本附件下附文献为风电行业常用的一款基于非线性模型的开源仿真工具SimWindFarm所发表的**期刊**论文，及一份测试报告的部分内容（内容来源于欧盟资助的CLwindcon D1.2项目）。论文详细介绍了SimWindFarm的结构、原理和相应公式。SimWindFarm工具基于Matlab/Simulink搭建，其所能够体现的动态过程可覆盖本题目所有仿真需求。该工具下载链接为：https://github.com/kihso/MRTCC/tree/master/simwindfarm-v1.1。

需要注意，若选手直接采用此模型作为问题二的答案，评分时会酌情减分。

需要说明，附属论文中的Fig 3展示了风机模型输入输出结构。其中，v_rot是风轮处风速；v_nac是机舱处风速；二者存在微小差异。本赛题仅考虑功率的优化调度，因此可将v_rot和v_nac近似认为是相同值，即赛题中提到的风速。

由于风电场不存在燃料消耗的成本，其运营成本主要来自设备损失等维护成本[5-6]，因此可将AGC 问题与降低设备的疲劳水平相结合，以减少运行过程中设备损失带来的成本[7]。现有研究提出多种表征风电机组疲劳的表征方式。文献[7-8]通过构建成本函数来表征相关的运行成本。文献[9-10]认为主轴转矩和塔弯矩的标准差可用作风电机组的疲劳评估指标。文献[11]指出，主轴转矩和塔弯矩的标准差确实与机组的疲劳相关，但它们之间并非简单的线性相关。一些研究人员采用了损伤等效载荷（Damage Equivalent Load, DEL）作为疲劳程度评估指标[12]。DEL 在工程应用中得到了广泛认可，但是DEL 的计算过程复杂，难以用于在线优化过程。

通过机理模型分析可知，风电机组的主轴转矩、塔弯矩及 DEL 等用于表征疲劳的参数均与机组转子转速和桨距角直接相关，而风电机组 AGC 的参数也是转速和桨距角。因此在 AGC 控制过程中完成疲劳载荷抑制的研究思路是合乎物理规律的。针对风电机组的 AGC，现有文献中常用的控制方法包括转子转速控制[13-14]、桨距角控制[15]以及两种控制的协调控制[16-17]。为了获得更好的控制效果，文献[18]提出同时激活桨距角和转子转速控制，桨距角控制用于在较大的时间尺度内跟踪参考功率，转子转速用于在较小的时间尺度内精确调整功率。文献[19]提出了一种类似的协调控制策略，认为应进一步使用转子惯性减少桨距角伺服系统的动作，从而降低其疲劳程度。然而，现有文献中，转子转速和桨距角的协调控制涉及多个控制器的协调，并不能通过单一控制器同时进行调整，桨距角需要在转子转速达到设定的阈值后才开始调整[20]，牺牲了桨距角调整的自由度。

由于风电场需要控制多个分布式单元机组，AGC 过程除了需要考虑风电机组的控制，风电场层面的优化同样重要[21]。在风电场层面，将电网运营商所需的有功功率合理地分配给风电机组是目前研究热点[22-23]。文献[9]对风电机组进行机组疲劳进行建模，然后采用模型预测控制（Model Predictive Control, MPC）实现考虑疲劳优化的机组功率分配。文献[10]直接利用机理模型获得有功功率与疲劳参数的偏导数关系，从而通过求解器获得优化疲劳分布的风电机组功率指令。然而以上文献中，研究人员采用的均为集中式控制方法，计算复杂度较高，且在风电机组层面均只采用了传统的转速和桨距角协调控制方法。文献[24-25]中利用多智能体系统完成了风电场有功功率的分布式调度，机组之间通过相互通信即可完成调度指令的分配。文献[6]提出一种基于分布式信息同步和估计的模型预测控制算法，用于对风电场有功功率和电压同时进行优化。文献[26]在等比例分配的策略下利用分布式的对偶梯度法完成了风电场有功功率的优化控制。然而，与文献[9-10]中存在的问题相似，以上分布式控制算法也是在传统风电机组协调控制模型的基础上完成的，无法实现利用转子转速和桨距角的同时优化控制来改善机组疲劳特性。

针对以上研究现状，本文提出一种可同时调整转子转速和桨距角的风电机组 AGC 模型。基于改进的机组控制框架，设计了一种基于交替方向乘子法（Alternating Direction Multiplier Method, ADMM）的分布式模型预测控制（Distributed Model Predictive Control, DMPC）框架来完成风电场功率控制，确保场内风电机组相互协调，针对 AGC 指令跟踪和疲劳载荷抑制的目标完成转子转速和桨距角的同时控制。在疲劳载荷抑制目标方面，本文结合当前研究成果，以降低主轴转矩和塔弯矩的波动性作为在线优化目标，以 DEL 为算法优化效果的评价指标。仿真实验验证了本文提出的风电机组控制模型的准确性和风电场分布式 AGC 算法的有效性。

1 风电机组建模

1.1 风电机组非线性模型

风电机组建模是相关控制器设计的基础，对于进一步提高风电机组控制性能和风电场运行水平具有重要意义。风电机组的建模对象主要包括气动系统、传动系统、变桨伺服系统、电气系统和控制系统。除控制系统外，对于其他子系统的建模已有大量文献进行了研究。各个子系统之间的关联如图 1 所示。本文以 NREL5MW 风电机组模型为例，得到各个子系统的数学模型。

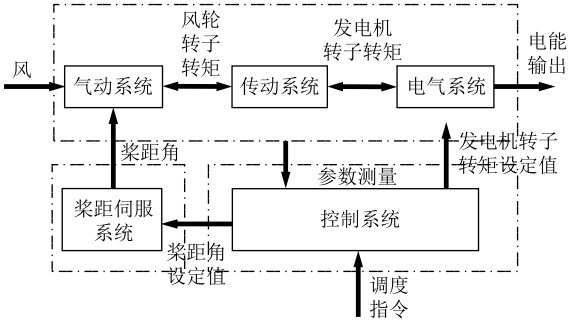

图 1 风电机组子系统的相互关联

Fig.1 Interconnection of wind turbine subsystems

气动系统

$$P_{\mathrm{m}}=0.5\pi\rho R^2C_{\mathrm{p}}(\lambda,\beta)v^3 \tag{1}$$

式中，P_{m} 为风轮机吸收的机械功率（W）；ρ 为空气密度（kg/m³）；R 为风轮半径（m）；λ 为叶尖速比；β 为桨距角（°）；v 为风速（m/s）；C_{p} 为风能捕获系数。

$$F_{\mathrm{t}}=0.5\pi\rho R^2C_{\mathrm{t}}(\lambda,\beta)v^2 \tag{2}$$

式中，F_{t} 为风电机组承受的推力（N）；C_{t} 为推力系数。

$$\lambda=\frac{\omega_{\mathrm{r}}R}{v} \tag{3}$$

式中，ω_{r} 为转子转速（rad/s）。

$$C_{\mathrm{p}}=0.5176\left(116\frac{1}{\lambda^*}-0.4\beta-5\right)\mathrm{e}^{-\frac{21}{\lambda^*}}+0.0068\lambda \tag{4}$$

$$\frac{1}{\lambda^*}=\frac{1}{\lambda+0.08\beta}-\frac{0.035}{\beta^3+1} \tag{5}$$

$$M_{\mathrm{t}}=hF_{\mathrm{t}} \tag{6}$$

式中，h 为塔架高度；M_{t} 为塔根弯矩（N·m）。

传动系统

$$J_{\mathrm{t}}\dot{\omega}_{\mathrm{r}}=T_{\mathrm{m}}-N_{\mathrm{gear}}T_{\mathrm{e}} \tag{7}$$

$$T_{\mathrm{s}}=T_{\mathrm{m}}-J_{\mathrm{m}}\dot{\omega}_{\mathrm{r}} \tag{8}$$

式中，J_{m} 为风轮转子的转动惯量（kg·m²）；J_{t} 为传动系统等效的转动惯量（kg·m²），$J_{\mathrm{t}}=J_{\mathrm{m}}+N_{\mathrm{gear}}^2J_{\mathrm{e}}$，$J_{\mathrm{e}}$ 为发电机转子的转动惯量（kg·m²）；T_{m} 为机械转矩（N·m）；T_{e} 为电磁转矩（N·m）；T_{s} 为主轴转矩（N·m）；N_{gaer} 为齿轮箱变速比。

电气系统

$$T_{\mathrm{e}}+\tau_{\mathrm{e}}\dot{T}_{\mathrm{e}}=T_{\mathrm{e}}^{\mathrm{ref}} \tag{9}$$

式中，τ_{e} 为发电机惯性时间常数（s）；$T_{\mathrm{e}}^{\mathrm{ref}}$ 为电磁转矩设定值（N·m）。

桨距伺服系统

$$\dot{\beta}=\frac{1}{\tau_\beta}(\beta^{\mathrm{ref}}-\beta) \tag{10}$$

式中，β^{ref} 为桨距角设定值（°）；τ_β 桨距角伺服系统惯性时间常数（s）。

1.2　风电机组模型线性化　线性化精度需要提升

根据 1.1 节描述的数学模型，由于风电机组中存在高度的非线性特性，难以用于适合线性模型的先进控制算法。因此，本文通过小信号法[27]对上述模型进行线性化处理，得到线性化描述的风电机组模型。

气动系统

$$\Delta T_{\mathrm{m}}=\frac{\partial T_{\mathrm{m}}}{\partial\omega_{\mathrm{r}}}\Delta\omega_{\mathrm{r}}+\frac{\partial T_{\mathrm{m}}}{\partial v}\Delta v+\frac{\partial T_{\mathrm{m}}}{\partial\beta}\Delta\beta \tag{11}$$

$$\Delta M_{\mathrm{t}}=h\Delta F_{\mathrm{t}}=\frac{\partial M_{\mathrm{t}}}{\partial\omega_{\mathrm{r}}}\Delta\omega_{\mathrm{r}}+\frac{\partial M_{\mathrm{t}}}{\partial v}\Delta v+\frac{\partial M_{\mathrm{t}}}{\partial\beta}\Delta\beta \tag{12}$$

传动系统

$$\Delta\dot{\omega}_{\mathrm{r}}=\frac{1}{J_{\mathrm{t}}}\left(\Delta T_{\mathrm{m}}-N_{\mathrm{gear}}\Delta T_{\mathrm{e}}\right) \tag{13}$$

$$\Delta T_{\mathrm{s}}=\frac{N_{\mathrm{gear}}^2J_{\mathrm{e}}}{J_{\mathrm{t}}}\Delta T_{\mathrm{m}}+\frac{N_{\mathrm{gear}}J_{\mathrm{m}}}{J_{\mathrm{t}}}\Delta T_{\mathrm{e}} \tag{14}$$

电气系统

$$\Delta\dot{T}_{\mathrm{e}}=\frac{1}{\tau_{\mathrm{e}}}(\Delta T_{\mathrm{e}}^{\mathrm{ref}}-\Delta T_{\mathrm{e}}) \tag{15}$$

桨距伺服系统

$$\Delta\dot{\beta}=\frac{1}{\tau_\beta}(\Delta\beta^{\mathrm{ref}}-\Delta\beta) \tag{16}$$

根据式（1），计算各项偏微分可以得到 $\partial T_{\mathrm{m}}/\partial\omega_{\mathrm{r}}$、$\partial T_{\mathrm{m}}/\partial v$ 和 $\partial T_{\mathrm{m}}/\partial\beta$ 的表达式分别为

$$\left.\frac{\partial T_{\mathrm{m}}}{\partial\omega_{\mathrm{r}}}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0}=-0.5\pi\rho\omega_{\mathrm{r}0}^{-2}Rv_0^3\left(C_{\mathrm{p}0}v_0-\omega_{\mathrm{r}0}R\left.\frac{\partial C_{\mathrm{p}}}{\partial\lambda}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0}\right) \tag{17}$$

$$\left.\frac{\partial T_{\mathrm{m}}}{\partial v}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0}=\pi\rho R^2v_0\left(\omega_{\mathrm{r}0}^{-1}v_0C_{\mathrm{p}0}-0.5R\left.\frac{\partial C_{\mathrm{p}}}{\partial\lambda}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0}\right) \tag{18}$$

$$\left.\frac{\partial T_{\mathrm{m}}}{\partial\beta}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0}=0.5\pi\rho\omega_{\mathrm{r}0}^{-1}R^2v_0^3\left.\frac{\partial C_{\mathrm{p}}}{\partial\beta}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0} \tag{19}$$

式中，下标为 0 代表线性化模型的初始工况点。

同理，根据式（2）和式（10）可以计算偏微分得到 $\partial M_{\mathrm{t}}/\partial\omega_{\mathrm{r}}$、$\partial M_{\mathrm{t}}/\partial v$ 和 $\partial M_{\mathrm{t}}/\partial\beta$ 的表达式分别为

$$\left.\frac{\partial M_{\mathrm{t}}}{\partial\omega_{\mathrm{r}}}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0}=-0.5\pi h\rho R^3v_0\left.\frac{\partial C_{\mathrm{t}}}{\partial\lambda}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0} \tag{20}$$

$$\left.\frac{\partial M_{\mathrm{t}}}{\partial v}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0}=\pi h\rho R^2\left(v_0-0.5R\omega_{\mathrm{r}0}\left.\frac{\partial C_{\mathrm{t}}}{\partial\lambda}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0}\right) \tag{21}$$

$$\left.\frac{\partial M_{\mathrm{t}}}{\partial\beta}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0}=0.5\pi h\rho R^2v_0^2\left.\frac{\partial C_{\mathrm{t}}}{\partial\beta}\right|_{v_0,\omega_{\mathrm{r}0},\beta_0} \tag{22}$$

值得注意的是，根据式（4）和式（5），C_p具有高度非线性特性。为了进行线性化，本文根据 C_p-λ-β 曲线查表计算 C_p的偏导数。假设工作点的 C_p值为 C_{p0}，则通过查表法可获得其周围的插值点，如图 2 所示。

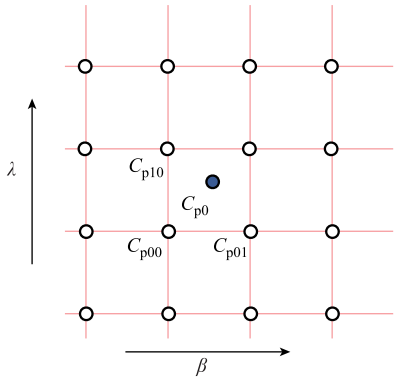

图 2 C_p的局部插值示意图

Fig. 2 Schematic diagram of local interpolation of C_p

进一步地，可以得到偏导数 $\partial C_p/\partial\lambda$ 和 $\partial C_p/\partial\beta$ 的表达式为

$$\left.\frac{\partial C_p}{\partial\lambda}\right|_{v_0,\omega_{t0},\beta_0}=\frac{C_{p10}-C_{p00}}{\Delta\lambda} \tag{23}$$

$$\left.\frac{\partial C_p}{\partial\beta}\right|_{v_0,\omega_{t0},\beta_0}=\frac{C_{p01}-C_{p00}}{\Delta\beta} \tag{24}$$

同理可得到 C_t的相关偏导数的表达式为

$$\left.\frac{\partial C_t}{\partial\lambda}\right|_{v_0,\omega_{t0},\beta_0}=\frac{C_{t10}-C_{t00}}{\Delta\lambda} \tag{25}$$

$$\left.\frac{\partial C_t}{\partial\beta}\right|_{v_0,\omega_{t0},\beta_0}=\frac{C_{t01}-C_{t00}}{\Delta\beta} \tag{26}$$

2 风电机组多入多出 AGC 控制模型

风轮机从风能中捕获得到的机械功率可以由式（1）模型进行表示。根据该模型，若风电机组需要运行于 AGC 状态，仅需调整 C_p使其偏离最优值即可。根据风电机组工作原理，C_p的取值与 λ 和 β 相关，且相同的 C_p值可以对应于不同的 λ 和 β 组合。

以常见的超速+变桨协调控制为例，当风电机组需要降功率时，该策略首先通过使转子转速超速来降低 C_p。当转子转速增加到上限时，则通过增加桨距角进一步降低 C_p。传统 AGC 策略示意图如图 3 所示，所述过程如图 3 中的箭头所示。

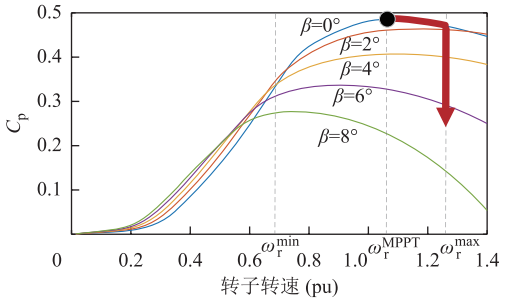

图 3 传统 AGC 策略示意图

Fig. 3 Diagram of traditional AGC strategy

研究人员认为，主轴转矩 T_s和塔根弯矩 M_t的波动可用于计算风电机组的疲劳载荷，降低风电机组运行过程中 T_s和 M_t的标准差可降低机组疲劳。根据式（6）和式（8）可知，T_s、M_t与转子转速、桨距角的波动有直接关系。在如图 3 所示的变速变桨控制中，桨距角的变化是在转子转速之后的，受到转子转速设定值的控制，因此在实现有功功率控制时不能使两个参数同时自由调整，即难以进一步考虑对风电机组疲劳载荷的优化。

基于此，本文提出了一种 AGC 优化控制框架。在以上描述的有功功率优化控制框架下，可以将式（11）~式（26）所描述的线性化模型写成状态空间方程，即

$$\begin{cases}\dot{\boldsymbol{x}}=\boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{u}+\boldsymbol{E}\boldsymbol{d}\\ \boldsymbol{y}=\boldsymbol{C}\boldsymbol{x}+\boldsymbol{D}\boldsymbol{u}+\boldsymbol{W}\boldsymbol{d}\end{cases} \tag{27}$$

其中

$$\boldsymbol{A}=\begin{pmatrix}\dfrac{1}{J_t}\dfrac{\partial T_m}{\partial\omega_r} & -\dfrac{N_{gear}}{J_t} & \dfrac{1}{J_t}\dfrac{\partial T_m}{\partial\beta}\\ 0 & -\dfrac{1}{\tau_e} & 0\\ 0 & 0 & -\dfrac{1}{\tau_\beta}\end{pmatrix}\quad \boldsymbol{B}=\begin{pmatrix}0 & 0\\ \dfrac{1}{\tau_e} & 0\\ 0 & \dfrac{1}{\tau_\beta}\end{pmatrix}$$

$$\boldsymbol{C}=\begin{pmatrix}\dfrac{\partial P_e}{\partial\omega_r} & \dfrac{\partial P_e}{\partial T_e} & 0\\ \dfrac{\partial M_t}{\partial\omega_r} & 0 & \dfrac{\partial M_t}{\partial\beta}\\ \dfrac{N_{gear}^2J_e}{J_t}\dfrac{\partial T_m}{\partial\omega_r} & \dfrac{N_{gear}J_m}{J_t} & \dfrac{N_{gear}^2J_e}{J_t}\dfrac{\partial T_m}{\partial\beta}\end{pmatrix}$$

$$\boldsymbol{D}=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}^T\quad \boldsymbol{E}=\begin{bmatrix}\dfrac{1}{J_t}\dfrac{\partial T_m}{\partial v} & 0 & 0\end{bmatrix}^T$$

$$\boldsymbol{W}=\begin{bmatrix}0 & \dfrac{\partial M_{\mathrm{t}}}{\partial v} & \dfrac{N_{\mathrm{gear}}^2 J_{\mathrm{e}}}{J_{\mathrm{t}}}\dfrac{\partial T_{\mathrm{m}}}{\partial v}\end{bmatrix}^{\mathrm{T}} \quad d=\Delta v$$

$$\boldsymbol{u}=\begin{bmatrix}\Delta T_{\mathrm{e}}^{\mathrm{ref}} & \Delta\beta^{\mathrm{ref}}\end{bmatrix}^{\mathrm{T}} \quad \boldsymbol{x}=\begin{bmatrix}\Delta\omega_{\mathrm{r}} & \Delta T_{\mathrm{e}} & \Delta\beta\end{bmatrix}^{\mathrm{T}}$$

$$\boldsymbol{y}=\begin{bmatrix}\Delta P_{\mathrm{e}} & \Delta M_{\mathrm{t}} & \Delta T_{\mathrm{s}}\end{bmatrix}^{\mathrm{T}}$$

在式（27）所示模型中，风电机组有功控制系统的控制输入为电磁转矩和桨距角的变化量，输出为有功功率、塔弯矩和主轴转矩的变化量。基于上述模型，通过设计优化控制器，可以让控制器同时生成电磁转矩和桨距角的参考值，实现调度指令功率跟踪，减少风电机组运行疲劳载荷。

3　风电场分布式 AGC 控制框架与求解算法

3.1　风电场分布式控制框架

风电场的 AGC 信号通常由风电场控制中心计算并分配给场内每台风电机组。在这样的分配过程中，难以根据风电机组的实际运行状态和风况实施闭环控制。本文提出一种基于 ADMM 算法的 DMPC 框架进行风电机组参考功率的分配，由风电机组的本地控制器和风电场的控制中心协调实现，如图 4 所示。该框架可以在完成风电场水平所需的功率的前提下，通过对每台风电机组的优化控制进一步降低风电机组的疲劳载荷。

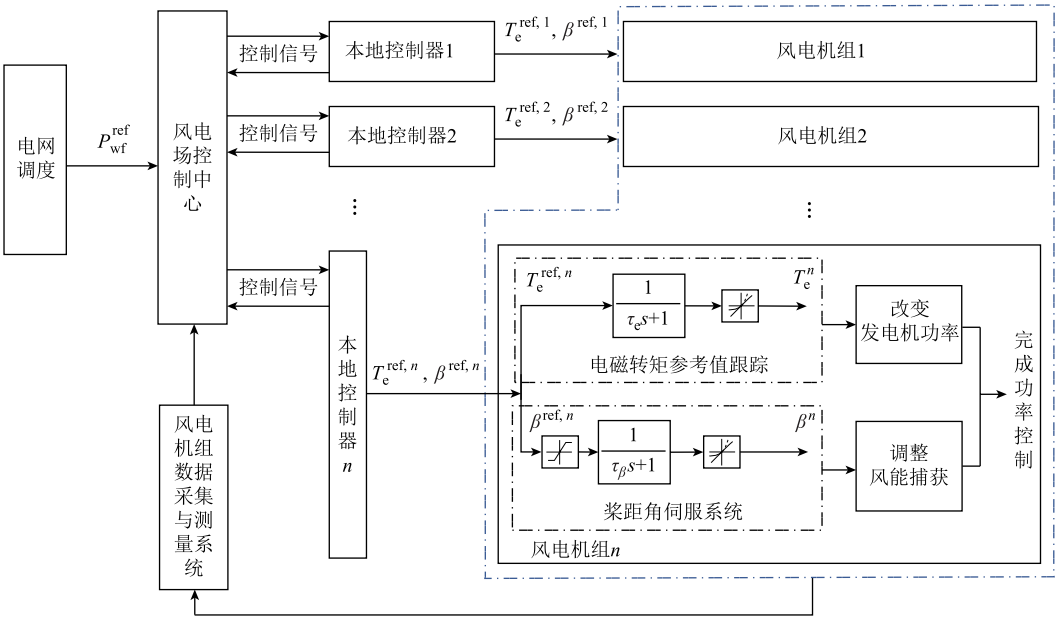

图 4　风电场分布式控制框架

Fig. 4　Distributed wind farm control framework

在如图 4 所示的风电场分布式 AGC 控制框架中，风电场控制中心在获得调度中心下达的 AGC 控制指令后，与风电机组的本地控制器进行双向的信息通信，完成 3.3 节中算法所需的参数迭代计算。通过中心控制器和本地控制器的协调计算，每台风电机组均可获得本机的电磁转矩与桨距角设定值，通过本机执行机构中完成有功功率的跟踪。同时，每台风电机组均通过实时测量元件获得包括机舱风速、转子转速以及桨距角等机组运行参数，反馈给机组的本地控制器，从而形成闭环控制。

值得注意的是，尾流效应和湍流强度作为影响风电场运行特性的重要因素，在风电场优化运行过程中必须考虑。更进一步地，在风电场 AGC 的过程中还需要考虑由于功率调整而产生的动态尾流。本文所建风电场模型在仿真中借助丹麦奥尔堡大学开发的 Simwindfarm 工具箱，在风电场优化中均考虑了基于 Larsen 模型的尾流计算和基于 IEC 61400-3 所规定的湍流模型，此处不再对相关模型进行详细描述[28]。

在以上所述尾流及湍流模型的基础上，对于仿真中的风电机组本地控制器，与其风速相关的状态反馈来自所测量的本机机舱风速。

3.2　风电场 AGC 功率优化控制目标

采用第 2 节中所提风电机组 AGC 功率控制模型，风电机组的本地控制器可以计算单机层面的优化控制方案。进一步地，通过与风电场控制中心进行的信息共享和迭代，确保全场 AGC 功率总和满足全局约束。根据以上定义，可将风电场有功功率的最佳控制的全局目标函数设置为式（28）和式（29），即

$$\min \sum_{j=1}^{N_t}\sum_{i=1}^{N_c}\left\{\left\|\Delta P_{e,j}(k+i|k)-\Delta P_{e,j}^{ref}\right\|_{\boldsymbol{Q}_P}^2+\left\|\Delta M_{t,i}(k+i|k)\right\|_{\boldsymbol{Q}_M}^2+\left\|\Delta T_{s,i}(k+i|k)\right\|_{\boldsymbol{Q}_T}^2\right\} \tag{28}$$

$$\text{s.t.}\begin{cases}\sum_{i=1}^{N_c}(\Delta P_{wf}^{ref}(k+i|k)-\sum_{j=1}^{N_t}\Delta P_{e,j}(k+i|k))=0\\ \Delta P_{e,i}^{min}\leqslant\Delta P_{e,i}\leqslant\Delta P_{e,i}^{max}\\ 0\leqslant\left|\Delta M_{t,i}\right|\leqslant\Delta M_{t,i}^{max}\\ 0\leqslant\left|\Delta T_{s,i}\right|\leqslant\Delta T_{s,i}^{max}\end{cases} \tag{29}$$

式中，N_t为风电场内风电机组数量；N_c为控制时域；k为当前时刻；$\boldsymbol{Q}_P$、$\boldsymbol{Q}_M$、$\boldsymbol{Q}_T$为各输出值的惩罚矩阵；ΔP_{wf}^{ref}为风电场有功功率参考值的变化量。

3.3 风电场分布式 AGC 求解算法

ADMM 是乘子法的扩展，它允许采用分布式公式，在对偶公式中使用增广拉格朗日函数，是一种收敛性较好的 DMPC 求解方法。针对 3.2 节中所描述的优化问题，采用 ADMM 进行求解的具体方法如下。

使用 Hessian 矩阵 $\boldsymbol{H}_i\in\mathbf{R}^{2N_c\times 2N_c}$ 和系数向量 $\boldsymbol{f}_i\in\mathbf{R}^{2N_c\times 1}$，将式（28）、式（29）中的问题改写为标准的二次规划问题。式（28）中的问题可分解为 N_t 个局部问题，其中第 i 个局部问题可写为

$$\min\ (\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{H}_i\boldsymbol{U}_i+\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{U}_i)+\gamma\left\|z_i\right\|_2^2 \tag{30}$$

$$\begin{cases}\boldsymbol{P}\boldsymbol{Y}_i=\Delta P_i^{ref}-z_i\\ A_{c,i}\boldsymbol{U}_i\leqslant b_i\end{cases} \tag{31}$$

式中，$A_{c,i}$和 b_i为将式（29）中的后三个约束转换成标准二次规划形式时获得的约束矩阵。同时，

$$\begin{cases}\boldsymbol{Y}_i=\left[y_i(k+1|k)\ \ y_i(k+2|k)\ \ \cdots\ \ y_i(k+N_c|k)\right]^{\mathrm{T}}\\ \boldsymbol{U}_i=\left[u_i(k|k)\ \ y_i(k+1|k)\ \ \cdots\ \ y_i(k+N_c-1|k)\right]^{\mathrm{T}}\end{cases} \tag{32}$$

$$\boldsymbol{P}=[1\ \ 0\ \ 0\ \ 1\ \ 0\ \ 0\ \ \cdots\ \ 1\ \ 0\ \ 0]_{1\times 3N_c} \tag{33}$$

$$\begin{aligned}\boldsymbol{Y}_i=&\begin{bmatrix}\boldsymbol{C}_d\boldsymbol{B}_d & 0 & 0 & 0\\ \boldsymbol{C}_d\boldsymbol{A}_d\boldsymbol{B}_d & \boldsymbol{C}_d\boldsymbol{B}_d & 0 & 0\\ \vdots & \vdots & & 0\\ \boldsymbol{C}_d\boldsymbol{A}_d^{N_c-1}\boldsymbol{B}_d & \boldsymbol{C}\boldsymbol{A}_d^{N_c-2}\boldsymbol{B}_d & \cdots & \boldsymbol{C}_d\boldsymbol{B}_d\end{bmatrix}\boldsymbol{U}_i+\\ &\begin{bmatrix}\boldsymbol{C}_d\boldsymbol{A}_d\\ \boldsymbol{C}_d\boldsymbol{A}_d^2\\ \vdots\\ \boldsymbol{C}_d\boldsymbol{A}_d^{N_c}\end{bmatrix}\boldsymbol{x}_i(k|k)+\begin{bmatrix}\boldsymbol{C}_d\boldsymbol{E}_d\\ \boldsymbol{C}_d\boldsymbol{E}_d^2\\ \vdots\\ \boldsymbol{C}_d\boldsymbol{E}_d^{N_c}\end{bmatrix}\boldsymbol{d}_i(k|k)\\ =&\boldsymbol{\mathcal{M}}\boldsymbol{x}_i+\boldsymbol{\mathcal{Q}}\boldsymbol{U}_i+\boldsymbol{\mathcal{V}}\boldsymbol{d}_i\end{aligned} \tag{34}$$

进一步地，可以得到 ADMM 算法所需增广拉格朗日函数，即

$$\begin{aligned}\mathcal{L}_i=&(\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{H}_i\boldsymbol{u}_i+\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{U}_i)+\gamma\left\|z_i\right\|_2^2+\\ &\lambda_i^{\mathrm{T}}(A_{c,i}\boldsymbol{U}_i-b_i+t_i)+\frac{\rho}{2}\left\|A_{c,i}\boldsymbol{U}_i-b_i+t_i\right\|_2^2+\\ &\mu_i^{\mathrm{T}}(\boldsymbol{P}\boldsymbol{\mathcal{M}}\boldsymbol{x}_i+\boldsymbol{P}\boldsymbol{\mathcal{Q}}\boldsymbol{U}_i+\boldsymbol{P}\boldsymbol{\mathcal{V}}\boldsymbol{d}_i-\Delta P_i^{ref}+z_i)+\\ &\frac{\eta}{2}\left\|\boldsymbol{P}\boldsymbol{\mathcal{M}}\boldsymbol{x}_i+\boldsymbol{P}\boldsymbol{\mathcal{Q}}\boldsymbol{U}_i+\boldsymbol{P}\boldsymbol{\mathcal{V}}\boldsymbol{d}_i-\Delta P_i^{ref}+z_i\right\|_2^2\end{aligned} \tag{35}$$

由于 $\boldsymbol{x}_i$ 和 $\boldsymbol{d}_i$ 是一个控制周期内的可测量参数，为简化表达式，令

$$\Theta(\boldsymbol{U}_i)=\boldsymbol{P}\boldsymbol{\mathcal{M}}\boldsymbol{x}_i+\boldsymbol{P}\boldsymbol{\mathcal{Q}}\boldsymbol{U}_i+\boldsymbol{P}\boldsymbol{\mathcal{V}}\boldsymbol{d}_i \tag{36}$$

因此，可以推导出式（30）的求解公式为

$$\begin{aligned}\boldsymbol{U}_i^{k+1}=\arg\min_{\boldsymbol{U}_i}\Big(&\boldsymbol{U}_i^{\mathrm{T}}\boldsymbol{H}_i\boldsymbol{U}_i+\boldsymbol{f}_i^{\mathrm{T}}\boldsymbol{U}_i+\\ &\frac{\rho}{2}\left\|A_{c,i}\boldsymbol{U}_i-b_i+t_i^k+\frac{\lambda_i^k}{\rho_i}\right\|_2^2+\\ &\frac{\eta}{2}\left\|\Theta(\boldsymbol{U}_i)-\Delta P_i^{ref}+z_i^k+\frac{\mu_i^k}{\eta_i}\right\|_2^2\Big)\end{aligned} \tag{37}$$

$$z_i^{k+1}=\arg\min_{z_i}\left(\gamma\left\|z_i\right\|_2^2+\frac{\eta}{2}\left\|\Theta(\boldsymbol{U}_i^{k+1})-\Delta P_i^{ref}+z_i+\frac{\mu_i}{\eta_i}\right\|_2^2\right) \tag{38}$$

$$t_i^{k+1}=\max\left\{0,A_{c,i}\boldsymbol{U}_i-b_i+\frac{\lambda_i}{\rho}\right\} \tag{39}$$

$$\lambda_i^{k+1}=\lambda_i^k+\rho(A_{c,i}\boldsymbol{U}_i-b_i+t_i^{k+1}) \tag{40}$$

$$\mu_i^{k+1}=\mu_i^k+\eta(\Theta(\boldsymbol{U}_i^{k+1})-\Delta P_i^{ref}+z_i^{k+1}) \tag{41}$$

在如 3.1 节中所描述的风电场 DMPC 框架下，风电场控制中心将计算出的风电机组参考功率 ΔP_i^{ref} 传递给机组，风电机组根据式（37）~式（41）在本地控制器上迭代计算得到优化的机组控制指令 U_i，最终实现控制目标。

4 算例分析

4.1 仿真系统搭建

本文通过 Matlab/Simulink 搭建仿真系统以验证本文所建风电机组多入多出控制模型和风电场分布式控制算法。用于做验证的风电机组为 NREL 5MW 机组，主要参数如文献[29]所介绍。在单机模型的基础上，本文搭建了一个 3×3 排列的风电场模型，

风机排列情况如图 5 所示，场内风电机组之间的间距为 400m。仿真过程中风电场的来流风速平均值为 14m/s，湍流强度为 0.1，场内尾流的计算基于 Sim Wind Farm[28]，如图 6 所示，场内 9 台机组的风速波动范围约为 9~18m/s，覆盖了机组多种运行工况。

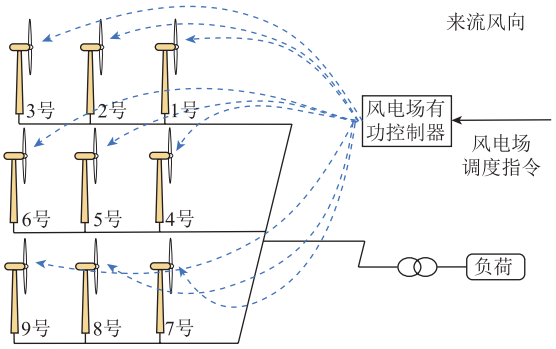

图 5　机组排布与尾流效果图

Fig.5　Unit layout and wake effect diagram

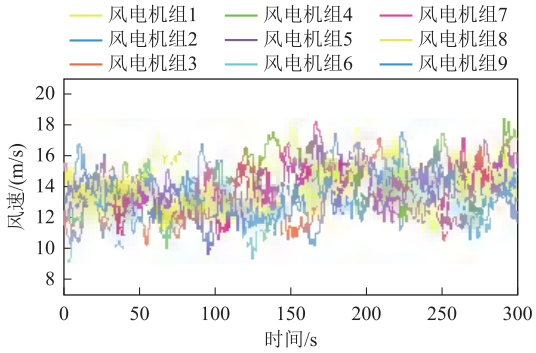

图 6　风电场 9 台机组仿真风速

Fig.6　Simulation wind speed of 9 units in wind farm

在仿真过程中，风电机组层面的控制方案由传统的超速+变桨控制[30]与本文提出的多入多出控制模型进行对比；风电场层面的功率分配策略采用传统的以最大发电能力计算的等比例分配策略[31]和本文提出的灵活分配策略进行对比；功率控制器则由传统的 PI 控制器、MPC 控制器[9]与本文提出的 DMPC 控制器相对比。仿真中策略的组合情况见表 1。

表 1　仿真中策略的组合情况

Tab.1　Combination of strategies in simulation

策略	风电机组	风电场功率分配	控制器方案
1	超速+变桨控制	等比例分配	PI 控制
2	多入多出控制	等比例分配	MPC 控制[9]
3	多入多出控制	灵活分配	DMPC 控制

4.2　风电场 AGC 控制结果分析

以丹麦风电并网标准为例，风电场需要具备限制有功出力和功率爬坡的能力[32]。本算例中，风电场处于受调度指令控制的状态，初始功率指令为 25MW，60s 时指令开始爬坡，180s 时指令增加至 30MW 并一直保持恒定至 300s。基于表 1 所示策略的风电场有功出力结果如图 7 所示。

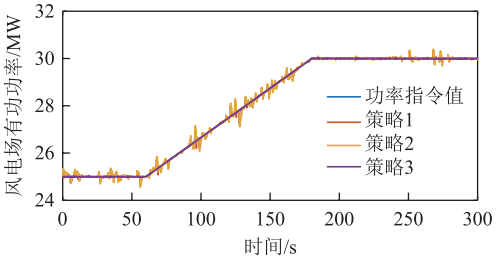

图 7　不同策略下风电场出力对比

Fig.7　Comparison of output under different strategies

由图 7 可知三种策略均能有效实现风电场 AGC 功率指令值。其中，策略 1 和策略 2 的出力结果较指令值出现了明显的毛刺，这是由于采用等比例分配时需通过风电机组风速的测量来计算机组的最大出力，再按比例进行功率分配。这一过程中存在风速测量误差，从而导致最终的风电场输出功率出现了与指令值的偏差。虽然三种策略都能够较好地完成调度对风电场的 AGC 功率要求，但相比之下，采用功率灵活分配框架的策略 3 则更好地完成了调度指令。

进一步地，根据 MCrunch 算法计算得到场内风电机组在三种策略下的疲劳载荷情况见表 2。

表 2　不同策略下风电机组的损伤等效载荷

Tab.2　DELs of wind turbines under different strategies

（单位：MN·m）

参数		WT1	WT2	WT3	WT4	WT5	WT6	WT7	WT8	WT9	平均值
策略 1	DEL-T_s	0.788	0.894	1.087	1.096	1.271	1.517	0.807	0.819	0.929	1.023
	DEL-M_t	29.55	29.79	55.99	28.56	27.18	75.62	29.20	30.73	33.17	37.75
策略 2	DEL-T_s	0.698	0.863	1.034	0.809	0.973	1.165	0.748	0.778	0.887	0.884
	DEL-M_t	18.18	17.40	20.37	17.18	17.63	22.25	17.30	21.63	23.91	19.54
策略 3	DEL-T_s	0.709	0.672	0.663	0.679	0.674	0.711	0.716	0.707	0.661	0.688
	DEL-M_t	18.22	16.98	18.63	16.92	17.57	22.01	16.63	18.52	21.09	18.50

表 2 中，DEL-T_s 和 DEL-M_t 分别为传动系统和塔架对应的 DEL 值。由计算结果可以看出，对比策略 1，由于策略 2 采用了本文提出的多入多出控制模型，对比传统的机组控制方案，疲劳载荷得到了降低，全场机组传动系统和塔架疲劳的平均值分别下降了 12.3%和 38.5%。然而策略 2 对场级功率分配没有进行优化，本文所提的策略 3 通过进一步的优化分配，使得机组的疲劳载荷得到了更进一步的降低，全场机组传统系统和塔架疲劳的平均值相对于策略 2 再次降低了 22.2%和 5.3%。

由此可见，在风电场层面，本文所建机组控制模型和分布式控制框架可以在完成调度要求的功率指令的前提下，有效降低机组的疲劳程度。

4.3 风电机组控制结果分析

4.2 节从风电场层面讨论了本文所建模型和所提控制策略的效果，可以看到风电场内所有机组的疲劳载荷较采用传统的策略 1 均有了明显降低，但各机组降低的程度有所不同。本节以采用策略 1 时疲劳载荷最高的 6 号风机（WT6）为例，分析各策略下风电机组层面的状态变化情况。

WT6 的机舱测量风速如图 8a 所示，由于湍流的存在，风速在 9~17m/s 间波动。采用等比例分配时，低于额定风速会被分配比例较低的功率指令，因此有功功率如图 8b 所示，策略 1 和策略 2 在图 8a 中风速低于 12m/s 时功率都出现了明显波动。而策略 1 和策略 2 都是工作于等比例分配的框架下，单机接收到功率指令值相同，因此出力也基本相同，这也进一步证实了本文所提控制模型的有效性。对于本文所提的策略 3，由于采用了分布式的控制框架，在图 8b 中，功率指令更为平滑。

在图 8c 和图 8d 中，采用策略 2 和策略 3 时可以同时调整转速设定值和桨距角设定值，发电机转速不需要一直限制在额定值附近波动，桨距角波动更为平缓。如图 8e 和图 8f 所示，这样的状态变化导致机组的主轴转矩和塔根弯矩的波动在采用策略 2 和策略 3 时都更加平缓，尤其是采用策略 3 时，这与表 2 中的疲劳载荷结果相吻合。

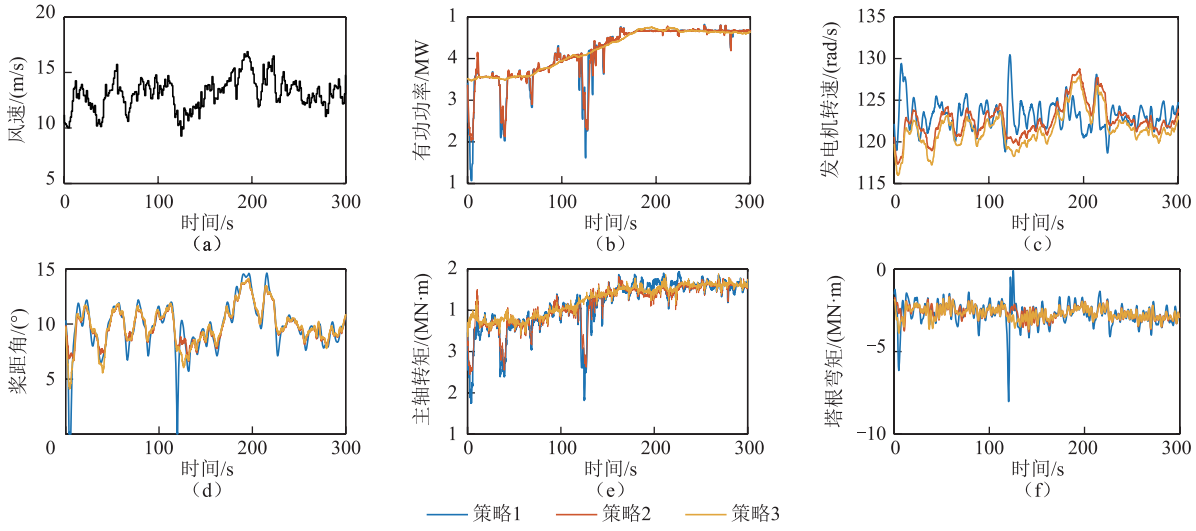

图 8 WT6 机组在仿真过程中的状态变化

Fig.8 State change of WT6 during simulation

5 结论

本文以风电场 AGC 控制过程为研究对象，在跟踪调度指令的前提下，考虑在控制过程中的风电机组疲劳载荷优化问题，并就此提出了一种风电机组控制模型和分布式的风电场 AGC 框架，得出以下结论：

1）本文通过对风电机组非线性模型的线性化，建立了一种考虑有功功率输出和疲劳载荷指标的多入多出线性模型，通过单一控制器可以完成对转子转速和桨距角的同时调整，较传统控制方法更加灵活，可明显降低机组运行的疲劳载荷。

2）本文建立了一种基于 ADMM 的风电场分布式 AGC 控制框架，实现对多入多出风电机组模型的有效控制，可以有效得完成调度设定的有功功率参考值，并且通过优化计算进一步降低机组运行的疲劳载荷。

参考文献

[1] 穆钢，蔡婷婷，严干贵，等. 双馈风电机组参与持

续调频的双向功率约束及其影响[J]. 电工技术学报, 2019, 34(8): 1750-1759.
Mu Gang, Cai Tingting, Yan Gangui, et al. Bidirectional power constraints and influence of doubly fed induction generator participating in continuous frequency regulation[J]. Transaction of China Electrotechnical Society, 2019, 34(8): 1750-1759.

[2] Guo Xiaoqiang, Chen Weijian. Control of multiple power inverters for more electronics power systems: a review[J]. CES Transactions on Electrical Machines and Systems, 2018, 2(3): 255-263.

[3] 王涛, 诸自强, 年珩. 非理想电网下双馈风力发电系统运行技术综述[J]. 电工技术学报, 2020, 35(3): 455-471.
Wang Tao, Zhu Ziqiang, Nian Hang. Review of operation technology of doubly-fed induction generator-based wind power system under nonideal grid conditions[J]. Transactions of China Electrotechnical Society, 2021, 36(3): 579-587.2020, 35(3): 455-471.

[4] 麻秀范, 王戈, 朱思嘉, 等. 计及风电消纳与发电集团利益的日前协调优化调度[J]. 电工技术学报, 2021, 36(3): 579-587.
Ma Xiufan, Wang Ge, Zhu Sijia, et al. Coordinated day-ahead optimal dispatch considering wind power consumption and the benefits of power generation group[J]. Transactions of China Electrotechnical Society, 2021, 36(3): 579-587.

[5] 杨正清, 汪震, 展肖娜, 等. 考虑风电有功主动控制的两阶段系统备用双层优化模型[J]. 电力系统自动化, 2016, 40(10): 31-37.
Yang Zhengqing, Wang Zhen, Zhan Xiaona, et al. Bi-level optimization model of two-stage reserve scheduling with proactive wind power control[J]. Automation of Electric Power Systems, 2016, 40(10): 31-37.

[6] Guo Yifei, Gao Houlei, Wu Qiuwei, et al. Distributed coordinated active and reactive power control of wind farms based on model predictive control[J]. International Journal of Electrical Power & Energy Systems, 2019, 104: 78-88.

[7] Hu Jianqiang, Chen M Z Q, Cao Jinde, et al. Coordinated active power dispatch for a microgrid via distributed lambda iteration[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2017, 7(2): 250-261.

[8] 苏永新, 段斌, 朱广辉, 等. 海上风电场疲劳分布与有功功率统一控制[J]. 电工技术学报, 2015, 30(22): 190-198.
Su Yongxin, Duan Bin, Zhu Guanghui, et al. Fatigue distribution and active power combined control in offshore wind farm[J]. Transaction of China Electrotechnical Society, 2015, 30(22): 190-198.

[9] Riverso S, Mancini S, Sarzo F, et al. Model predictive controllers for reduction of mechanical fatigue in wind farms[J]. IEEE Transactions on Control Systems Technology, 2017, 25(2): 535-549.

[10] Zhao Haoran, Wu Qiuwei, Huang Shaojun, et al. Fatigue load sensitivity-based optimal active power dispatch for wind farms[J]. IEEE Transactions on Sustainable Energy, 2017, 8(3): 1247-1259.

[11] Knudsen T, Bak T, Svenstrup M. Survey of wind farm control-power and fatigue optimization[J]. Wind Energy, 2015, 18(8): 1333-1351.

[12] Zhang Baohua, Soltani M, Hu Weihao, et al. Optimized power dispatch in wind farms for power maximizing considering fatigue loads[J]. IEEE Transactions on Sustainable Energy, 2018, 9(2): 862-871.

[13] 刘吉臻, 孟洪民, 胡阳. 采用梯度估计的风力发电系统最优转矩最大功率点追踪效率优化[J]. 中国电机工程学报, 2015, 35(10): 2367-2374.
Liu Jizhen, Meng Hongmin, Hu Yang. Efficiency optimization of optimum torque maximum power point tracking based on gradient approximation for wind turbine generator system[J]. Proceedings of the CSEE, 2015, 35(10): 2367-2374.

[14] 姚琦, 刘吉臻, 胡阳, 等. 含异步变速风机的风电场一次调频等值建模与仿真[J]. 电力系统自动化, 2019, 43(23): 185-192.
Yao Qi, Liu Jizhen, Hu Yang, et al. Equivalent modeling and simulation for primary frequency regulation of wind farm with asynchronous variable-speed wind turbines[J]. Automation of Electric Power Systems, 2019, 43(23): 185-192.

[15] Wang Shu, Seiler P. LPV Active power control and robust analysis for wind turbines[C]//33rd Wind Energy Symposium, Kissimmee, Florida: American

Institute of Aeronautics and Astronautics, 2015.

[16] 刘军, 张彬彬, 赵婷. 基于模糊评价的风电场有功功率分配算法[J]. 电工技术学报, 2019, 34(4): 786-794.
Liu Jun, Zhang Binbin, Zhao Ting. Research on wind farm active power dispatching algorithm based on fuzzy evaluation[J]. Transaction of China Electrotechnical Society, 2019, 34(4): 786-794.

[17] Zhao Haoran, Wu Qiuwei, Guo Qinglai, et al. Distributed model predictive control of a wind farm for optimal active power control part I: clustering-based wind turbine model linearization[J]. IEEE Transactions on Sustainable Energy, 2015, 6(3): 831-839.

[18] Luo Haocheng, Hu Zechun, Zhang Hongcai, et al. Coordinated active power control strategy for deloaded wind turbines to improve regulation performance in AGC[J]. IEEE Transactions on Power Systems, 2019, 34(1): 98-108.

[19] Tang Xuesong, Yin Minghui, Shen Chun, et al. Active power control of wind turbine generators via coordinated rotor speed and pitch angle regulation[J]. IEEE Transactions on Sustainable Energy, 2019, 10(2): 822-832.

[20] Hoek D van der, Kanev S, Engels W. Comparison of down-regulation strategies for wind farm control and their effects on fatigue loads[C]//2018 Annual American Control Conference (ACC), 2018: 3116-3121.

[21] Spudic V. Coordinated optimal control of wind farm active power[D]. Croatia: University of Zagreb, 2012.

[22] Jin Jingliang, Zhou Dequn, Zhou Peng, et al. Dispatching strategies for coordinating environmental awareness and risk perception in wind power integrated system[J]. Energy, 2016, 106: 453-463.

[23] Zhang Jinhua, Liu Yongqian, Infield D, et al. Optimal power dispatch within wind farm based on two approaches to wind turbine classification[J]. Renewable Energy, 2017, 102: 487-501.

[24] Wang Lei, Wen Jie, Cai Ming, et al. Distributed Optimization control schemes applied on offshore wind farm active power regulation[J]. Energy Procedia, 2017, 105: 1192-1198.

[25] Yao Qi, Hu Yang, Chen Zhe, et al. Active power dispatch strategy of the wind farm based on improved multi-agent consistency algorithm[J]. IET Renewable Power Generation, 2019, 13(14): 2693-2704.

[26] Zhao Haoran, Wu Qiuwei, Guo Qinglai, et al. Distributed model predictive control of a wind farm for optimal active power control part II: implementation with clustering-based piece-wise affine wind turbine model[J]. IEEE Transactions on Sustainable Energy, 2015, 6(3): 840-849.

[27] 李世春, 黄悦华, 王凌云, 等. 基于转速控制的双馈风电机组一次调频辅助控制系统建模[J]. 中国电机工程学报, 2017, 37(24): 7077-7086, 7422.
Li Shichun, Huang Yuehua, Wang Lingyun, et al. Modeling primary frequency regulation auxiliary control system of doubly fed induction generator based on rotor speed control[J]. Proceedings of the CSEE, 2017, 37(24): 7077-7086, 7422.

[28] Aalborg University. SimWindFarm Toolbox[EB/OL]. /2020-03-07. http://www.ict-aeolus.eu/SimWindFarm/index.html.

[29] Jonkman J, Butterfield S, Musial W, et al. Definition of a 5-MW reference wind turbine for offshore system development[R]. NREL/TP-500-38060, 947422, 2009.

[30] Yao Qi, Liu Jizhen, Hu Yang. Optimized active power dispatching strategy considering fatigue load of wind turbines during de-loading operation[J]. IEEE Access, 2019, 7: 17439-17449.

[31] Merahi F, Berkouk E, Mekhilef S. New management structure of active and reactive power of a large wind farm based on multilevel converter[J]. Renewable Energy, 2014, 68: 814-828.

[32] Regulations for grid connection | Energinet[EB/OL]. 2019-10-24. https://en.energinet.dk/Electricity/Rules-and-Regulations/Regulations-for-grid-connection.

作者简介

姚 琦 男，1994 年生，博士，讲师，研究方向为风力发电系统及其控制。

E-mail：qiyao@jnu.edu.cn

綦 晓 男，1992 年生，博士，讲师，研究方向为综合能源系统控制。

E-mail：qixiao@jnu.edu.cn （通信作者）

（编辑 赫蕾）

CL-Windcon

Closed Loop Wind Farm Control

DELIVERABLE REPORT

D1.2: Description of the reference and the control-oriented wind farm models

Deliverable No.	D1.2	Work Package No.	WP1	Task/s No.	Task 1.2
Work Package Title		Wind farm control-oriented model development			
Linked Task/s Title		Multi-fidelity modeling			
Status		**Final**	(Draft/Draft Final/Final)		
Dissemination level		**PU-Public**	(PU-Public, PP, RE-Restricted, CO-Confidential) (https://www.iprhelpdesk.eu/kb/522-which-are-different-levels-confidentiality)		
Due date deliverable		2018-04-30	Submission date		2018-04-30
Deliverable version		1.0 (Final, latest update: April 30th)			

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 727477

7 SIMWINDFARM

7.1 Model description

7.1.1 Description summary

The SimWindFarm (SWF) toolbox is initially developed as part of the [http://www.ict-aeolus.eu Aeolus FP7] project and is aimed at providing a fast wind farm simulation environment for development of wind farm control algorithms.

7.1.2 Model purpose

There has been made a lot of research on wind farm control strategies based on static models Brand and Wagenaar (2010). To increase the confidence in the results also dynamic wind farm simulation tools has been developed. The state of the art in this category is SOWFA Churchfield and Lee (2013); Fleming et al. (2013). The high fidelity of SOWFA comes with the cost of long simulation time. The main idea with SimWindFarm is to fit in between these two extremes. The main purpose was to be used as a simulator for testing wind farm controllers.

7.1.3 Model specifications

SWF is designed as a Simulink toolbox, which can generate wind farm models based on the users choice of layout and turbine. The Simulink model generated has the structure shown in Figure 7.1. The two main elements are the turbines block and the wind field block. The turbines block simulates the dynamics of the wind turbines based on wind speed and power set point. Each of the turbines produce a standard set of outputs to be used during the design of the farm controller along with the thrust coefficient of the individual turbines, used to calculate the wake effects. Included are also an example farm controller, network controller and grid models, which can be replaced with whatever models or algorithms the control designer wants. The example farm controller is a simple proportional distribution algorithm, which distributes the power reference to the turbines based on a simple estimate of its current production capacity and the total power demand from the network operator.

The SimWindFarm toolbox supports the following features:

- Simulation of wind farms with arbitrary layout
- Automatic generation of wind farm models

- Integration into Simulink
- Simple farm control and network operation
- Fatigue analysis

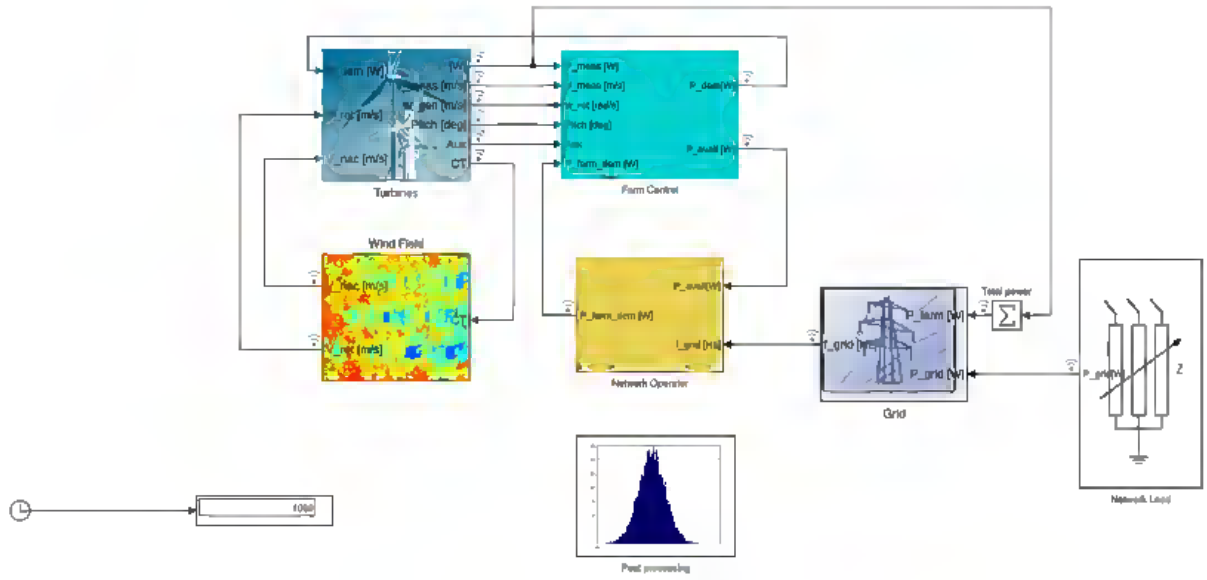

Figure 7.1: Top level structure of SWF as illustrated by the Simulink diagram.

The following turbine models are available:

- NREL5MW This is the well known virtual turbine from NREL Jonkman et al. (2009).
- DTU10MW This is the newer virtual turbine from the UpWind project Bak et al. (2013).
- DTU10MWCLWC This is the DT10MW with the CLWindcon controller (OpenDiscon).

7.1.4 Detailed mathematical derivation

The detailed mathematical derivation is given in Appendix A.

7.2 Installation and usage

7.2.1 System requirements

The toolbox requires Matlab/Simulink. The current release (v1.2) is tested with matlab version R2017b on Ubuntu LTS16.04. SimWindFarm should run on the platforms that supports

Matlab/Simulink. The earlier versions of SimWindFarm has been tested with earlier versions of Matlab back to at least 2010a.

7.2.2 Installation instructions

The toolbox is available via GitHub (TBD). Previous versions, up to and including V1.1, are available in zipped format by the link http://www.ict-aeolus.eu/SimWindFarm/simwindfarm-v1.1.zip.

For easiest use of the toolbox write permissions to the Matlab path-file are needed. Else to redo the install every time Matlab is started or to set the necessary paths is needed.

- Download (and unzip) the software to a suitable location
- Run `install.m`
- Alternatively to running `install.m` the necessary path can be set by something like `addpath(genpath('Directory with SimWindFarm files'),'-end');`
- Open the Simulink Library browser and there should be a new SimWindFarm item
- Note that an installed Matlab mex compiler (run mex -setup) is needed.

The wind data can be generated using the m functions gen_windfield*.m. A precalculated wind field can also be downloaded at http://www.ict-aeolus.eu/SimWindFarm/benchmarkwind_no_taylor.mat. Remember to check if it fits the application.

7.2.3 Folder hierarchy

The directory structure is very flat with most files at the root. Only 4 sub directories are used: `DTU10MW`, `nrel5mw`, `mcrunch` and `private`. The first two holds information on the DTU10MW and NREL5MW turbines and the last two has files on fatigue etc.

7.2.4 Setting up a simulation case

In earlier versions of SimWindFarm it was possible to construct wind farm Simulink models solely by using a dialog started by double clicking e.g. `Farm Template`. This seems not to work presently with ubuntu 16.04 and Matlab 2017b. Alternatively, a programmatic approach can be followed as explained below.

Which files to change and how

Wind is generated by running `GenWind.m`. In this file the farm geometry, turbine name, mean wind speed, turbulence intensity, size of wind field and simulation time are specified. The file, `GenWind.m`, and the next ones mentioned here, are supposed to be self explanatory.

The farm is generated by running `GenWindFarm.m`. In this file the turbine name and wind data are specified.

The wind turbine parameters used by the turbine block in Simulink are stored in various matlab structs `env`, `pub` and `wt`. Examples of setting these parameters are found in `./nrel5mw/nrelvals.m` and `./DTU10MW/DTU10MWPar.m` for the NREL5MW and DTU10MW turbine respectively.

- env only holds the air density in `env.rho`.
- pub holds rotor radius, rated power and fatigue parameters.

```
pub =
  struct with fields:
    rotor: [1x1 struct]
    rated: 5000000
    tower: [1x1 struct]
    shaft: [1x1 struct]
```

- `wt` holds everything else needed in the dynamic model including the controller.

```
wt =
  struct with fields:
       cp: [1x1 struct]
       ct: [1x1 struct]
    blade: [1x1 struct]
      hub: [1x1 struct]
      nac: [1x1 struct]
```

```
tower: [1x1 struct]
  gen: [1x1 struct]
rotor: [1x1 struct]
pitch: [1x1 struct]
  top: [1x1 struct]
 ctrl: [1x1 struct]
 meas: [1x1 struct]
```

Farm controller configuration

The build in farm controller, shown in Figure 7.2, is only meant as a simple example.

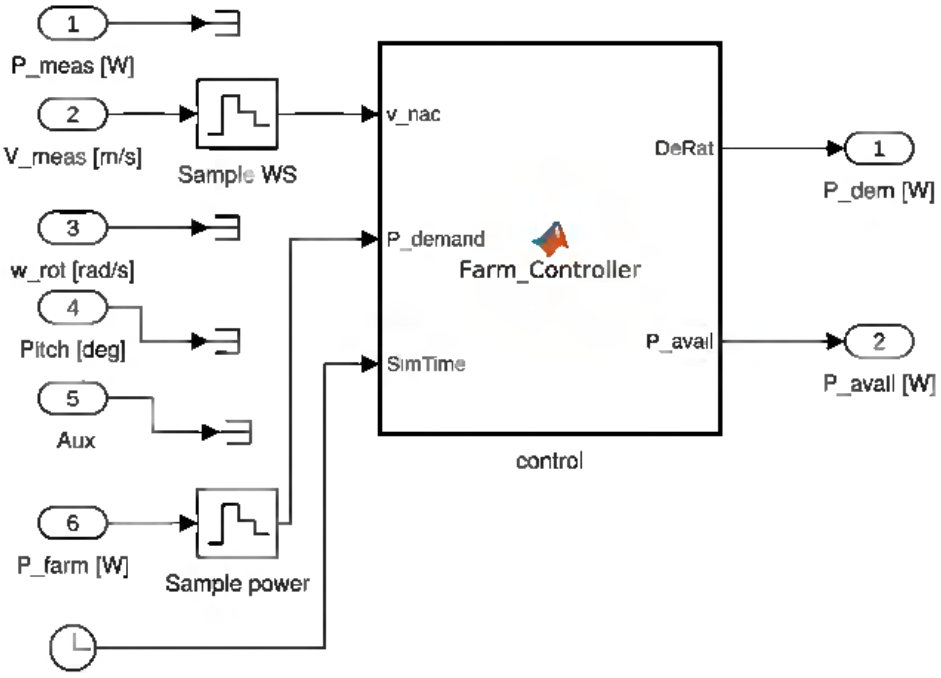

Figure 7.2: Simple farm controller used in SWF.

In previous versions, it sends specific power set points to the turbines based on measured turbine powers and nacelle wind speed. Also it outputs an estimate of available power to the system operator block in Figure 7.1. Notice also that the single turbine controller for the first two turbines NREL5MW and DTU10MW only has an input for the power reference. Derating is then only possible by setting this power reference lower than available power. A probably better way of derating, below rated, is relative power derating which is included in the DTU10MWCLWC turbine controller.

Therefore, the current version of the farm controller m script includes both options for absolute and relative derating. A `Mode` switch is used to select different modes as follows:

- Mode= 0: Original distribution after available power using absolute power reference.
- Mode= 1: Original distribution after available power but using relative derating.
- Mode= 2: Open loop signal.
- Mode= 3: Full power on all turbines.

All the above modes can, of course, be changed by the user of SWF by building/rebuilding blocks to SWF.

Single turbine controller based on NREL5MW

For the two turbines NREL5MW and DTU10MW the single turbine controllers are based on the NREL5MW controller documented in Jonkman et al. (2009). This controller is build using Simulink blocks and the parameters are given in the struct `wt`. For example the proportional gain in the PI pitch controller parameters are in `wt.ctrl.pitch.Pgain` with the rest of the parameters being:

```
wt.ctrl.pitch =
  struct with fields:
       beta: [476x1 double]
        pwr: [1000000 2000000 3000000 4000000 5000000 5290000]
         gs: [6x476 double]
      Pgain: -0.1965
      Igain: -0.0842
    ratelim: 8
       ulim: 90
       llim: 0
```

Single turbine controller based on OpenDiscon

For the third turbine DTU10MWCLWC the controller is written in C and then implemented in Simulink via a wrapper. The parameters and the algorithms in this controller can be changed by changing them in the C code. See https://github.com/ielorza/OpenDiscon for more information on this controller.

OpenDiscon provides the means to integrate the CL-Windcon controller in a Simulink model via a C MEX S-Function. This requires a C mex compiler (e.g. gcc for Linux; run mex -setup in Matlab to see if it is there). Download, compilation and use are as follows:

1. Clone OpenDiscon with Git: open the directory where you wish OpenDiscon to be cloned in a terminal and run

   ```
   git clone https://github.com/ielorza/OpenDiscon.git
   ```

2. Checkout the right branch: OpenDiscon is actively developed by IK4-IKERLAN, so the latest features are likely to be in the dev branch. To see all branches, open your local OpenDiscon directory in a terminal and run

   ```
   gitk -a
   ```

   To checkout the dev branch run

   ```
   git checkout dev
   ```

   NOTE: you are by no means limited to the dev branch. OpenDiscon is a regular Git repository, and you may navigate its versions with Git at will. For comprehensive Git documentation visit https://git-scm.com/doc.

3. Initialize and update submodules: OpenDiscon uses OpenWitcon as a submodule. To initialize and update it, run

   ```
   git submodule init
   git submodule update
   ```

   That will checkout the right version of OpenWitcon for your version of OpenDiscon.

4. Create the build directory: OpenDiscon needs to be built prior to its use, so you need a directory for that build. Open your local OpenDiscon directory in a terminal and run

   ```
   mkdir build
   cd build
   cmake -DISTRIBUTION=S-Function ..
   ```

This will create a directory called 'build' and will put some files in it. The ones needed are:

(a) makeOpenDiscon.m : this contains a function for easy compilation.

(b) OpenDiscon_block.mdl : this contains an OpenDiscon Simulink block for use in Simulink models.

NOTE: SimWindFarm's DTU10MWCLWC turbine model already includes the OpenDiscon block, but you still need to compile OpenDiscon for it to work.

5. Compile OpenDiscon: for the OpenDiscon Simulink block to work, it needs a MEX S-Function. To get it, open the 'build' directory in Matlab and run

   makeOpenDiscon

   That will build a file called 'OpenDiscon.mexa64', or something similar, depending on your system. Put it where Matlab can find it when you run your Simulink model.

OpenDiscon controller parameters are modifiable at compile time. All CL-Windcon controller parameters are documented in the OpenDiscon documentation. To build it, after steps 1-3 above, open your local OpenDiscon directory in a terminal and run

cd doc

doxygen

This will build html documentation in a directory called 'html'. To browse it, open file 'html/index.html'. The CL-Windcon controller parameters, including how to modify them, are documented in html/clwindcon.html. They are all defined as constants in file:

CONFIGURATION/CL-Windcon/src/ikClwindconWTConfig/ikClwindconWTConfig.c.

NOTE: the controller allows for considerably more flexibility than that achieved by modifying these parameters, which merely provide a conveniently high-level approach for those without the time or inclination for lower-level parameter modification. The same source file is a good example of how one would go about said lower-level work, and the rest of the OpenDiscon documentation goes into much detail about it.

After modification of OpenDiscon parameters, it is necessary to repeat step 5 above for the changes to take effect.

7.3 Performing simulations

7.3.1 Generic 3 turbine case

The first example is in partial load where the effect of derating is easy to see. The following parameters are chosen:

- wind speed 8 m/s
- Turbulence intensity 0.1
- Direction along the row
- Time 1200 sec.
- Operation mode
- Full power
- At start: Full power, at 400 sec 0.2 derating of front turbine, 800 sec. 0.2 derating of second turbine

Wind field generation

The wind field is generated by running `GenWind.m`. The body of this file is listed below and should be self explanatory.

```
%% Parameters
PlotWind= 1; PrintStats=1;
% The farm has three turbines, all of them the DTU10MW turbine
TurbineName= 'DTU10MWCLWC';
D= 178.3;                                % Find in parameters if possible
% The position of the turbines.
% (make sure that the turbines are within the grid of the wind field)
% CLWINDCON 3 turbine farm in CLWINDCON orientation where wind is blowing
% in the y direction and in rotor diameters
farm.pos=[0 0 0.5; %X values
   0 5 10]; %Y values;
% File and dir name
FileName='CLWindcon3WTWind.mat';
DirName= './';
% Average wind speed in m/s
U0= 8;
% Turbulence intensity
% 2018-02-27 10:26:36 tk: Notice that the below gives a turbulence
intensity of
% Sigmau/Ua= Ti*(0.75*Ua+5.6)/Ua i.e. not Ti which is strange
% See gen_windfield_no_taylor.m for details
Ti= 0.1;                                 % For following the used
     % specifications IEC 61400-1(2005)
```

```
Ti= 0.1*U0/(0.75*U0+5.6);                % For exactly Ti on single wind
speeds
% Grid spacing in m
d= 15;
% Length of the wind field in m
Lx= 2e3;
% Width of the wind field in m
Ly= 1e3;
% Maximum simulation time in s
SimTime= 1200;

%% Definitions etc.
% Give all turbines the same name
farm.turbines= repmat({TurbineName},1,size(farm.pos,2));
% SimWindFarm has wind along the x-axis so rotate -pi/2 and scale with D;
% Rotate by v using [cos(v) -sin(v); sin(v) cos(v)]
farm.pos= [0 1; -1 0]*farm.pos*D;

%% Algorithm
% Generate the wind field
wind= gen_windfield_no_taylor(U0,Ti,d,Lx,Ly,farm,SimTime);
% Save the wind field
save([DirName FileName],'wind');
```

Wind farm generation

The wind farm is generated by running `GenWindFarm.m`. The body of this file is listed below:

```
%% Parameters
% The farm has three turbines, all of them the DTU10MW turbine
TurbineName= 'DTU10MWCLWC';              % With CLWindcon controller
D= 178.3;                                % Find in parameters if possible
% The position of the turbines should be available e.g. from GenWind.m.
% Filename
slxfile='CLWindcon3WT.slx';
DirName= './';

%% Definitions etc.
% Give all turbines the same name
farm.turbines= repmat({TurbineName},1,size(farm.pos,2));
slxname= slxfile(1:end-4);               % Exclude extension
% Load the wind data used
l=load('CLWindcon3WTWind.mat');
wind=l.wind;

%% Algorithm
% Generate the wind farm
gen_windfarm_no_taylor_add_turbulence(slxfile,DirName,farm,wind);
% Open wind farm Simulink diagram
open(slxfile);
```

The last line of code above opens a Simulink model which should look like figure 7.1. This model should be able to run in both normal and accelerator mode.

Results

The Simulink file can now be run in the model window. The results can be accessed in many ways e.g. by using the Simulink "simulation data inspector", which is opened by clicking the small symbols looking like a WI-FI symbol at the input and outputs from the blocks. In addition, the simulation generates an output structure `logsout` (Simulink.SimulationData.Dataset) which holds many signals. Based on `logsout` the file `PlotSimOutput.m` plots some of the most relevant signals as seen below in figure 7.3

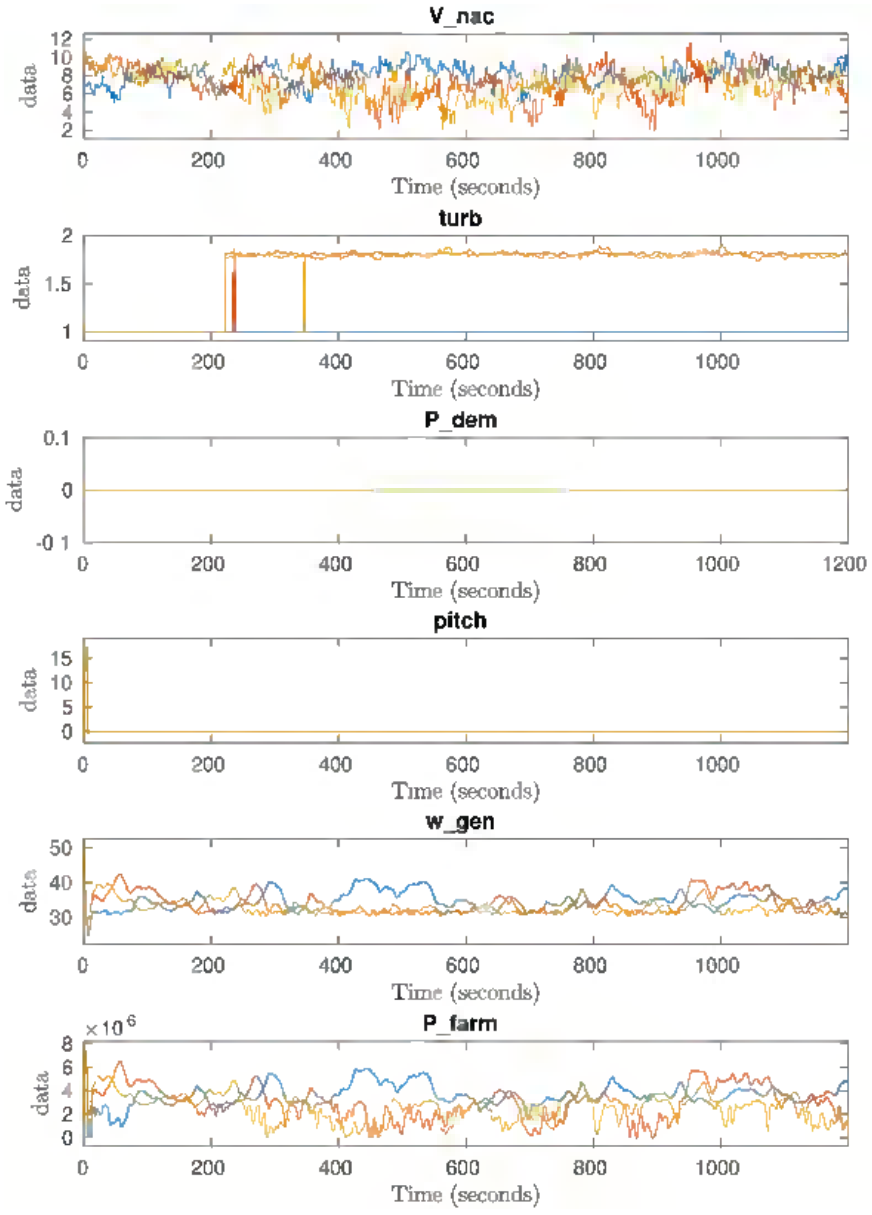

Figure 7.3: 3 wind turbine case simulated at 8 m/s with all turbines at max power. Legend: blue, front turbine, red, middle turbine and yellow, last turbine.

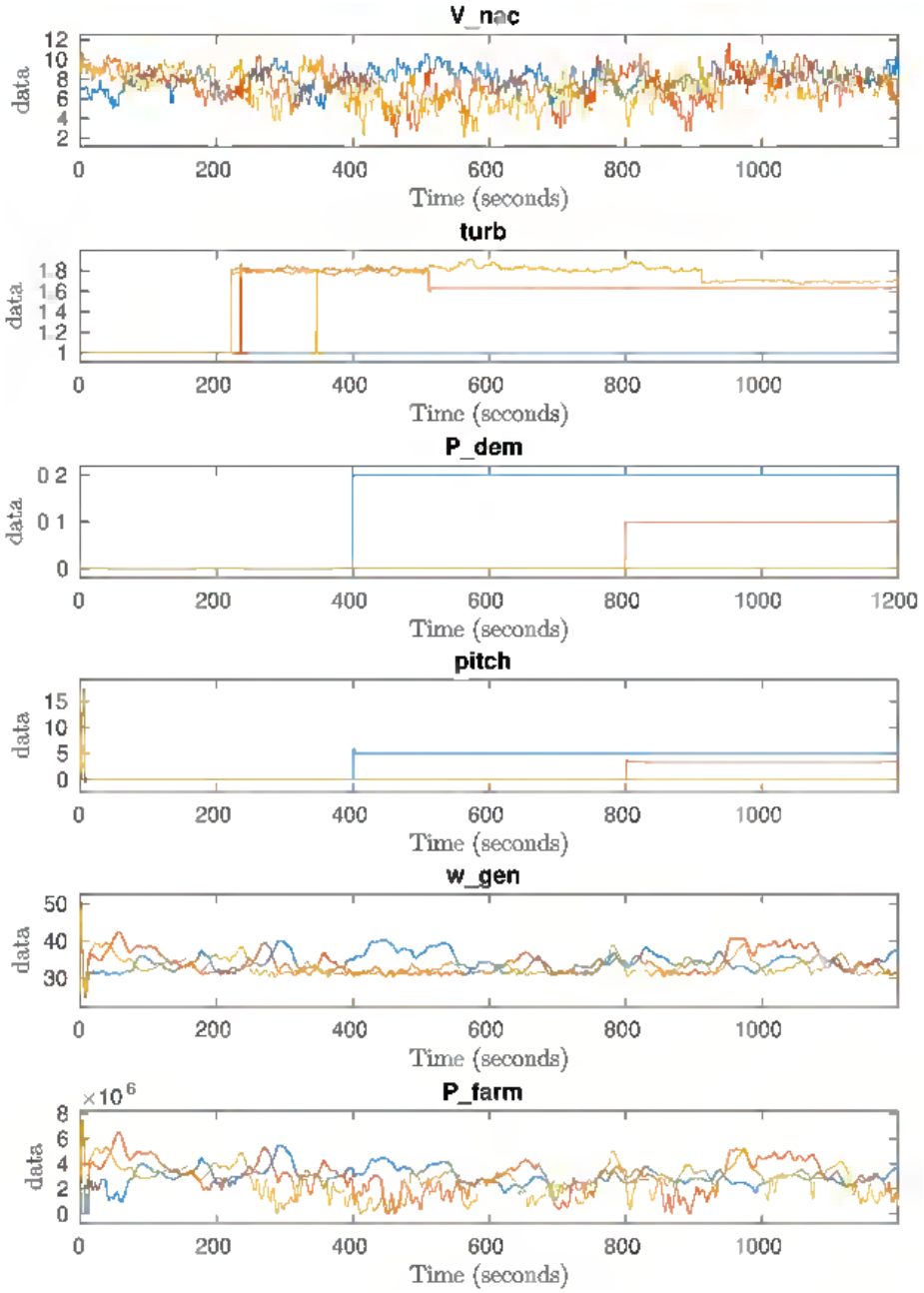

Figure 7.4: 3 wind turbine case simulated at 8 m/s with the front turbines derated in steps as indicated in subplot 3.

The first subplot is the nacelle wind speed. The second is the total turbulence, included added turbulence. The third is not absolute power demand but power derating demand. Forth subplot is the pitch, fifth, the generator speed and sixth, the generator power. In Figure 7.3 the Mode switch in the farm controller is 3 while it is 2 in Figure 7.4.

7.4 References

C. Bak, F. Zahle, R. Bitsche, T. Kim, A. Yde, L. C. Henriksen, A. Natarajan, and M. H. Hansen. Description of the dtu 10 mw reference wind turbine. Technical Report Report-I-0092, DTU Wind Energy, 2013.

A. Brand and J. Wagenaar. A quasi-steady wind farm flow model in the context of distributed control of the wind farm. In *European Wind Energy Conference and Exhibition (EWEC), April 20-23, 2010, Warsaw, Poland*. EWEA, EWEA, 2010.

M. Churchfield and S. Lee. Nwtc design codes - sowfa. `http://wind.nrel.gov/designcodes/simulators/SOWFA/`, 2013.

P. Fleming, P. Gebraad, M. Churchfield, S. Lee, K. Johnson, J. Michalakes, J.-W. van Wingerden, and P. Moriarty. Sowfa + super controller user's manual. Technical Report Technical Report NREL/TP-5000-59197, National Renewable Energy Laboratory (NREL), 2013.

J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, 1617 Cole Boulevard, Golden, Colorado, USA, 2009.

14 APPENDIX A

Appendix: Detailed mathematical description for SimWindFarm

Wind turbine

Simplified turbine model for Simulink

The turbines included in the toolbox are either the NREL5MW or DTU10MW virtual turbine. The models used are dynamic but relatively simple. The equations and parameters are described in details below.

The turbine simulation model used in SimWindFarm is a simplified aeroelastic model based on lookup tables for the aerodynamics (C_P, C_T), a simple 3rd order drive train model, a 1st order generator model, 1st order pitch actuator, and a 2nd order tower dynamics.

The turbines NREL5MW are controlled using the control strategy from Jonkman et al. [2009], which includes a simplified start up procedure and pitch control for full load operation. The turbine DTU10MW has a similar controller but with parameters adapted to DTU10MW. The DTU10MWCLWC turbine uses the OpenDiscon controller `https://github.com/ielorza/OpenDiscon` IK4-IKERLAN [2017].

Figure 1 illustrates the structure of the turbine model, with inputs being the wind speed at the nacelle, the average wind speed over the rotor area (effective wind speed), and the power reference supplied by the farm controller. The outputs available to the farm controller are produced power, measured generator speed, nacelle wind speed, and blade pitch angle.

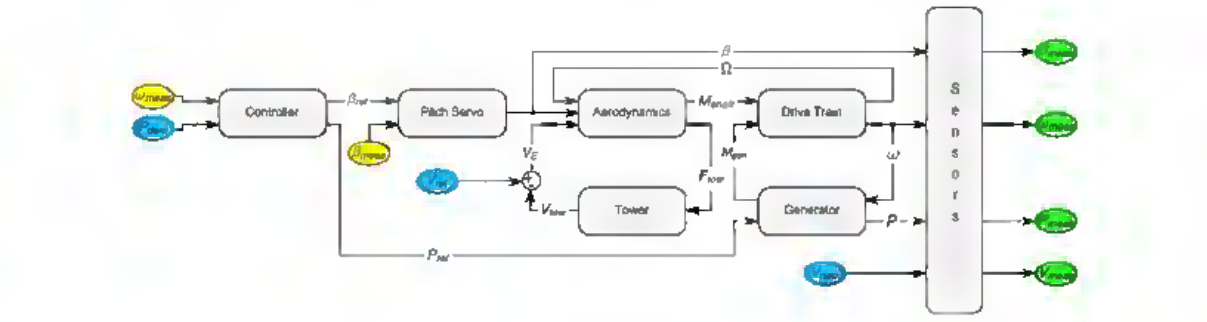

Figure 1: Top level structure of SWF as illustrated by the Simulink diagram.

The variables used in the turbine model are shown in table `tab:Variables`.

Variable	Description	unit
Ω	rotor speed	$^{rad}/_{s}$
ω	generator speed	$^{rad}/_{s}$
β	pitch angle	degrees
M_{shaft}	main shaft torque	Nm
M_{gen}	generator torque	Nm
F_{tow}	tower thrust	N
P_{dem}	power demand	W
P_{ref}	generator power reference	W
β_{ref}	pitch angle reference	degrees
V_{nac}	nacelle wind speed	$^{m}/_{s}$
V_{rot}	average wind speed over the rotor	$^{m}/_{s}$
V_E	effective wind speed	$^{m}/_{s}$
V_{meas}	measured wind speed	$^{m}/_{s}$
β_{meas}	measured pitch angle	degrees
P_{meas}	measured produced power	W
ω_{meas}	measured generator speed	$^{rad}/_{s}$

Aerodynamics

The aerodynamics of the turbine can be described using two static relationships,

$$M_{shaft} = \frac{1}{2} v_{rot}^3 \rho A C_P(\lambda, \beta) \Omega^{-1}$$
$$F_{tow} = \frac{1}{2} v_{rot}^2 \rho A C_T(\lambda, \beta)$$

where C_P and C_T are two look-up tables derived from the geometry of the blades with inputs tip speed ration ($\lambda = {R\Omega}/{v_{rot}}$) and pitch angle β. The parameters are air density ,ρ, and rotor disc area, A.

Drive Train

The 3rd order drive train model is based on two rotating shafts connected through a gearbox with torsion spring constant K_{shaft}, viscous friction B_{shaft}, and gear ration N.

$$\dot{\Omega} = \frac{1}{I_{tot}} \left(M_{shaft} - \phi K_{shaft} - \dot{\phi} B_{shaft} \right)$$
$$\dot{\omega} = \frac{1}{I_{gen}} \left(-M_{gen} + \frac{1}{N} (\phi K_{shaft} + \dot{\phi} B_{shaft}) \right)$$
$$\dot{\phi} = \Omega - \frac{1}{N} \omega$$

where ϕ is the shaft torsion angle and I_{gen}, I_{rot} are the inertias of the generator and rotor respectively.

Generator

In the baseline NREL 5MW turbine there is no generator model, but a simple 1st order model is included in this benchmark, with input P_{ref} and time constant τ_{gen}.

$$\dot{M}_{gen} = \frac{1}{\tau_{gen}} \left(\frac{P_{ref}}{\omega} - M_{gen} \right)$$

Notice that the baseline turbine assumes a torque reference, but a power reference is used here instead.

Tower

The tower deflection, z, is modeled as a spring-damper system with spring constant K_{tow} and damping B_{tow}.

$$\ddot{z} = \frac{1}{m_{tow}} \left(F_{tow} - K_{tow} z - B_{tow} \dot{z} \right)$$

Pitch Servo

The pitch actuator does not use the NREL model which is a spring-damper system and not used in their FAST simulation. Instead a second order system with a time constant of τ_β and input delay λ from input u_β to pitch rate $\dot{\beta}$ is used.

The actuator is controlled by a proportional regulator with constant K_{beta} resulting in a pitch servo.

$$\ddot{\beta} = \frac{1}{\tau_\beta} \left(u_\beta^\lambda - \dot{\beta} \right)$$
$$u_\beta = K_\beta \left(\beta_{ref} - \beta_{meas} \right)$$

Control

Notice that the different turbines in the Simulink SimWindFarm turbine library can have different controllers. Currently (*<2018-01-18 Thu>*) NREL5MW and DTU10MW has the NREL5MW type controller explained below while the DTU10MWCLWC has the IK4-IKERLAN controller IK4-IKERLAN [2017].

Control based on NREL5MW

The control strategy for the NREL5MW is divided into two regions; 1) partial load and 2) full load.

In region **1**) the control is a simple lookup-table with generator speed as input and generator power reference as output. The blade pitch is kept constant at 0, see figure 2 below.

In region 2) the generator power reference is kept constant at the rated power while the rotor speed is controlled using the blade pitch angle, by a gain scheduled PI controller.

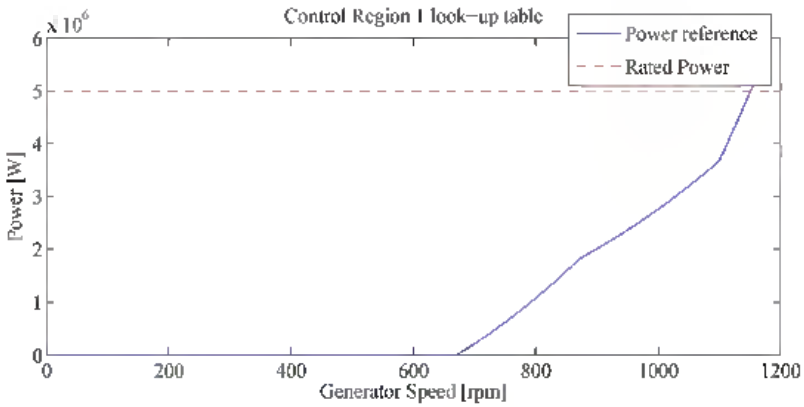

Figure 2: Turbine Pitch Control Regions.

The gain scheduled PI controller is shown in the equation below, and the gains are based on linearization of the power production sensitivity to blade pitch angle.

$$\beta_{ref} = K_P(\beta)\omega_{err} + K_I(\beta)\int_0^t \omega_{err}$$

$$K_{P/I}(\beta) = K_{P/I,0}\frac{\beta_2}{\beta_2 + \beta}$$

where $\omega_{err} = \omega_{rated} - \omega$, and $K_{P/I}$ are the proportional and integral gains, $K_{P/I,0}$ is the base gain at $\beta = 0°$ and β_2 is the pitch angle where the pitch sensitivity is doubled.

The controller presented in Jonkman et al. [2009] always operates at full power rating and in order to be able to derate the turbine the control strategy has been altered slightly by letting the dynamic power rating change the transition point between region 1 and region 2.

Wind and wake part

Wind field modelling

The wind simulation and modelling can be divided into an ambient field model, which describes the wind field if no turbines where present and a wake model which describes the turbines effect on the ambient wind field. The following description is relevant for both SimWindFarm, expect where stated otherwise, i.e., where there are differences a subsection titled SWF Taylor will describe the original version and SWF No Taylor will describe the new version.

SWF Taylor

In this version of SWF both ambient field model and wake model assume Taylor's frozen turbulence hypothesis for inviscid flow Davidson [2004] to be true. This greatly simplifies the effort of generating an ambient wind field, and provides relatively simple equations for the wake effect models.

SWF No Taylor

In this version of SWF the wake model assumes Taylor's frozen turbulence hypothesis for inviscid flow Davidson [2004] to be true. This assumption is, however, not made in the ambient wind field model. The coherence between two points separated by a distance downwind will be one if Taylor's frozen turbulence hypothesis is assumed, i.e., the wind at these points will be identical except for a time delay. This is not realistic, which is the reason for introducing a version without the assumption.

Ambient wind field

The ambient wind in a wind farm is usually described by spectral matrices describing the wind speed variation at a number of points in the wind farm and their relation using the method described in Veers [1988]. The wind model assumes a constant mean wind speed and a zero lateral mean wind speed, i.e. the mean wind direction is constant in the longitudinal direction.

- SWF Taylor Due to the frozen turbulence assumption in this version it is only necessary to simulate stochastic processes in the first line in the lateral direction in the park. These velocities are then assumed to travel with the average wind speed in the longitudinal direction. In this way the wind speeds at downwind grid points will eventually be generated.
- SWF No Taylor lateral wind component, more or less, is only used for wake meandering and is therefore generated using Taylor - The lateral component, which is the largest part of wind turbines see, is generated for x point at each turbine according to the spectrum described next. where the lateral wind is implemented as a number of stochastic processes, one for each point.

Turbulence Spectrum

The wind field is generated according to the recommendations in IEC 61400-3 concerning offshore turbines, which state that for non-site specific wind conditions the parameter values in IEC 61400-1 (2005) Commission [2005] can be used. The spectrum used is the Kaimal spectrum

$$S_k(f) = \sigma_k^2 \frac{4\frac{L_k}{U}}{(1+6f\frac{L_k}{U})^{\frac{5}{3}}}$$

where L_k is velocity component integral scale parameter, f is the frequency in Hertz, k denotes the velocity component, U is the hub height mean wind speed and σ_k is the variance determined by the turbulence intensity, T_i, given by [Commission, 2005, P. 24, eqn. (11)]

$$\sigma_x = T_i\left(\frac{3}{4}U + 5.6\right)$$
$$\sigma_y = 0.8\sigma_x$$

Notice that the above relation between σ_x and T_i results in $T_i \neq \sigma_x/U$ but rather

$$\frac{\sigma_x}{U} = T_i\left(\frac{3}{4} + \frac{5.6}{U}\right)$$

The coherence between two point separated by distance l is given as

$$C_k(f) = e^{-c_k f \frac{l}{U}}$$

where c_k is the coherence parameter, which depends on the separation direction. In SWF three coherence parameters are used, c_{xx} is used for the coherence of longitudinal wind speed component between points separated by a longitudinal distance, c_{xy} is used for the coherence of longitudinal wind speed component between points separated by a lateral distance, and c_{yy} is used for the coherence of lateral wind speed component between points separated by a lateral distance.

As the hub height is assumed to be above $60m$ then $L_x = 340.2$ and $L_y = 113.4$ and according to Kristensen and Jensen [1979] c_{xy} and c_{yy} can be set to 7.1 and 4.2 respectively.

- SWF Taylor: In this version only c_{xy} and c_{yy} is needed for wind generation as Taylor's frozen turbulence is assumed and only point separated by a lateral displacement is needed. Notice that the simple wind field model using the frozen turbulence assumption has a coherence of 1 in the longitudinal direction, which does not conform to IEC 61400-1, and care should thus be taken not to design controllers that rely on this fact.
- SWF No Taylor: Generating wind in this version is based on Sørensen et al. [2002]. The coherence between each turbine is needed and as turbines need not be displaced only in the lateral or longitudinal direction the following equation is used

  $$c_{rc}(\alpha) = \sqrt{(c_{xx}\cos\alpha)^2 + (c_{xy}\sin\alpha)^2}$$

  where α is the angle between wind direction and a line between the two turbine r and turbine c . Furthermore, the delay from turbine to turbine is needed, which can be calculated as

  $$\tau_{rc} = \frac{d_{rc}\cos\alpha}{U_0}$$

  where d is the distance between turbines, α the angle between turbines, and U_0 is the mean wind speed. Now the cross spectrum between turbines can be found from

  $$S_{rc}(f) = C_{rc}(f)\sqrt{S_{rr}(f)S_{cc}(f)}\exp\left(-j2\pi f\tau_{rc}\right)$$

  where S_{rc} is the cross spectrum between turbine r and turbine c, C_{rc} is the coherence, S_{rr} is the auto spectrum at turbine r, S_{cc} is the auto spectrum at turbine c, and τ_{rc} is the time delay from turbine r to turbine c.

Wake effects

It has been shown in Larsen et al. [2008] that a good approximation of the meandering is to consider the wake center as a passive tracer which moves downwind with the mean wind speed. It is, therefore, possible to rank turbines relative to each other as being either downwind or upwind.

For wake effect calculations at a given turbine it is only necessary to consider upwind turbines, and as this relationship is fixed it considerably simplifies the calculations.

SWF considers three wake effects; deficit, expansion and center, where wake deficit is a measure of the decrease in downwind wind speed, wake expansion describes the size of the downwind area affected by the wake and wake center defines the lateral position (meandering) of the wake area, see the figure below and Larsen et al. [2008]. The wake effects are illustrated in figure 3.

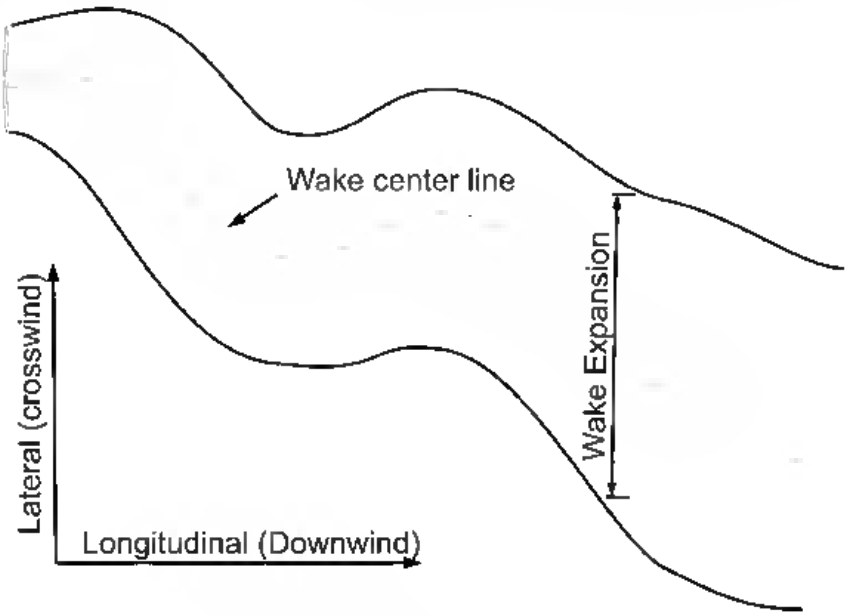

Figure 3: Wind turbine wake illustration.

Expressions for wake deficit, center and expansion was developed in Frandsen et al. [2006], Jensen [1983]. The wake center, expansion and deficit at a given point p downwind from a turbine at time t_1 is defined by the wind field at the turbine and its coefficient of thrust at the time of tracer release, t_0,

$$t_0 = t_1 - \frac{d}{U}$$

where d is the longitudinal distance between the upwind turbine and p.

Wake Expansion

- SWF Taylor: In this version of SWF in particular we use the equation below from Jensen [1983] to calculate the wake expansion radius, which is a simplified model independent of C_T.

$$e_i(d) = 2R\sqrt{1 + \frac{d}{4R}} = \sqrt{4R^2 + dR},$$

  where R is the rotor radius.

- SWF No Taylor: In this version of SWF, we use the equation below from Frandsen et al.

[2006] to calculate the wake expansion (radius of the wake, $e_i(d)$)

$$e_i(d) = \frac{1}{2}\left(\beta^{\frac{k}{2}} + \alpha\frac{d}{D_0}\right)^{\frac{1}{k}} D_0,$$

where D_0 is the rotor diameter, α and k are parameters which here are set to 0.5 and 2, respectively. Furthermore,

$$\beta = \frac{1+\sqrt{1-C_T}}{2\sqrt{1-C_T}}.$$

Wake Center

The wake center is computed as a passive tracer, such that the center at time t_k is a function of the center at time t_{k-1} and the average lateral wind speed, over the wake area,

$$W_i(d, t_{k-1}) = \{(x, y)|x = d \wedge y \in W_y\}$$
$$W_y = [w_i(d, t_{k-1}) - e_i(d); w_i(d, t_{k-1}) + e_i(d)]$$
$$w_i(d, t_k) = w_i(d, t_{k-1}) + \bar{v}_{y,i}\left(d, W_i(d, t_{k-1})\right)$$

where $w_i(d, t_k)$ is the wake center at time k, $W_i(d, t_{k-1})$ is the wake area distance d from the turbine at time t_{k-1} and $\bar{v}_{y,i}\left(d, W_i(d, t_{k-1})\right)$ is the average lateral wind speed over the wake area.

Wake Deficit

- SWF Taylor: The deficit from turbine i at distance d can according to Jensen [1983] be approximated as

$$dU_i(d) = \frac{U(d)}{U_0} = 1 - 2a_i(t_0)\left(\frac{R}{R+\kappa d}\right)^2$$

where dU_i is the wind deficit, $\kappa = 0.1$, $U(d)$ is the wind speed distance d down wind, U_0 is the ambient wind speed.a_i is the induction factor of turbine i at time t_0 and R is the radius of the rotor.

Instead of using the induction factor, a more realistic result is obtained by using the simulated thrust coefficient. The thrust coefficient is given by

$$C_T = 4a_i(1 - a_i)$$

According to Frandsen et al. [2006], the above expressions can be approximated by

$$dU_i(d) \approx 1 - \frac{1}{2}C_{T,i}(t_0)\left(1 + \frac{d}{4R}\right)^{-1}$$

where $C_{T,i}(t_0)$ is the coefficient of thrust of turbine i at time t_0.

- SWF No Taylor: In this version of SWF the deficit behind a single turbine is given by the equation below from Frandsen et al. [2006]

$$c = \frac{U(d)}{U_0} \approx 1 - \frac{C_T}{2}\frac{D_0^2}{D^2(d)}$$

where $U(d)$ is the wind speed distance d down wind, U_0 is the ambient wind speed, C_T is the thrust coefficient, D_0 is the rotor diameter, and $D(d)$ is the wake diameter distance d down wind.

Wake Merging

- SWF Taylor: Using equations the equations for meandering, expansion and deficit it is possible to calculate the wake contributions for each turbine and this enables us to calculate the actual wind speed at any point in the farm as

$$v_s(x,y) = u_s(x,y)\Pi_{i\in I} a_i(d_i)$$

where I contains the indices of all turbines where the point (x,y) is in its wake area, $y \in W_i(x - P_{x,i})$, with $P_{x,i}$ being the longitudinal position of the i'th turbine.

- SWF No Taylor: In this version of SWF a slightly different approach is taken. In Frandsen et al. [2006] a row of turbines is considered and the deficit at turbine $n+1$ can be described as

$$c_{n+1} = 1 - \left(\frac{D_n^2}{D_{n+1}^2}(1 - c_n) + \frac{C_{T_n}}{2}\frac{D_R^2}{D_{n+1}^2}c_n\right)$$

where $c_{n+1} = {}^{U_{n+1}}/_{U_0}$ is the deficit at turbine $n+1$, D_n is the wake diameter at turbine n, D_{n+1} is the diameter at turbine $n+1$, c_n is the deficit at turbine n, C_{T_n} is the thrust coefficient of turbine n and D_R is the diameter of the rotor.

Model of Added Turbulence Intensity

- SWF No Taylor: The model of added turbulence intensity which is used here is the one suggested in the IEC 61400-1 Commission [2014] which comes from Frandsen [2007] and is given by

$$I_{add,j} = \frac{\sigma_{add,j}}{U_j} = \frac{1}{1.5 + 0.8\frac{s_{ij}}{\sqrt{C_{T,i}}}}$$

where $\sigma_{add,j}$ is the added standard deviation in the wind field at the j'th WT which is in wake; U_j is the effective wind speed at the j'th WT; s_{ij} is the spacing in rotor diameters between the wake generating WT and the WT in wake; $C_{T,i}$ is the thrust coefficient of the wake generating WT.

References

I. E. Commission. International standard 61400-1 - wind turbines. Technical report, International Electrotechnical Commission, 2005.

I. E. Commission. 61400-1 ed iec:2014(e) draft. Technical report, International Electrotechnical Commission, 2014.

P. A. Davidson. *Turbulence*. Oxford University Press, 2004.

S. Frandsen, R. Barthelmie, S. Pryor, O. Rathmann, S. Larsen, and J. Højstrup. Analytical modelling of wind speed deficit in large offshore wind farms. *Wind Energy*, 9:39–53, 2006.

S. T. Frandsen. Turbulence and turbulence- generated structural loading in wind turbine clusters. Technical report, Risø National Laboratory, 2007. Danish Doctor Technices Thesis Risø-R-1188(EN).

IK4-IKERLAN. Open source implementation of the legacy ghbladed external controller interface. Internet `https://github.com/ielorza/OpenDiscon`, 2017.

N. Jensen. A note on wind generator interaction. Technical Report ris-m-2411, Risø National Laboratory, 1983.

J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-mw reference wind turbine for offshore system development. Technical Report NREL/TP-500-38060, National Renewable Energy Laboratory, 1617 Cole Boulevard, Golden, Colorado, USA, 2009.

L. Kristensen and N. O. Jensen. Lateral coherence in isotropic turbulence and in the natural wind. *Boundary-Layer Meteorology*, 17:353–373, 1979. doi: http://dx.doi.org/10.1007/BF00117924.

G. C. Larsen, H. A. Madsen, K. Thomsen, and T. J. Larsen. Wake meandering: A pragmatic approach. *Wind Energy*, 11:377 395, 2008.

P. Sørensen, A. D. Hansen, P. André, and C. Rosas. Wind models for simulation of power fluctuations from wind farms. *Journal of Wind Engineering and Industrial Aerodynamics*, 90: 1381–1402, 2002.

P. S. Veers. Three-dimensional wind simulation. Technical Report SAND88-0152 UC-261, Sandia National Laboratories, 1988.