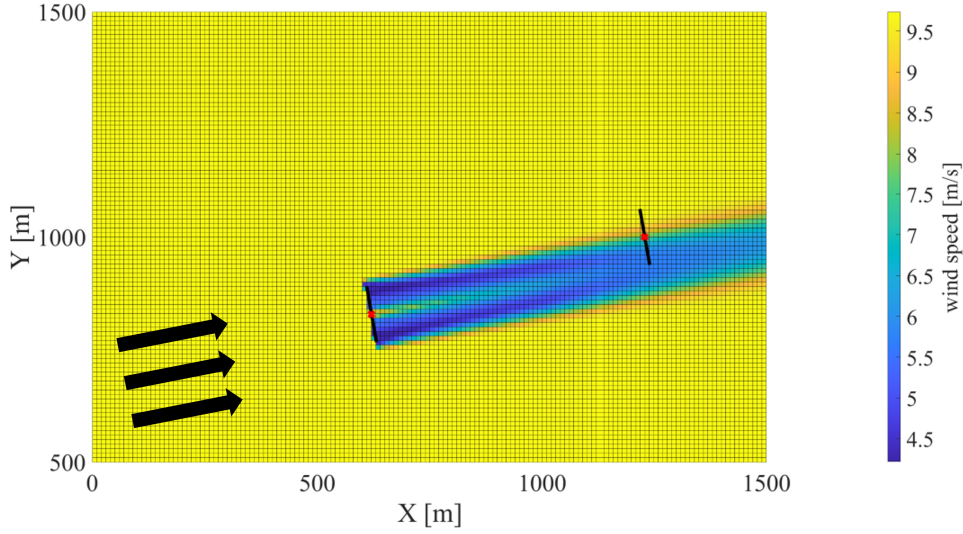


(b) Wake of the upwind turbine. The speed displayed is $\sqrt{U^2 + V^2}$ computed using only the deficit caused by the first turbine. The black lines represent the turbines' rotors and the red crosses their centers. the arrows show the wind direction.

Figure 5.6: Case with yaw angle 0 degree and y -distance $0.5 D$.



(a) Speed estimation. The estimation is computed at time $t = 100$ s so that $t - \ell\Delta t = 0$. The negative time values correspond to a portion of the time window $[t - 2\bar{\tau} - \ell\Delta t, t - \ell\Delta t] = [-2\bar{\tau}, 0]$, where (5.57) holds, that is, the speed estimate is constant. For the green curve, the constant value for $\hat{U}_\infty$ at negative times is 30 m/s.



(b) Wake of the upwind turbine. The speed displayed is $\sqrt{U^2 + V^2}$ computed using only the deficit caused by the first turbine. The black lines represent the turbines' rotors and the red crosses their centers. The arrows show the wind direction.

Figure 5.7: Case with yaw angle 10 degrees and y -distance $0.5 D$.

terms of the direction estimate, but not in terms of the speed estimation. This indicates a trade-off between these two errors. Observe that, contrary to the simulations carried out with Shapiro’s model, here one can obtain quite inaccurate wind speed estimates with this identification method, as illustrated in Fig. 5.7a. This may question the procedure followed by Algorithm 2 where the line search for the wind direction is evaluated for the fixed wind speed trajectory previously estimated. A different path to explore would consist in running the optimization procedure for each step of the wind direction line search. However, the computational cost of calling the optimization solver would only make this procedure possible for a finite number of points. This should be investigated further in future works.

Also, all simulations carried out in this section concern two turbines, but we expect we would obtain better results in general with wind farms with more turbines, since that would yield more information to use in the optimization procedure. This could improve the estimation errors reported in the simulations results in this section and constitutes an important path to explore in the near future.

Another important point is that our model considers a point-wise speed measurement, which can be inaccurate when the second turbine is only partially inside the wake zone of the first. Figs. 5.5b, 5.6b, and 5.7b show the wind speed computed using only the deficit caused by the first turbine at a certain time instant, we see that only in the first case the second turbine is completely inside the wake zone of the first one.

5.6 Conclusion

This chapter presented a optimization-based approach to estimate the free-flow wind speed and direction for a wind farm using REWS measurements from each of the turbines in said wind farm. The proposed method relies on a simplified 2-D wake model and we showed results where the FAST.Farm simulator plays the role of the real wind farm to provide measurements. We get a good agreement between the estimated values and the reference ones which shows the interest of using this method in control applications in the future.

Even though the proposed method is not real-time compliant, in the sense that it cannot be solved iteratively at each time step due to its computational cost, it is a priori fast enough to be used to provide a periodic update of the wind speed direction, which can be considered as constant within the scale of minutes. Yet, the numerical tests carried out in this chapter are not sufficient to evaluate this feature, in particularly with respect to the number of turbines under consideration. Thus, further numerical tests are required in the future. However, it may be possible to combine this approach with the one proposed for wind speed estimation in the previous

chapter. Future studies should thus focus on combining these techniques, which implies to study how to extend the real-time estimation strategy of Chapter 4 to handle two dimensions.

Also, an important point would be to combine the method presented with a data-based REWS estimation technique such as the ones presented in Chapter 3. This would yield an algorithm that estimates the free-flow wind speed and direction from the Supervisory Control And Data Acquisition (SCADA) system data in multiple steps. This requires to quantify more precisely the robustness of the method proposed here to local wind measurement uncertainties.

This chapter also opens numerous challenging and interesting questions for future research, such as the impact of the number of turbines and the application of this technique to large wind farms where is not evident which turbines' wakes affect which other turbines. Indeed, in this chapter, we assumed that we were able a priori to order the turbines according to their relative wake interactions. This order obviously depends on the wind direction. Thus, for wind farms with a more intricate geometry, the combinations of different wake geometry should be taken into account.

Chapter 6

Conclusions and perspectives

6.1 Conclusions

This thesis presented a number of estimation methods to be used for wind estimation in the context of wind turbines and wind farms.

In Chapter 3, we presented two different methods to estimate the Rotor Effective Wind Speed (REWS) from data obtained from the turbine's Supervisory Control And Data Acquisition (SCADA) system. Both methods aim to provide this estimate without a priori knowledge of physical parameters such as the turbine's torque coefficient table. This way, these methods can be used by wind turbine operators who do not have access to the constructor's data needed to obtain these parameters. For training and testing, we used experimental SCADA and REWS data (obtained with a Light Detection And Ranging (LiDAR) sensor) provided by Engie Green and Leosphere in the framework of the ANR project SmartEole.

The first method we presented uses Gaussian Process Regression (GPR) to find a map from the SCADA data – namely, the rotor angular speed, the produced power, and the pitch angle of the blades – to the REWS. More specifically, we use a combination of GPR and least squares regression, together with a clustering technique, to find a computational model that outputs an estimate of the REWS given the recent history (values in the current and previous r time-steps) of the SCADA data.

The second method combines GPR with a high-gain observer in a two-steps approach. The first step is to estimate the aerodynamic torque acting on the turbine using a GPR model. The second is to estimate the REWS using a high-gain observer. This method uses a lower dimensional input to the GPR. We also presented some conditions for convergence of the estimate, which holds implications for the high-gain observer literature in general.

In Chapter 4, we introduced a simple and real-time compliant estimation approach to estimate the free-flow wind speed when the only speed measurement available is inside the wake zone of an array of aligned tur-

bines, i.e., a measurement that is taken in the area affected by the presence of the turbines themselves and therefore does not correspond to the free-flow wind speed. We employed a one-dimensional analytical model of the wake (the effect of the turbines on the wind field) to compute an artificial measurement based on the current estimate of the free-flow wind speed and then integrate the difference between this artificial measurement and the true one to update the free-flow speed estimate. We proved that, under certain assumptions, the estimation error is bounded by a term that can be arbitrarily reduced by increasing the gain of the integrator. Simulation results obtained with measurements provided by a higher-fidelity simulator illustrated the interest of the method.

In Chapter 5, we presented an optimization-based technique to estimate both the free-flow wind speed and direction from the REWS of each turbine in the wind farm. This method adopts a two-dimensional model of the wake, an extension of the one-dimensional model used in Chapter 4. We employed this model and its analytical solution based on transport delays to generate artificial measurements and solve a corresponding least-square optimization problem. Yet, this optimization problem reveals to be particularly challenging as, due to the appearance of transport delays, the size of the decision variable depends on the decision variable values themselves. To circumvent this issue, we propose to simplify the problem and estimate an average value of the free-flow wind speed history. Again, we presented numerical results where the measurements come from a higher-fidelity simulator than the one used in the optimization procedure. These simulation results also illustrated the interest of the proposed method.

6.2 Perspectives for future research

This research opens several promising avenues for future exploration and expansion.

Regarding Chapter 3, where we presented REWS estimation methods, one can explore the possibility of employing jointly with GPR different observers than the high-gain observer proposed in this thesis. Future research may focus on the possibility to adapt the ideas presented in [25] to the considered set-up and estimate both dynamics and states, with the hope to cope with time variations of the turbine’s torque coefficient map, due to aging.

Regarding Chapter 4, where we presented a method to estimate the free-flow wind speed using a one-dimensional wake model, it would be interesting to extend the proposed technique to handle the case where the array of wind turbines under consideration is not aligned. A possible solution could be, considering the wind direction as known, to ground on the extension of Shapiro’s model to two dimensions described in Chapter 5 to provide an estimate of the local measurement and then update the free-flow wind

speed with a simple integrator such as the one proposed in Chapter 4. The convergence analysis performed in Chapter 4 would then have to be revisited.

Regarding Chapter 5, where we presented a method to estimate the free-flow wind speed and direction, there exists a number of interesting questions to be further investigated.

First is the impact of the number of turbines in the estimate and in particular how to deal with a large wind farm with many wind turbines, one could explore concepts such as clustering of the turbines and decentralization of the computation. This ties to a second question, which is the matter of the computation burden associated with the method presented and how one can proceed to reduce it. Indeed, even though it seems unlikely that the proposed method can be made real-time compliant, optimizing its computational time is an important point to be considered in the future. In any case, for a small enough number of turbines, the proposed estimation methodology can a priori be computed fast enough so that it can be used to provide a periodic update of the wind speed direction, which can be considered as constant within the scale of minutes. This specific question should nevertheless be confirmed by further numerical tests than those carried out in this thesis, involving a larger number of turbines. It would then be interesting to combine this approach with the one proposed for wind speed estimation in the previous chapter. Future studies should thus investigate this question, with the aim of providing real-time combined estimation of both free-flow wind speed and direction for wind farm control.

This thesis thus opens interesting and challenging questions for future research.

Bibliography

- [1] M. A. Abdullah, A. H. M. Yatim, C. W. Tan, and R. Saidur. “A review of maximum power point tracking algorithms for wind energy systems”. In: *Renewable and Sustainable Energy Reviews* 16.5 (2012), pp. 3220–3227. ISSN: 1364-0321. DOI: 10.1016/j.rser.2012.02.016.
- [2] Ahmed G. Abo-Khalil and Dong-Choon Lee. “MPPT Control of Wind Generation Systems Based on Estimated Wind Speed Using SVR”. In: *IEEE Transactions on Industrial Electronics* 55.3 (2008), pp. 1489–1490. DOI: 10.1109/TIE.2007.907672.
- [3] Leif Erik Andersson, Olimpo Anaya-Lara, John Olav Tande, Karl Otto Merz, and Lars Imsland. “Wind farm control - Part I: A review on control system concepts and structures”. In: *IET Renewable Power Generation* 15.10 (2021), pp. 2085–2108. DOI: 10.1049/rpg2.12160.
- [4] Henrik Anfinssen, Haavard Holta, and Ole Morten Aamo. “Adaptive Control of a Scalar 1-D Linear Hyperbolic PDE with Uncertain Transport Speed Using Boundary Sensing”. In: *2020 American Control Conference (ACC)*. 2020, pp. 1575–1581. DOI: 10.23919/ACC45564.2020.9147863.
- [5] Jennifer Annoni, Christopher Bay, Kathryn Johnson, Emiliano Dall’Anese, Eliot Quon, Travis Kemper, and Paul Fleming. “Wind direction estimation using SCADA data with consensus-based optimization”. In: *Wind Energy Science* 4.2 (2019), pp. 355–368. DOI: 10.5194/wes-4-355-2019.
- [6] Jennifer Annoni, Peter Seiler, Kathryn Johnson, Paul Fleming, and Pieter Gebraad. “Evaluating wake models for wind farm control”. In: *2014 American Control Conference*. IEEE. 2014, pp. 2517–2523. DOI: 10.1109/ACC.2014.6858970.
- [7] David Arthur and Sergei Vassilvitskii. “k-Means++: The Advantages of Careful Seeding”. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA ’07. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. ISBN: 9780898716245. URL: <https://dl.acm.org/doi/pdf/10.5555/1283383.1283494>.

- [8] Luis Arturo Soriano, Wen Yu, Jose de Jesus Rubio, et al. “Modeling and control of wind turbine”. In: *Mathematical Problems in Engineering* 2013 (2013). DOI: 10.1155/2013/982597.
- [9] Muhammad Mahmood Aslam Bhutta, Nasir Hayat, Ahmed Uzair Farooq, Zain Ali, Sh. Rehan Jamil, and Zahid Hussain. “Vertical axis wind turbine – A review of various configurations and design techniques”. In: *Renewable and Sustainable Energy Reviews* 16.4 (2012), pp. 1926–1939. ISSN: 1364-0321. DOI: 10.1016/j.rser.2011.12.004.
- [10] J. L. Avila, C. Bonnet, H. Özbay, J. Clairambault, S. I. Niculescu, P. Hirsch, and F. Delhommeau. “A coupled model for healthy and cancerous cells dynamics in Acute Myeloid Leukemia”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 7529–7534. ISSN: 1474-6670. DOI: 10.3182/20140824-6-ZA-1003.02574.
- [11] R. J. Barthelmie, S. C. Pryor, S. T. Frandsen, K. S. Hansen, J. G. Schepers, K. Rados, W. Schlez, A. Neubert, L. E. Jensen, and S. Neckelmann. “Quantifying the Impact of Wind Turbine Wakes on Power Output at Offshore Wind Farms”. In: *Journal of Atmospheric and Oceanic Technology* 27.8 (2010), pp. 1302–1317. DOI: 10.1175/2010JTECHA1398.1.
- [12] Hedi Basbas, Yong-Chao Liu, Salah Laghrouche, Mickaël Hilairet, and Franck Plestan. “Review on Floating Offshore Wind Turbine Models for Nonlinear Control Design”. In: *Energies* 15.15 (2022). DOI: 10.3390/en15155477.
- [13] Majid Bastankhah and Fernando Porté-Agel. “A new analytical model for wind-turbine wakes”. In: *Renewable Energy* 70 (2014). Special issue on aerodynamics of offshore wind energy systems and wakes, pp. 116–123. ISSN: 0960-1481. DOI: 10.1016/j.renene.2014.01.002.
- [14] Majid Bastankhah, Bridget L. Welch, Luis A. Martínez-Tossas, Jennifer King, and Paul Fleming. “Analytical solution for the cumulative wake of wind turbines in wind farms”. In: *Journal of Fluid Mechanics* 911 (2021), A53. DOI: 10.1017/jfm.2020.1037.
- [15] Marcus Becker, Bastian Ritter, Bart Doekemeijer, Daan van der Hoek, Ulrich Konigorski, Dries Allaerts, and Jan-Willem van Wingerden. “The revised FLORIDyn model: Implementation of heterogeneous flow and the Gaussian wake”. In: *Wind Energy Science* 7.6 (2022), pp. 2163–2179. DOI: 10.5194/wes-7-2163-2022.
- [16] Pauline Bernard, Vincent Andrieu, and Daniele Astolfi. “Observer design for continuous-time dynamical systems”. In: *Annual Reviews in Control* 53 (2022), pp. 224–248. ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2021.11.002.

- [17] Pauline Bernard, Laurent Praly, and Vincent Andrieu. “Observers for a non-Lipschitz triangular form”. In: *Automatica* 82 (2017), pp. 301–313. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2017.04.054.
- [18] Michelangelo Bin and Florent Di Meglio. “Boundary Estimation of Boundary Parameters for Linear Hyperbolic PDEs”. In: *IEEE Transactions on Automatic Control* 62.8 (2017), pp. 3890–3904. DOI: 10.1109/TAC.2016.2643442.
- [19] S. Boersma, B. M. Doekemeijer, P. M. O. Gebraad, P. A. Fleming, J. Annoni, A. K. Scholbrock, J. A. Frederik, and J-W. van Wingerden. “A tutorial on control-oriented modeling and control of wind farms”. In: *2017 American Control Conference (ACC)*. 2017, pp. 1–18. DOI: 10.23919/ACC.2017.7962923.
- [20] Sjoerd Boersma, Bart Doekemeijer, Mehdi Vali, Johan Meyers, and Jan-Willem van Wingerden. “A control-oriented dynamic wind farm model: WFSim”. In: *Wind Energy Science* 3.1 (2018), pp. 75–95. DOI: 10.5194/wes-3-75-2018.
- [21] Ervin Bossanyi. “Optimising yaw control at wind farm level”. In: *Journal of Physics: Conference Series*. Vol. 1222. 1. IOP Publishing. 2019, p. 012023. DOI: 10.1088/1742-6596/1222/1/012023.
- [22] Boubekeur Boukhezzar and Houria Siguerdidjane. “Nonlinear Control of a Variable-Speed Wind Turbine Using a Two-Mass Model”. In: *IEEE Transactions on Energy Conversion* 26.1 (2011), pp. 149–162. DOI: 10.1109/TEC.2010.2090155.
- [23] Dimitris Bourlis and J. A. M. Bleijs. “A wind speed estimation method using adaptive Kalman filtering for a variable speed stall regulated wind turbine”. In: *2010 IEEE 11th International Conference on Probabilistic Methods Applied to Power Systems*. 2010, pp. 89–94. DOI: 10.1109/PMAPS.2010.5528980.
- [24] Delphine Bresch-Pietri and Nicolas Petit. “Implicit Integral Equations for Modeling Systems with a Transport Delay”. In: *Recent Results on Time-Delay Systems: Analysis and Control*. Ed. by Emmanuel Witrant, Emilia Fridman, Olivier Sename, and Luc Dugard. Cham: Springer International Publishing, 2016, pp. 3–21. ISBN: 978-3-319-26369-4. DOI: 10.1007/978-3-319-26369-4_1.
- [25] Mona Buisson-Fenet, Valery Morgenthaler, Sebastian Trimpe, and Florent Di Meglio. “Joint State and Dynamics Estimation With High-Gain Observers and Gaussian Process Models”. In: vol. 5. 5. 2021, pp. 1627–1632. DOI: 10.1109/LCSYS.2020.3042412.

- [26] Eduardo F. Camacho, Tariq Samad, Mario Garcia-Sanz, and Ian Hiskens. “Control for renewable energy and smart grids”. In: *The Impact of Control Technology, Control Systems Society* 4.8 (2011), pp. 69–88. URL: <https://ieeecss.org/sites/ieeecss/files/2019-07/IoCT-Part1-06RESG.pdf>.
- [27] H. Camblong. “Digital robust control of a variable speed pitch regulated wind turbine for above rated wind speeds”. In: *Control Engineering Practice* 16.8 (2008). Special Section: IFAC Conference on Analysis and Design of Hybrid Systems (ADHS’06), pp. 946–958. ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2007.11.004.
- [28] Antonello Cherubini, Andrea Papini, Rocco Vertechy, and Marco Fontana. “Airborne Wind Energy Systems: A review of the technologies”. In: *Renewable and Sustainable Energy Reviews* 51 (2015), pp. 1461–1476. ISSN: 1364-0321. DOI: 10.1016/j.rser.2015.07.053.
- [29] Charles-Henri Clerget and Nicolas Petit. “Optimal Control of Systems Subject to Input-Dependent Hydraulic Delays”. In: *IEEE Transactions on Automatic Control* 66.1 (2021), pp. 245–260. DOI: 10.1109/TAC.2020.2982862.
- [30] L. Colombo, M. L. Corradini, G. Ippoliti, and G. Orlando. “Pitch angle control of a wind turbine operating above the rated wind speed: A sliding mode control approach”. In: *ISA Transactions* 96 (2020), pp. 95–102. ISSN: 0019-0578. DOI: 10.1016/j.isatra.2019.07.002.
- [31] Sébastien Da Veiga and Amandine Marrel. “Gaussian process regression with linear inequality constraints”. In: *Reliability Engineering & System Safety* 195 (2020), p. 106732. ISSN: 0951-8320. DOI: 10.1016/j.ress.2019.106732.
- [32] Florent Di Meglio, Delphine Bresch-Pietri, and Ulf Jakob F. Aarsnes. “An adaptive observer for hyperbolic systems with application to UnderBalanced Drilling”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 11391–11397. ISSN: 1474-6670. DOI: 10.3182/20140824-6-ZA-1003.02365.
- [33] *Documentation for fmincon*. <https://mathworks.com/help/optim/ug/fmincon.html>.
- [34] B. M. Doekemeijer, S. Boersma, L. Y. Pao, and J. W. van Wingerden. “Ensemble Kalman filtering for wind field estimation in wind farms”. In: *2017 American Control Conference (ACC)*. 2017, pp. 19–24. DOI: 10.23919/ACC.2017.7962924.
- [35] Bart Doekemeijer and Jan-Willem van Wingerden. “Observability of the ambient conditions in model-based estimation for wind farm control: A focus on static models”. In: *Wind Energy* 23.9 (2020), pp. 1777–1791. DOI: 10.1002/we.2495.

- [36] F. M. Ebrahimi, A. Khayatiyan, and E. Farjah. “A novel optimizing power control strategy for centralized wind farm control system”. In: *Renewable Energy* 86 (2016), pp. 399–408. ISSN: 0960-1481. DOI: 10.1016/j.renene.2015.07.101.
- [37] Paul Fleming, Pieter Gebraad, Matt Churchfield, Sang Lee, Kathryn Johnson, John Michalakes, Jan-Willem van Wingerden, and Patrick Moriarty. *SOWFA + Super Controller User’s Manual*. Tech. rep. National Renewable Energy Lab.(NREL), Golden, CO (United States), 2013. URL: <https://www.nrel.gov/docs/fy13osti/59197.pdf>.
- [38] Emilia Fridman, Catherine Bonnet, Frederic Mazenc, and Walid Djema. “Stability of the cell dynamics in acute myeloid leukemia”. In: *Systems & Control Letters* 88 (2016), pp. 91–100. ISSN: 0167-6911. DOI: 10.1016/j.sysconle.2015.09.006.
- [39] Jean-Paul Gauthier and Ivan Kupka. *Deterministic observation theory and applications*. Cambridge university press, 2001.
- [40] P. M. O. Gebraad, P. A. Fleming, and J. W. van Wingerden. “Wind turbine wake estimation and control using FLORIDyn, a control-oriented dynamic wind plant model”. In: *2015 American Control Conference (ACC)*. 2015, pp. 1702–1708. DOI: 10.1109/ACC.2015.7170978.
- [41] Michael B. Giles and Niles A. Pierce. “An Introduction to the Adjoint Approach to Design”. In: *Flow, turbulence and combustion* 65 (2000), pp. 393–415. DOI: 10.1023/A:1011430410075.
- [42] Dan Givoli. “A tutorial on the adjoint method for inverse problems”. In: *Computer Methods in Applied Mechanics and Engineering* 380 (2021), p. 113810. ISSN: 0045-7825. DOI: 10.1016/j.cma.2021.113810.
- [43] Tuhfe Göçmen, Paul van der Laan, Pierre-Elouan Réthoré, Alfredo Peña Diaz, Gunner Chr. Larsen, and Søren Ott. “Wind turbine wake models developed at the technical university of Denmark: A review”. In: *Renewable and Sustainable Energy Reviews* 60 (2016), pp. 752–769. ISSN: 1364-0321. DOI: 10.1016/j.rser.2016.01.113.
- [44] Jacob Deleuran Grunnet, Mohsen Soltani, Torben Knudsen, Martin Nygaard Kragelund, and Thomas Bak. “Aeolus Toolbox for Dynamics Wind Farm Model, Simulation and Control”. English. In: *European Wind Energy Conference and Exhibition, EWEC 2010*. 2010. URL: <https://vbn.aau.dk/en/publications/aeolus-toolbox-for-dynamics-wind-farm-model-simulationand-control>.

- [45] I. Guenoune, F. Plestan, A. Chermitti, and C. Evangelista. “Modeling and robust control of a twin wind turbines structure”. In: *Control Engineering Practice* 69 (2017), pp. 23–35. ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2017.08.009.
- [46] F. Guillemin, H.-N. Nguyen, G. Sabiron, D. Di Domenico, and M. Boquet. “Real-time three dimensional wind field reconstruction from nacelle LiDAR measurements”. In: vol. 1037. 3. IOP Publishing, June 2018, p. 032037. DOI: 10.1088/1742-6596/1037/3/032037.
- [47] Yujia Hao, Matthew A. Lackner, Rolf-Erik Keck, Sang Lee, Matthew J. Churchfield, and Patrick J. Moriarty. “Implementing the Dynamic Wake Meandering Model in the NWTC Design Codes”. In: *32nd ASME Wind Energy Symposium*. DOI: 10.2514/6.2014-1089.
- [48] Agus Hasan, Ole Morten Aamo, and Miroslav Krstic. “Boundary observer design for hyperbolic PDE–ODE cascade systems”. In: *Automatica* 68 (2016), pp. 75–86. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2016.01.058.
- [49] Dominique P. Held and Jakob Mann. “Lidar estimation of rotor-effective wind speed – an experimental comparison”. In: *Wind Energy Science* 4.3 (2019), pp. 421–438. DOI: 10.5194/wes-4-421-2019.
- [50] A. Honrubia-Escribano, E. Gómez-Lázaro, J. Fortmann, P. Sørensen, and S. Martin-Martinez. “Generic dynamic wind turbine models for power system stability analysis: A comprehensive review”. In: *Renewable and Sustainable Energy Reviews* 81 (2018), pp. 1939–1952. ISSN: 1364-0321. DOI: 10.1016/j.rser.2017.06.005.
- [51] E. L. van der Hooft and T. G. van Engelen. *Estimated wind speed feed forward control for wind turbine operation optimisation*. Nov. 2004. URL: <https://publicaties.ecn.nl/PdfFetch.aspx?nr=ECN-RX--04-126>.
- [52] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. “Multilayer feedforward networks are universal approximators”. In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: 10.1016/0893-6080(89)90020-8.
- [53] Michael F. Howland and John O. Dabiri. “Influence of Wake Model Superposition and Secondary Steering on Model-Based Wake Steering Control with SCADA Data Assimilation”. In: *Energies* 14.1 (2021). ISSN: 1996-1073. DOI: 10.3390/en14010052.
- [54] Michael F. Howland, Aditya S. Ghate, Sanjiva K. Lele, and John O. Dabiri. “Optimal closed-loop wake steering – Part 1: Conventionally neutral atmospheric boundary layer conditions”. In: *Wind Energy Science* 5.4 (2020), pp. 1315–1338. DOI: 10.5194/wes-5-1315-2020.

- [55] Mark Hutchinson and Feng Zhao. *Global Wind Report 2023*. Technical Report. GWEC, Mar. 2023. URL: https://gwec.net/wp-content/uploads/2023/04/GWEC-2023_interactive.pdf.
- [56] Vinay Ingle, Stephen Kogon, and Dimitris Manolakis. *Statistical and adaptive signal processing*. Artech, 2005.
- [57] IRENA Wind energy. <https://web.archive.org/web/20230929092140/https://www.irena.org/Energy-Transition/Technology/Wind-energy>.
- [58] Alberto Isidori. *Lectures in feedback design for multivariable systems*. Springer, 2017. DOI: 10.1007/978-3-319-42031-8.
- [59] Debashisha Jena and Saravanakumar Rajendran. “A review of estimation of effective wind speed based control of wind turbines”. In: *Renewable and Sustainable Energy Reviews* 43 (2015), pp. 1046–1062. ISSN: 1364-0321. DOI: 10.1016/j.rser.2014.11.088.
- [60] Niels Otto Jensen. *A note on wind generator interaction*. Citeseer, 1983.
- [61] Ángel Jiménez, Antonio Crespo, and Emilio Migoya. “Application of a LES technique to characterize the wake deflection of a wind turbine in yaw”. In: *Wind Energy* 13.6 (2010), pp. 559–572. DOI: 10.1002/we.380.
- [62] Kathryn E. Johnson, Lucy Y. Pao, Mark J. Balas, and Lee J. Fingersh. “Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture”. In: *IEEE Control Systems Magazine* 26.3 (2006), pp. 70–81. DOI: 10.1109/MCS.2006.1636311.
- [63] J. Jonkman, P. Doubrawa, N. Hamilton, J. Annoni, and P. Fleming. “Validation of FAST.Farm Against Large-Eddy Simulations”. In: *Journal of Physics: Conference Series* 1037.6 (June 2018), p. 062005. DOI: 10.1088/1742-6596/1037/6/062005.
- [64] J. M. Jonkman and M. L. Buhl Jr. *FAST User’s Guide*. Tech. rep. National Renewable Energy Laboratory, Oct. 2005. URL: <https://www.nrel.gov/docs/fy06osti/38230.pdf>.
- [65] Jason Jonkman, Sandy Butterfield, Walter Musial, and George Scott. *Definition of a 5-MW reference wind turbine for offshore system development*. Tech. rep. National Renewable Energy Laboratory (NREL), 2009. DOI: 10.2172/947422.
- [66] Jason M. Jonkman, Jennifer Annoni, Greg Hayman, Bonnie Jonkman, and Avi Purkayastha. “Development of FAST.Farm: A new multi-physics engineering tool for wind-farm design and analysis”. In: *35th Wind Energy Symposium*. 2017. DOI: 10.2514/6.2017-0454.

- [67] Jason Mark Jonkman and Kelsey Shaler. *FAST.Farm user's guide and theory manual*. National Renewable Energy Laboratory Golden, CO, 2021. URL: <https://www.nrel.gov/docs/fy21osti/78485.pdf>.
- [68] John K. Kaldellis, Panagiotis Triantafyllou, and Panagiotis Stinis. "Critical evaluation of Wind Turbines' analytical wake models". In: *Renewable and Sustainable Energy Reviews* 144 (2021), p. 110991. ISSN: 1364-0321. DOI: 10.1016/j.rser.2021.110991.
- [69] Hassan K. Khalil. "High-gain observers in nonlinear feedback control". In: *2009 IEEE International Conference on Control and Automation*. 2009, pp. 1527–1528. DOI: 10.1109/ICCA.2009.5410227.
- [70] Hassan K. Khalil. *High-gain observers in nonlinear feedback control*. Society for Industrial and Applied Mathematics, 2017. ISBN: 978-1-611974-85-0.
- [71] Hassan K. Khalil and Laurent Praly. "High-gain observers in nonlinear feedback control". In: *International Journal of Robust and Nonlinear Control* 24.6 (2014), pp. 993–1015. DOI: 10.1002/rnc.3051.
- [72] Torben Knudsen, Thomas Bak, and Mikael Svenstrup. "Survey of wind farm control—power and fatigue optimization". In: *Wind Energy* 18.8 (2015), pp. 1333–1351. DOI: 10.1002/we.1760.
- [73] Matthias Kretschmer, Jason Jonkman, Vasilis Pettas, and Po Wen Cheng. "FAST.Farm load validation for single wake situations at alpha ventus". In: *Wind Energy Science* 6.5 (2021), pp. 1247–1262. DOI: 10.5194/wes-6-1247-2021.
- [74] Gunner C. Larsen, H. Aagaard Madsen, and F. Bingöel. *Dynamic wake meandering modeling*. June 2007. URL: <https://www.osti.gov/etdeweb/servlets/purl/20941220>.
- [75] Gunner C. Larsen, Helge Aa. Madsen, Kenneth Thomsen, and Torben J. Larsen. "Wake meandering: a pragmatic approach". In: *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology* 11.4 (2008), pp. 377–395. DOI: 10.1002/we.267.
- [76] Jeremy Ledoux, Sebastián Rizzo, and Julien Salomon. "Analysis of the blade element momentum theory". In: *SIAM Journal on Applied Mathematics* 81.6 (2021), pp. 2596–2621. DOI: 10.1137/20M133542X.
- [77] Hui Li, K. L. Shi, and P. G. McLaren. "Neural-network-based sensorless maximum wind energy capture with compensated power coefficient". In: *IEEE Transactions on Industry Applications* 41.6 (2005), pp. 1548–1556. DOI: 10.1109/TIA.2005.858282.
- [78] Jacques Louis Lions. *Optimal control of systems governed by partial differential equations*. Vol. 170. Springer, 1971.

- [79] Jing Liu, Htet Lin, and Jun Zhang. “Review on the technical perspectives and commercial viability of vertical axis wind turbines”. In: *Ocean Engineering* 182 (2019), pp. 608–626. ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2019.04.086.
- [80] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.
- [81] B. Maronga, M. Gryschka, R. Heinze, F. Hoffmann, F. Kanani-Sühring, M. Keck, K. Ketelsen, M. O. Letzel, M. Sühring, and S. Raasch. “The Parallelized Large-Eddy Simulation Model (PALM) version 4.0 for atmospheric and oceanic flows: model formulation, recent developments, and future perspectives”. In: *Geoscientific Model Development* 8.8 (2015), pp. 2515–2551. DOI: 10.5194/gmd-8-2515-2015.
- [82] Luis A. Martínez-Tossas, Jennifer Annoni, Paul A. Fleming, and Matthew J. Churchfield. “The aerodynamics of the curled wake: a simplified model in view of flow control”. In: *Wind Energy Science* 4.1 (2019), pp. 127–138. DOI: 10.5194/wes-4-127-2019.
- [83] Luis A. Martínez-Tossas, Matthew J. Churchfield, and Stefano Leonardi. “Large eddy simulations of the flow past wind turbines: actuator line and disk modeling”. In: *Wind Energy* 18.6 (2015), pp. 1047–1060. DOI: 10.1002/we.1747.
- [84] Jana Mayer, Ajit Basarur, Mariana Petrova, Fabian Sordon, Antonio Zea, and Uwe D. Hanebeck. “Position and Speed Estimation of PMSMs Using Gaussian Processes”. In: *21st IFAC World Congress*. 2020. DOI: 10.1016/j.ifacol.2020.12.261.
- [85] Sarah Mechhoud, Emmanuel Witrant, and Luc Dugard. “Adaptive space-time distributed parameter and input estimation in heat transport with unknown bounds”. In: *3rd International Conference on Systems and Control*. 2013, pp. 671–676. DOI: 10.1109/ICoSC.2013.6750931.
- [86] Lucie Membrado, Delphine Demeautis, Cédric Ramauge, Julien Taisne, and Céline Bonnet. *Projet de parc éolien d’Ablaincourt*. <https://www.somme.gouv.fr/contenu/telechargement/18292/127719/file/RNTEIE-Valorem-Ablaincourt-150610.pdf>.
- [87] Eduardo José Novaes Menezes, Alex Maurício Araújo, and Nadège Sophie Bouchonneau da Silva. “A review on wind turbine control and its associated methods”. In: *Journal of Cleaner Production* 174 (2018), pp. 945–953. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2017.10.297.

- [88] Johan Meyers, Carlo Bottasso, Katherine Dykes, Paul Fleming, Pieter Gebraad, Gregor Giebel, Tuhfe Göçmen, and Jan-Willem van Wingerden. “Wind farm flow control: prospects and challenges”. In: *Wind Energy Science Discussions* 2022 (2022), pp. 1–56. DOI: 10.5194/wes-7-2271-2022.
- [89] Johan Meyers and Charles Meneveau. “Large Eddy Simulations of large wind-turbine arrays in the atmospheric boundary layer”. In: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. 2010. DOI: 10.2514/6.2010-827.
- [90] Louis Melville Milne-Thomson. *Theoretical aerodynamics*. Courier Corporation, 1973.
- [91] Mahmood Mirzaei, Hans Henrik Niemann, and Niels Kjølstad Poulsen. “A μ -synthesis approach to robust control of a wind turbine”. In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. 2011, pp. 645–650. DOI: 10.1109/CDC.2011.6161124.
- [92] Iulian Munteanu and Gildas Besançon. “Control-based strategy for effective wind speed estimation in wind turbines”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 6776–6781. ISSN: 1474-6670. DOI: 10.3182/20140824-6-ZA-1003.01952.
- [93] Iulian Munteanu and Gildas Besançon. “Identification-based prediction of wind park power generation”. In: *Renewable Energy* 97 (2016), pp. 422–433. ISSN: 0960-1481. DOI: 10.1016/j.renene.2016.05.088.
- [94] Yoon-Su Nam, Jeong-Gi Kim, In-Su Paek, Young-Hwan Moon, Seog-Joo Kim, and Dong-Joon Kim. “Feedforward pitch control using wind speed estimation”. In: *Journal of Power Electronics* 11.2 (2011), pp. 211–217. DOI: doi:10.6113/jpe.2011.11.2.211.
- [95] Van Tri Nguyen, Didier Georges, and Gildas Besançon. “Adjoint-method-based estimation of Manning roughness coefficient in an overland flow model”. In: *2015 American Control Conference (ACC)*. 2015, pp. 1977–1982. DOI: 10.1109/ACC.2015.7171023.
- [96] Van Tri Nguyen, Didier Georges, and Gildas Besançon. “State and parameter estimation in 1-D hyperbolic PDEs based on an adjoint method”. In: *Automatica* 67 (2016), pp. 185–191. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2016.01.031.
- [97] *OpenFAST Documentation*. <https://openfast.readthedocs.io/en/main/>.
- [98] *OpenFOAM*. <https://www.openfoam.com/>.
- [99] Alan V. Oppenheim, Alan S. Willsky, Syed Hamid Nawab, and Jian-Jiun Ding. *Signals and systems*. Vol. 2. Prentice hall Upper Saddle River, NJ, 1997.

- [100] Eduardo B. R. F. Paiva, Olivier Lepreux, and Delphine Bresch-Pietri. “Free-Flow Wind Speed Estimation for a Wind Turbine Affected by Wake”. In: *2023 American Control Conference (ACC)*. 2023, pp. 171–176. DOI: 10.23919/ACC55779.2023.10156025.
- [101] Eduardo B. R. F. Paiva, Olivier Lepreux, and Delphine Bresch-Pietri. “Transport Speed Estimation for a 1-D Hyperbolic PDE Based on Output Flow Measurement”. In: *2022 10th International Conference on Systems and Control (ICSC)*. 2022, pp. 72–77. DOI: 10.1109/ICSC57768.2022.9993925.
- [102] Eduardo B. R. F. Paiva, Hoai-Nam Nguyen, Olivier Lepreux, and Delphine Bresch-Pietri. “A Gaussian Process Based Approach to Estimate Wind Speed Using SCADA Measurements from a Wind Turbine”. In: *IFAC-PapersOnLine* 54.20 (2021). Modeling, Estimation and Control Conference MECC 2021, pp. 65–71. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2021.11.154.
- [103] Jayshree Pande, Paresh Nasikkar, Ketan Kotecha, and Vijayakumar Varadarajan. “A Review of Maximum Power Point Tracking Algorithms for Wind Energy Conversion Systems”. In: *Journal of Marine Science and Engineering* 9.11 (2021). ISSN: 2077-1312. DOI: 10.3390/jmse9111187.
- [104] Lucy Y. Pao and Kathryn E. Johnson. “Control of Wind Turbines”. In: *IEEE Control Systems Magazine* 31.2 (2011), pp. 44–62. DOI: 10.1109/MCS.2010.939962.
- [105] R. H. Perry, D. W. Green, and J. O. Maloney. *Perry’s Chemical Engineers’ Handbook*. Vol. 7. McGraw-Hill New York, 1984.
- [106] F. Plestan, C. Evangelista, P. Puleston, and I. Guenoune. “Control of a twin wind turbines system without wind velocity information”. In: *2018 15th International Workshop on Variable Structure Systems (VSS)*. 2018, pp. 150–155. DOI: 10.1109/VSS.2018.8460274.
- [107] Fernando Porté-Agel, Majid Bastankhah, and Sina Shamsoddin. “Wind-turbine and wind-farm flows: A review”. In: *Boundary-layer meteorology* 174.1 (2020), pp. 1–59. DOI: 10.1007/s10546-019-00473-0.
- [108] Manish Ram, Michael Child, Arman Aghahosseini, Dmitrii Bogdanov, Alena Lohrmann, and Christian Breyer. “A comparative analysis of electricity generation costs from renewable, fossil fuel and nuclear sources in G20 countries for the period 2015-2030”. In: *Journal of Cleaner Production* 199 (2018), pp. 687–704. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2018.07.159.
- [109] Jean-Pierre Richard. “Time-delay systems: an overview of some recent advances and open problems”. In: *Automatica* 39.10 (2003), pp. 1667–1694. DOI: 10.1016/S0005-1098(03)00167-5.

- [110] Jaakko Riihimäki and Aki Vehtari. “Gaussian processes with monotonicity information”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 645–652. URL: <https://proceedings.mlr.press/v9/riihimaki10a.html>.
- [111] B. Sanderse, S P. van der Pijl, and B. Koren. “Review of computational fluid dynamics for wind turbine wake aerodynamics”. In: *Wind Energy* 14.7 (2011), pp. 799–819. DOI: 10.1002/we.458.
- [112] John Sandoval-Moreno, Gildas Besançon, and John J. Martinez. “Observer-based maximum power tracking in wind turbines with only generator speed measurement”. In: *2013 European Control Conference (ECC)*. 2013, pp. 478–483. DOI: 10.23919/ECC.2013.6669847.
- [113] Daniel Sbarbaro and Ruben Peña. “A nonlinear wind velocity observer for a small wind energy system”. In: *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*. Vol. 4. 2000, 3086–3087 vol.4. DOI: 10.1109/CDC.2000.912169.
- [114] J. G. Schepers and S. P. van der Pijl. “Improved modelling of wake aerodynamics and assessment of new farm control strategies”. In: *Journal of Physics: Conference Series* 75.1 (July 2007), p. 012039. DOI: 10.1088/1742-6596/75/1/012039.
- [115] David Schlipf, Paul Fleming, Stefan Kapp, Andrew Scholbrock, Florian Haizmann, Fred Belen, Alan Wright, and Po Wen Cheng. “Direct Speed Control Using LIDAR and Turbine Data”. In: *2013 American Control Conference*. 2013, pp. 2208–2213. DOI: 10.1109/ACC.2013.6580163.
- [116] Eric Schulz, Maarten Speekenbrink, and Andreas Krause. “A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions”. In: *Journal of Mathematical Psychology* 85 (2018), pp. 1–16. ISSN: 0022-2496. DOI: 10.1016/j.jmp.2018.03.001.
- [117] K. Shaler, M. Debnath, and J. Jonkman. “Validation of FAST.Farm Against Full-Scale Turbine SCADA Data for a Small Wind Farm”. In: vol. 1618. 6. IOP Publishing, Sept. 2020, p. 062061. DOI: 10.1088/1742-6596/1618/6/062061.
- [118] Kelsey Shaler and Jason Jonkman. “FAST.Farm development and validation of structural load prediction against large eddy simulations”. In: *Wind Energy* 24.5 (2021), pp. 428–449. DOI: 10.1002/we.2581.
- [119] Carl R. Shapiro, Pieter Bauweraerts, Johan Meyers, Charles Meneveau, and Dennice F. Gayme. “Model-based receding horizon control of wind farms for secondary frequency regulation”. In: *Wind Energy* 20.7 (2017), pp. 1261–1275. DOI: 10.1002/we.2093.

- [120] Carl R. Shapiro, Dennice F. Gayme, and Charles Meneveau. “Modelling yawed wind turbine wakes: a lifting line approach”. In: *Journal of Fluid Mechanics* 841 (2018). DOI: 10.1017/jfm.2018.75.
- [121] Carl R. Shapiro, Johan Meyers, Charles Meneveau, and Dennice F. Gayme. “Dynamic wake modeling and state estimation for improved model-based receding horizon control of wind farms”. In: *2017 American Control Conference (ACC)*. 2017, pp. 709–716. DOI: 10.23919/ACC.2017.7963036.
- [122] Carl R. Shapiro, Genevieve M. Starke, Charles Meneveau, and Dennice F. Gayme. “A Wake Modeling Paradigm for Wind Farm Design and Control”. In: *Energies* 12.15 (2019), p. 2956. DOI: 10.3390/en12152956.
- [123] Jens Nørkær Sørensen and Gunner Christian Larsen. “A Minimalistic Prediction Model to Determine Energy Production and Costs of Offshore Wind Farms”. In: *Energies* 14.2 (2021). ISSN: 1996-1073. DOI: 10.3390/en14020448.
- [124] M. Steinbuch, W. W. de Boer, O. H. Bosgra, S. A. W. M. Peeters, and J. Ploeg. “Optimal control of wind power plants”. In: *Journal of Wind Engineering and Industrial Aerodynamics* 27.1-3 (1988), pp. 237–246. DOI: 10.1016/0167-6105(88)90039-6.
- [125] Anja Stieren, Srinidhi N. Gadde, and Richard J. A. M. Stevens. “Modeling dynamic wind direction changes in large eddy simulations of wind farms”. In: *Renewable Energy* 170 (2021), pp. 1342–1352. ISSN: 0960-1481. DOI: 10.1016/j.renene.2021.02.018.
- [126] R. Tandel, S. Shah, and S. Tripathi. “A state-of-art review on Bladeless Wind Turbine”. In: *Journal of Physics: Conference Series* 1950.1 (Aug. 2021), p. 012058. DOI: 10.1088/1742-6596/1950/1/012058.
- [127] *The Wind Power*. https://web.archive.org/web/20231006145148/https://www.thewindpower.net/turbine_en_464_senvion_mm82-2050.php. Accessed: 2023-10-06.
- [128] Chris Vermillion, Mitchell Cobb, Lorenzo Fagiano, Rachel Leuthold, Moritz Diehl, Roy S. Smith, Tony A. Wood, Sebastian Rapp, Roland Schmehl, David Olinger, and Michael Demetriou. “Electricity in the air: Insights from two decades of advanced control research and experimental flight testing of airborne wind energy systems”. In: *Annual Reviews in Control* 52 (2021), pp. 330–357. ISSN: 1367-5788. DOI: 10.1016/j.arcontrol.2021.03.002.
- [129] Haiyang Wen, Shi Shu, and Liping Wen. “A new generalization of Halanay-type inequality and its applications”. In: *Journal of Inequalities and Applications* 2018.1 (2018), pp. 1–12. DOI: 10.1186/s13660-018-1894-5.

- [130] Christopher K. I. Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.
- [131] Svante Wold, Kim Esbensen, and Paul Geladi. “Principal Component Analysis”. In: *Chemometrics and Intelligent Laboratory Systems* 2.1 (1987). Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists, pp. 37–52. ISSN: 0169-7439. DOI: 10.1016/0169-7439(87)80084-9.
- [132] Xiyun Yang, Xiaojuan Han, Lingfeng Xu, and Yibing Liu. “Soft Sensor Based on Support Vector Machine for Effective Wind Speed in Large Variable Wind”. In: *2006 9th International Conference on Control, Automation, Robotics and Vision*. 2006, pp. 1–4. DOI: 10.1109/ICARCV.2006.345278.
- [133] Kai Zenger and Antti J. Niemi. “Modelling and control of a class of time-varying continuous flow processes”. In: *Journal of Process Control* 19.9 (2009), pp. 1511–1518. DOI: 10.1016/j.jprocont.2009.07.008.
- [134] Cheng Zhang and Franck Plestan. “Individual/collective blade pitch control of floating wind turbine based on adaptive second order sliding mode”. In: *Ocean Engineering* 228 (2021), p. 108897. ISSN: 0029-8018. DOI: 10.1016/j.oceaneng.2021.108897.
- [135] Haohua Zong and Fernando Porté-Agel. “A momentum-conserving wake superposition method for wind farm power prediction”. In: *Journal of Fluid Mechanics* 889 (2020). DOI: 10.1017/jfm.2020.77.

RÉSUMÉ

Cette thèse propose des algorithmes pour estimer la vitesse et la direction du vent pour des éoliennes et des parcs éoliens.

Tout d'abord, nous proposons des méthodes basées sur les données pour estimer la vitesse effective du rotor (REWS) sans nécessiter la connaissance de certains paramètres physiques de l'éolienne, qui pourraient être inconnus de l'opérateur. Nous fournissons deux méthodes basées sur les données, l'une basée sur la régression par processus gaussien et l'autre combinant la régression par processus gaussien avec un observateur grand gain.

Ensuite, en nous basant sur cette estimation locale de la REWS, au niveau d'une éolienne, nous abordons la question de l'estimation du vent en écoulement libre au niveau du parc éolien.

Nous commençons par nous concentrer sur l'estimation de la vitesse du vent, pour une direction du vent connue. Pour un parc éolien de géométrie simple, nous démontrons qu'une mesure locale de la vitesse perturbée par la présence des éoliennes peut être utilisée pour estimer la vitesse du vent en écoulement libre. Nous fondons notre méthodologie d'estimation sur une modélisation simplifiée de l'effet de sillage qui consiste en des équations aux dérivées partielles hyperboliques du premier ordre en cascade, et dont la vitesse de transport est la vitesse du vent en écoulement libre. Nous proposons d'utiliser une solution analytique de ces équations, impliquant des retards de transport, pour effectuer une estimation de la mesure locale et mettre à jour l'estimation de la vitesse du vent en écoulement libre. Nous démontrons formellement la convergence de cette estimation et illustrons numériquement l'efficacité de cette méthode. Enfin, nous passons à une configuration plus générale où à la fois la vitesse et la direction du vent en écoulement libre sont inconnues. Nous proposons d'utiliser une modélisation bidimensionnelle du sillage et de nous appuyer sur une méthode basée sur l'optimisation. Le problème d'identification que nous formulons se révèle être particulièrement difficile en raison de l'apparition de retards de transport, mais nous montrons comment contourner cette difficulté en considérant une valeur moyenne de l'historique de la vitesse du vent en écoulement libre. Des résultats de simulation obtenus avec le simulateur FAST.Farm illustrent l'intérêt de la méthode proposée.

MOTS CLÉS

estimation de la vitesse du vent, estimation de parc éolien, énergie éolienne, régression par processus gaussien, observateur grand-gain, retards de transport, équations aux dérivées partielles hyperboliques du premier ordre.

ABSTRACT

This thesis designs algorithms to estimate the wind speed and direction for wind turbines and wind farms.

First, we propose data-based methods to estimate the Rotor Effective Wind Speed (REWS) for a single turbine without prior knowledge of certain physical parameters of the turbine that might be unknown to an operator. We provide two data-based methods, based respectively on Gaussian Process Regression (GPR) and on a combination of GPR with high-gain observers.

Second, grounding on this REWS estimation at the local level of one turbine, we address the question of estimating the free-flow wind at the level of a wind farm.

We start by focusing on wind speed estimation, for a given known wind direction. For a wind farm with a simple geometry, we prove that a local speed measurement disturbed by the presence of the turbines can be used to estimate the free-flow wind speed. We ground our estimation methodology on a simplified wake model, which consists of first-order hyperbolic partial differential equations, the transport speed of which is the free-flow wind speed. We propose to use an analytical solution of these equations, involving transport delays, to perform an estimate of the local measurement and to update the free-flow wind speed estimate. We formally prove the convergence of this estimate and numerically illustrate the efficiency of this method.

Finally, we move to a more general setup where both the free-flow wind speed and direction are unknown. We propose to use a two-dimensional wake model and to rely on an optimization-based method. This identification problem reveals to be particularly challenging due to the appearance of transport delays, but we illustrate how to circumvent this issue by considering an average value of the free flow wind speed history. Simulation results obtained with the simulator FAST.Farm illustrate the interest of the proposed method.

KEYWORDS

wind speed estimation, wind farm estimation, wind energy, Gaussian process regression, high-gain observer, transport delays, first-order hyperbolic partial differential equations.