Theoretical Foundations for Optimal Control of Floating Offshore Wind Farms

by

Ali Cherom Kheirabadi

B.Eng., Dalhousie University, 2014

M.A.Sc., Dalhousie University, 2016

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Mechanical Engineering)

The University of British Columbia
(Vancouver)

December 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Theoretical Foundations for Optimal Control of Floating Offshore Wind Farms

submitted by **Ali Cherom Kheirabadi** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Mechanical Engineering**.

Examining Committee:

Ryozo Nagamune, Mechanical Engineering
Supervisor

Rajeev Jaiman, Mechanical Engineering
Supervisory Committee Member

Guy Dumont, Electrical Engineering
University Examiner

Martin Ordonez, Electrical Engineering
University Examiner

Jan-Willem van Wingerden, Mechanical, Maritime, and Materials Engineering, Delft University of Technology, The Netherlands
External Examiner

Additional Supervisory Committee Members:

Anasavarapu Srikantha Phani, Mechanical Engineering
Supervisory Committee Member

Bhushan Gopaluni, Chemical Engineering
Supervisory Committee Member

Abstract

Due to a phenomenon termed the *wake effect*, wind turbines that are placed in close proximity within wind farms interact aerodynamically. In short, each turbine generates a wake within which wind speeds are reduced, and these wakes overlap with the rotors of machines located downstream. This interaction diminishes power production in wind farms by up to 60 %. Using a process referred to as *wind farm control*, individual wind turbines may be operated in a manner that increases power production from the collective.

This thesis investigates the potential of a wind farm control strategy named *yaw and induction-based turbine repositioning* (YITuR) that is specifically compatible with floating offshore wind farms. Since floating platforms are anchored to the seabed using slack mooring line cables, each turbine may be repositioned in real-time using the aerodynamic forces exerted on its rotor. By relocating floating platforms accordingly, the overlap area between the wakes generated by upstream turbines and the rotors of downstream machines may be reduced; leading to an increase in wind farm efficiency.

The potential of YITuR is assessed through several steps. First, a steady-state model of floating offshore wind farms is constructed and stationary optimization studies are carried out to determine the potential of YITuR under idealized steady wind conditions. Major findings from this study are that wind farm efficiency may increase by more than 40 % using YITuR over traditional wind farm operation; however, these benefits are strongly influenced by mooring system designs. Second, a dynamic floating wind farm model is developed to evaluate the performance of real-time control systems. Third, due to the non-convexity of the YITuR control problem, novel distributed economic model predictive control (DEMPC) theory is developed to guarantee power maximization. Existing DEMPC algorithms do not offer such a guarantee in the presence of non-convex objective functions. Finally, the DEMPC algorithm is evaluated using the dynamic simulation tool. Neural networks are used to estimate the dynamics of floating platforms in order to expedite decision-making in DEMPC. Simulation results indicate gains of 20 % in energy production when YITuR replaces traditional wind farm operation.

Lay Summary

Wind turbines that are clustered in close proximity within wind farms interact aerodynamically. This interaction decreases the efficiency of wind farms and results in 20 to 30 % loss of annual revenue. One approach to recuperating these losses is wind farm control. This strategy involves operating individual turbines in a manner that mitigates the negative effects of aerodynamic interaction; thus increasing wind farm efficiency.

This thesis investigates the potential of one particular wind farm control technique that is compatible with floating offshore wind farms and is referred to as *turbine repositioning*. In this method, each floating wind turbine uses the aerodynamic force of the wind to reposition itself along the ocean surface with the goal of reducing the scale of aerodynamic interaction and increasing wind farm efficiency. This research therefore delivers economic and environmental benefits by raising the competitiveness of floating offshore wind farms relative to fossil fuel-based energy generation methods.

Preface

The general topic of research investigated in this thesis was recommended by the author's supervisor, Ryozo Nagamune. Beyond this task, all efforts related to the design and implementation of the conducted research are the sole work of the author, Ali Cherom Kheirabadi. Further, the preparation of this thesis is also the sole work of the author with guidance and review offered by the research supervisor.

The main body chapters of this thesis comprise a collection of works that have been or are intended for publication in peer-reviewed journals. The following list contains information on each article including the corresponding chapter:

- The content from Chapter 2 has been published and is available in:
    - A. C. Kheirabadi, and R. Nagamune. A quantitative review of wind farm control with the objective of wind farm power maximization. *Journal of Wind Engineering and Industrial Aerodynamics*, 192(May), 45–73, 2019.
- The content from Chapter 3 has been published and is available in:
    - A. C. Kheirabadi, and R. Nagamune. Real-time relocation of floating offshore wind turbine platforms for wind farm efficiency maximization: An assessment of feasibility and steady-state potential. *Ocean Engineering*, 208(May), 2020.
- The content from Chapter 4 has been submitted for review with the following title and authorship information:

  - A. C. Kheirabadi, and R. Nagamune. A low-fidelity dynamic wind farm model for simulating time-varying wind conditions and floating platform motion.

- The content from Chapter 5 has been submitted for review with the following title and authorship information:

  - A. C. Kheirabadi, and R. Nagamune. Distributed economic model predictive control – Addressing non-convexity using social hierarchies.

- The content from Chapter 6 will be submitted for review with the following title and authorship information:

  - A. C. Kheirabadi, and R. Nagamune. Social hierarchy-based distributed economic model predictive control of floating offshore wind farms.

Table of Contents

List of Tables

List of Figures

Nomenclature

Chapter 1

Acronyms

CFD	Computational fluid dynamics
DEMPC	Distributed economic model predictive control
FOWFSim	Floating Offshore Wind Farm Simulator
FOWFSim-Dyn	Dynamic Floating Offshore Wind Farm Simulator
YITuR	Yaw and induction-based turbine repositioning

Chapter 2

a	Axial induction factor
C_P	Power coefficient
C_T	Thrust coefficient
P	Turbine power output, W
T_{gen}	Generator torque, N·m
x	Downwind displacement of a wind turbine, m
y	Crosswind displacement of a wind turbine, m

Greek letters

β	Blade pitch angle, rad
γ	Nacelle yaw angle, rad
λ	Tip-speed-ratio

Acronyms

AEP	Annual energy production
CFD	Computational fluid dynamics
DEL	Damage equivalent loading
ECN	Energy Research Center of the Netherlands
FAST	Fatigue, Aerodynamics, Structures, and Turbulence
FLORIDyn	Flow Redirection and Induction Dynamics
FLORIS	Flow Redirection in Steady-state
FOWFSim	Floating Offshore Wind Farm Simulator
GHG	Green house gas
IRENA	International Renewable Energy Agency
LCOE	Levelized cost of energy
LES	Large eddy simulation
NREL	National Renewable Energy Laboratory
RANS	Reynolds-averaged Navier-Stokes
RMS	Root-mean-square
RMSE	Root-mean-square-error
SOWFA	Simulator for Wind Farm Applications
UTD-WF	University of Texas at Dallas-Wind Farm
WFSim	Wind Farm Simulator
YITuR	Yaw and induction-based turbine repositioning

Chapter 3

$\mathbf{a}$	Vector containing the axial induction factors of all turbines
A	Rotor swept area of turbine rotors, m^2
A_c	Mooring line cable cross-sectional area, m^2
a_i	Axial induction factor of turbine i
$C_{P,i}$	Power coefficient of turbine i
$C_{P,\max}$	Power coefficient of turbine i under greedy operation
$C_{\mathrm{thrust},i}$	Thrust coefficient of turbine i
D	Diameter of turbine rotors, m
D_c	Mooring line cable diameter, m
E	Mooring line elastic modulus, N/m^2
$\mathscr{F}$	Set containing the indices of all turbines in the wind farm

$f_1(\cdot)$ Newton-Raphson function for computing tensions of a mooring line that is partially resting along the seabed, N

$f_2(\cdot)$ Newton-Raphson function for computing tensions of a mooring line that is fully lifted above the seabed, N

$f_{\text{moor}}(\cdot)$ Mooring line tension lookup function, N

$F_{\text{moor},i}$ Total mooring line force vector acting on the platform of turbine i, N

$\mathbf{F}_{\text{moor},i,j}$ Mooring line force vector exerted by cable j on the platform of turbine i, N

F_{s} Static friction force acting on the portion of a mooring line cable that is resting along the seabed, N

$\mathbf{F}_{\text{thrust},i}$ Aerodynamic thrust force vector acting on the rotor of turbine i, N

F_w Total weight of mooring line cable, N

H Horizontal tension along mooring line cable, N

H_{A} Horizontal tension at mooring line anchor, N

H_{F} Horizontal tension at mooring line fairlead, N

$H_{\text{F},2\to3}$ Horizontal tension that causes mooring line cable to fully lift off of the seabed, N

$H_{\text{F},i,j}$ Magnitude of the horizontal tension along mooring line j of turbine i, N

L Stretched mooring line cable length, m

$L_{\text{A}\to\text{F},i,j}$ Distance from anchor to fairlead for mooring line j of turbine i, m

$L_{\text{A}\to\text{G,neutral}}$ Distance from anchors to platform centers in their neutral positions, m

L_{c} Unstretched mooring line cable length, m

$L_{\text{c,min}}$ Shortest permissible mooring line cable length, m

$L_{\text{c,max}}$ Longest permissible mooring line cable length, m

$L_{\text{G}\to\text{F}}$ Distance from platform center to fairlead locations, m

L_{s} Length of cable resting along the seabed, m

$\mathscr{M}$ Set containing the indices of mooring line cables for each platform

N Number of turbines in the wind farm

P_{farm} Wind farm power output, W

$P_{\text{farm},\infty}$ Wind farm power output in the absence of the wake effect, W

P_i Power output of turbine i, W

p_P Tuning parameter for estimating turbine power coefficients

R	Radius of turbine rotors, m
$\mathbf{r}_{\mathrm{A}\rightarrow\mathrm{F},i,j}$	Position vector from anchor to fairlead for mooring line j of turbine i, m
$\mathbf{r}_{\mathrm{A},i,j}$	Position vector identifying location of the anchor of mooring line j of turbine i, m
$\mathbf{r}_{\mathrm{G}\rightarrow\mathrm{F},i,j}$	Position vector from platform center to fairlead for mooring line j of turbine i, m
$\mathbf{r_i}$	Position vector indicating the location of turbine i, m
s	Arbitrary arc length along a mooring line cable, m
T	Tension along mooring line cable, N
U_∞	Free stream wind speed, m/s
U_i	Effective incident wind speed on the rotor of turbine i, m/s
$\mathscr{U}_i$	Set containing the indices of all machines located upstream of turbine i
V	Vertical tension along a mooring line cable, N
V_A	Vertical tension at a mooring line anchor, N
V_F	Vertical tension at a mooring line fairlead, N
w	Weight per-unit-length of mooring line cables in ocean water, N/m
x	Axis aligned with the free stream wind direction
$\mathbf{x}$	Vector containing positions of all turbines along the x axis, m
x_F	Distance from an anchor to a fairlead in a catenary analysis, m
$x_{\mathrm{F},1\rightarrow2}$	Distance from an anchor to a fairlead that first leads to a catenary profile, m
$x_{\mathrm{F},2\rightarrow3}$	Distance from an anchor to a fairlead that fully lifts a cable off of the seabed, m
$x_{\mathrm{F,neutral}}$	Distance from anchors to fairleads while platforms are in their neutral positions, m
x_i	Position of turbine i along the x axis, m
x_s	Distance along a mooring cable portion resting on the seabed where the static friction force balances with the horizontal cable tension, m
y	Axis perpendicular to the free stream wind direction
y_i	Position of turbine i along the y axis, m
$\mathbf{y}$	Vector containing positions of all turbines along the y axis, m
z	Axis pointing upward from the ocean surface

z_{F}	Vertical location of mooring line fairleads above the seabed, m

Greek letters

α_{cable}	Modification factor used in mooring line cable length studies
γ	Vector containing the yaw angles of all turbines in the wind farm, deg
γ_i	Yaw angle of turbine i, rad
$\Delta\eta_{\mathrm{farm,rel}}$	Relative gain in wind farm efficiency, %
η_{farm}	Wind farm efficiency, %
η_P	Turbine generator efficiency
θ_{moor}	Rotation of the mooring system orientation relative to the baseline design, rad
μ_{s}	Coefficient of static friction between mooring line cables and the seabed
ρ	Density of air, kg/m^3

Acronyms

CFD	Computational fluid dynamics
FOWFSim	Floating Offshore Wind Farm Simulator
FLORIS	Flow Redirection in Steady-state
LES	Large eddy simulation
SOWFA	Simulator for Wind Farm Applications
YITuR	Yaw and induction-based turbine repositioning

Chapter 4

A	Generalized symbol to signify area-based integration, m^2
$A_{\mathrm{a},i,j}$	Added mass reference area of submerged component j of turbine i, m^2
$A_{\mathrm{d},i,j}$	Drag reference area of submerged component j of turbine i, m^2
$a_i(t)$	Axial induction factor of turbine i
A_i	Swept area of the rotor of turbine i, m^2
$C_{\mathrm{a},i,j}$	Added mass coefficient of submerged component j of turbine i
$C_{\mathrm{d},i,j}$	Drag coefficient of submerged component j of turbine i
$C_{\mathrm{p},i}(t)$	Power coefficient of turbine i

$C_{\mathrm{t},i}(t)$	Thrust coefficient of turbine i
D_i	Diameter of the rotor of turbine i, m
$\mathscr{D}_i$	Set containing the indices of all submerged components of turbine i
$D_{\mathrm{w},i}(\hat{x}_i,t)$	Diameter of the wake produced by turbine i over space and time, m
$D_{\mathrm{w},i,p}(t)$	Wake diameter at point p of the wake produced by turbine i, m
$D_{\mathrm{w},q\rightarrow i}(t)$	Diameter of the wake produced by turbine q at the location of the rotor of turbine i, m
$f(\cdot)$	Nonlinear continuous-time state-space model of the wind farm
$\mathscr{F}$	Set containing the indices of all turbines in the wind farm
$\mathbf{F}_{\mathrm{a},i}(t)$	Aerodynamic force vector acting on the rotor of turbine i, N
$\mathbf{F}_{\mathrm{h},i}(t)$	Hydrodynamic force vector acting on the platform of turbine i, N
$\mathbf{F}_i(t)$	Total force vector acting on turbine i, N
$\mathbf{F}_{\mathrm{m},i}(t)$	Total mooring line force vector acting on the platform of turbine i, N
$\mathbf{F}_{\mathrm{m},i,k}(t)$	Force vector of mooring line k of turbine i, N
$H_{\mathrm{F},i,k}(t)$	Horizontal tension along mooring line k of turbine i, N
k_t	Temporal wake expansion constant, m/s
k_x	Spatial wake expansion constant, m/m
$\mathbf{L}_i(\hat{x}_i,t)$	Linear momentum vector per unit length of the wake produced by turbine i over space and time, kg/s
$m_{\mathrm{a},i}$	Effective added hydrodynamic mass resulting from the acceleration of the platform of turbine i, kg
m_i	Total mass of turbine i including the floating platform mass, kg
$\mathscr{M}_i$	Set containing the indices of the mooring lines attached to the platform of turbine i
N	Number of turbines in the wind farm
$\mathbf{n}_\infty(t)$	Unit vector aligned with the free stream wind direction
$N_{\mathrm{h},i}$	Total number of submerged components of turbine i
$\mathbf{n}_i(t)$	Unit vector pointing outward from and normal to the rotor of turbine i
$N_{\mathrm{m},i}$	Total number of mooring lines attached to the platform of turbine i
$N_{\mathrm{p},i}$	Number of points used to spatially discretize the wake produced by turbine i
$P_{\mathrm{farm}}(t)$	Power output of the wind farm, W
$P_i(t)$	Power output of turbine i, W

r Arbitrary radius along the radial axis used to define Gaussian wake velocity profiles, m

$\mathbf{r}_{\mathrm{A},i,k}$ Position vector identifying the location of the anchor of mooring line k of turbine i, m

$\mathbf{r}_{\mathrm{F/A},i,k}(t)$ Position vector from anchor to fairlead for mooring line k of turbine i, m

$\mathbf{r}_{\mathrm{F/G},i,k}$ Position vector from platform center to fairlead for mooring line k of turbine i, m

$\mathbf{r}_i(t)$ Position vector identifying the location of turbine i, m

$U_\infty(t)$ Free stream wind speed along the $\hat{x}$ axis, m/s

$U_{\infty,\mathrm{ref}}$ Reference free stream wind speed used to convert between spatial and temporal wake expansion constants, m/s

$\mathbf{u}_{\mathrm{farm}}(t)$ Vector containing the inputs of all turbines in the wind farm

$\mathbf{u}_i(t)$ Vector containing the inputs of turbine i

$U_i(t)$ Component of $\mathbf{V}_i(t)$ aligned with the $\hat{x}$ axis, m/s

$\mathscr{U}_i$ Set containing the indices of all machines that are located upstream of turbine i

$u_{\mathrm{w},i}(\hat{x}_i,t)$ Average speed of the wake produced by turbine i in the $\hat{x}$ direction over space and time, m/s

$u_{\mathrm{w},i,p}(t)$ Average wake speed at point p of the wake produced by turbine i in the $\hat{x}$ direction, m/s

$V_\infty(t)$ Free stream wind speed along the $\hat{y}$ axis, m/s

$\mathbf{V}_\infty(t)$ Free stream wind velocity vector, m/s

$\mathbf{v}_i(t)$ Velocity vector of turbine i, m/s

$V_i(t)$ Component of $\mathbf{V}_i(t)$ aligned with the $\hat{y}$ axis, m/s

$\mathbf{V}_i(t)$ Effective wind velocity vector that is incident on the rotor of turbine i measured in the global stationary frame, m/s

$\mathbf{V}_{\mathrm{rel},i}(t)$ Effective wind velocity vector that is incident on the rotor of turbine i measured by an observer who is fixed to turbine i, m/s

$v_{\mathrm{w},i}(\hat{x}_i,t)$ Average speed of the wake produced by turbine i in the $\hat{y}$ direction over space and time, m/s

$\mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)$ Average velocity vector of the wake produced by turbine i over space and time, m/s

$\mathbf{v}_{\mathrm{w,init},i}(t)$	Average velocity vector of the wake produced by turbine i immediately downstream of its rotor, m/s
$v_{\mathrm{w},i,p}(t)$	Average wake speed at point p of the wake produced by turbine i in the $\hat{y}$ direction, m/s
$\check{\mathbf{v}}_{\mathrm{w},q\rightarrow i}(r,t)$	Gaussian velocity distribution of the wake produced by turbine q at the location of the rotor of turbine i, m/s
$\overline{\mathbf{v}}_{\mathrm{w},q\rightarrow i}(t)$	Effective velocity of the wake produced by turbine q that is incident on the rotor of turbine i, m/s
$v_{x,i}(t)$	Velocity of turbine i along the $\hat{x}$ axis, m
$v_{y,i}(t)$	Velocity of turbine i along the $\hat{y}$ axis, m
$\mathbf{w}(t)$	Ocean current velocity vector, m/s
$\hat{x}$	Global axis aligned with the predominant free stream wind direction
$\mathbf{x}(t)$	Vector containing the states of all turbines in the wind farm
$\mathbf{x}_{\mathrm{farm}}(t)$	Vector containing the states of all wakes and turbines in the wind farm
$x_i(t)$	Position of turbine i along the $\hat{x}$ axis, m
$\hat{x}_i$	Local axis aligned with the predominant free stream wind direction and fixed to turbine i
$\mathbf{x}_{\mathrm{w}}(t)$	Vector containing the states of all wakes in the wind farm
$\mathbf{x}_{\mathrm{w},i}(t)$	Vector containing the states of the wake produced by turbine i
$\mathbf{x}_{\mathrm{w},i,p}(t)$	Vector containing the states of point p of the wake produced by turbine i
$\hat{y}$	Global axis perpendicular to the predominant free stream wind direction
$y_i(t)$	Position of turbine i along the $\hat{y}$ axis, m
$\hat{y}_i$	Local axis perpendicular to the predominant free stream wind direction and fixed to turbine i
$y_{\mathrm{w},i}(\hat{x}_i,t)$	Centerline position of the wake produced by turbine i in the $\hat{y}$ direction over space and time, m
$y_{\mathrm{w},i,p}(t)$	Centerline position at point p of the wake produced by turbine i in the $\hat{y}$ direction, m

Greek letters

$\gamma_i(t)$	Yaw angle of the nacelle of turbine i measured relative to the $\hat{x}$ axis, rad
$\gamma_{\mathrm{rel},i}(t)$	Yaw angle of the nacelle of turbine i relative to $\mathbf{V}_{\mathrm{rel},i}(t)$, rad
$\theta_i(t)$	Angle of the vector $\mathbf{V}_{\mathrm{rel},i}(t)$ measured relative to the $\hat{x}$ axis, rad
$\xi_{\mathrm{w,init},i}(t)$	Skew angle of the wake produced by turbine i immediately past its rotor used in momentum conservation calculations, rad
ρ_{a}	Density of air, kg/m^3
ρ_{w}	Density of ocean water, kg/m^3
σ	Standard deviation of Gaussian wake velocity profiles, m
$\chi_i(t)$	Skew angle of the wake produced by turbine i immediately past its rotor used in vortex theory calculations, rad

Acronyms

FLORIDyn	Flow Redirection and Induction Dynamics
FLORIS	Flow Redirection in Steady-state
FOWFSim	Floating Offshore Wind Farm Simulator
FOWFSim-Dyn	Dynamic Floating Offshore Wind Farm Simulator
LES	Large eddy simulation
NREL	National Renewable Energy Laboratory
RMSE	Root-mean-square-error
WFSim	Wind Farm Simulator

Chapter 5

$\mathbf{f}_i(\cdot)$	Discrete-time state-space model of agent i
H	Number of time-steps along the prediction horizon
$\mathscr{I}$	Set containing the indices of all agents
$J(\cdot)$	Global stage cost function
$J_i(\cdot)$	Neighborhood-cooperative stage cost function of agent i
k	Time-step number along the prediction horizon
$\ell_i(\cdot)$	Local stage cost function of agent i
m_i	Number of inputs of agent i
N	Number of agents

$\mathscr{N}_{-i}$ Set containing the indices of all agents that influence $\ell_i(\cdot)$

$\mathscr{N}_{-i|J}$ Set containing the indices of all agents that influence $J_i(\cdot)$

$\mathscr{N}_{+i}$ Set containing the indices of all agents whose local cost functions are influenced by the states and inputs of agent i

$\mathscr{N}_{+i|J}$ Set containing the indices of all agents whose neighborhood-cooperative cost functions are influenced by the states and inputs of agent i

n_i Number of states of agent i

N_q Total number of social hierarchy levels

N_s Total number of possible social hierarchy levels that will resolve conflict

p Iteration number in Algorithm 1

P Probability that all conflicts will be resolved within a single iteration

q Current social hierarchy level being implemented by Algorithm 1

q_i Social hierarchy level of agent i

$\mathscr{U}_{-i}$ Set containing all permissible input vectors of neighboring agents in the set $\mathscr{N}_{-i}$

$\mathbf{u}^k_{-i|J}$ Vector containing the inputs of all neighboring agents in the set $\mathscr{N}_{-i|J}$ at time-step k along the prediction horizon

$\hat{\mathbf{u}}^k_{-i|J}$ Array containing the input vectors of all neighboring agents in the set $\mathscr{N}_{-i|J}$ at time-step k along the prediction horizon that agent i assumes while computing optimal decisions

$\breve{\mathbf{u}}^k_{-i|J}$ Array contain optimal input vectors of all neighboring agents in the set $\mathscr{N}_{-i|J}$ at time-step k along the prediction horizon

$\hat{\mathbf{u}}^s_{-i|J}$ Array containing the steady input vectors of all neighboring agents in the set $\mathscr{N}_{-i|J}$ that agent i assumes while computing optimal decisions

$\breve{\mathbf{u}}^s_{-i|J}$ Array containing optimal steady input vectors of all neighboring agents in the set $\mathscr{N}_{-i|J}$

$\mathbf{u}_i$ Current input vector of agent i

$\overline{\mathbf{u}}_i$ Input trajectory of agent i along the prediction horizon

$\mathscr{U}_i$ Set containing all permissible input vectors for agent i

$\overline{\mathscr{U}}_i$ Set containing all permissible input vector trajectories for agent i

$\overline{\mathbf{u}}^*_i$ Optimal input vector trajectory of agent i along the prediction horizon

$\mathbf{u}_i^k$	Candidate input vector of agent i at time-step k along the prediction horizon
$\breve{\mathbf{u}}_i^k$	Optimal input vector of agent i at time-step k along the prediction horizon
$\mathbf{u}_i^s$	Candidate steady input vector of agent i
$\breve{\mathbf{u}}_i^s$	Optimal steady input vector of agent i
$\tilde{\mathbf{u}}_j$	The input vector trajectory along the prediction horizon that agent i assumes for agent j
$\hat{\mathbf{u}}_j^s$	The steady input vector that agent i assumes for agent j
$V(\cdot)$	Global stage cost function value summed over the prediction horizon
$\hat{V}_i$	Naive stage cost function value of agent i summed over the prediction horizon
$\breve{V}_i$	Informed stage cost function value of agent i summed over the prediction horizon
$\hat{V}_i^s$	Naive stage cost function value of agent i at steady-state
$\breve{V}_i^s$	Informed cost function value of agent i at steady-state
$\mathscr{X}_{-i}$	Set containing all permissible state vectors of neighboring agents in the set $\mathscr{N}_{-i}$
$\mathbf{x}_{-i\|J}^k$	Vector containing the states of all neighboring agents in the set $\mathscr{N}_{-i\|J}$ at time-step k along the prediction horizon
$\hat{\mathbf{x}}_{-i\|J}^k$	Array containing the state vectors of all neighboring agents in the set $\mathscr{N}_{-i\|J}$ at time-step k along the prediction horizon that agent i assumes while computing optimal decisions
$\breve{\mathbf{x}}_{-i\|J}^k$	Array contain optimal state vectors of all neighboring agents in the set $\mathscr{N}_{-i\|J}$ at time-step k along the prediction horizon
$\hat{\mathbf{x}}_{-i\|J}^s$	Array containing the steady state vectors of all neighboring agents in the set $\mathscr{N}_{-i\|J}$ that agent i assumes while computing optimal decisions
$\breve{\mathbf{x}}_{-i\|J}^s$	Array containing optimal steady state vectors of all neighboring agents in the set $\mathscr{N}_{-i\|J}$
$\mathbf{x}_i$	Current state vector of agent i
$\mathscr{X}_i$	Set containing all permissible state vectors for agent i
$\overline{\mathscr{X}}_i$	Set containing all permissible state vector trajectories for agent i
$\overline{\mathbf{x}}_i^*$	Optimal state vector trajectory of agent i along the prediction horizon

$\mathbf{x}_i^{+}$	State vector of agent i at the subsequent sampling time
$\mathscr{X}_i^{0\rightarrow s}$	Set containing all initial state vectors that may be feasibly steered to a stationary point for agent i
$\mathbf{x}_i^k$	Candidate state vector of agent i at time-step k along the prediction horizon
$\breve{\mathbf{x}}_i^k$	Optimal state vector of agent i at time-step k along the prediction horizon
$\mathbf{x}_i^s$	Candidate steady state vector of agent i
$\breve{\mathbf{x}}_i^s$	Optimal steady state vector of agent i
$\mathscr{X}_i^s$	Set containing all feasible steady state vectors
$\tilde{\mathbf{x}}_j$	The state vector trajectory along the prediction horizon that agent i assumes for agent j
$\hat{\mathbf{x}}_j^s$	The steady state vector that agent i assumes for agent j
$\mathscr{Z}_i^{0\rightarrow s}$	Set containing all initial state vectors, input vector trajectories, and steady state vectors that satisfy system dynamics and constraints for agent i
$\mathscr{Z}_i^s$	Set containing all feasible stationary points for agent i

Acronyms

DMPC	Distributed model predictive control
DEMPC	Distributed economic model predictive control
EMPC	Economic model predictive control
MPC	Model predictive control

Chapter 6

a_i	Axial induction factor of turbine i
$a_{i,k}$	Axial induction factor of turbine i at time-step k along the prediction horizon
$a_{i,s}$	Axial induction factor of turbine i at the candidate stationary point
$A_{\mathrm{OL},i\rightarrow j}(\cdot)$	Function that outputs the normalized overlap area between the rotors of two adjacent turbines

$\mathbf{f}_i(\cdot)$	Discrete-time state-space model of turbine i that is identified using feed-forward neural networks
H	Number of time-steps along the prediction horizon
k	Time-step number along the prediction horizon
$\mathscr{N}_i$	Set containing the indices of all machines that are adjacent to turbine i
$\mathbf{Q}$	Input penalty weighting matrix
$\bar{\mathbf{u}}_i$	Input vector trajectory of turbine i along the prediction horizon
$\mathbf{u}_{i,k}$	Input vector of turbine i at time-step k along the prediction horizon
$\mathbf{u}_{i,s}$	Input vector of turbine i at the candidate stationary point
$\mathbf{V}_\infty$	Free stream wind velocity vector, m/s
$\overline{\mathbf{V}}_\infty$	Mean free stream wind velocity vector, m/s
$v_{x,i}$	Velocity of turbine i along the $\hat{x}$ axis, m/s
$v_{x,i,k}$	Velocity of turbine i along the $\hat{x}$ axis at time-step k along the prediction horizon, m/s
$v_{y,i}$	Velocity of turbine i along the $\hat{y}$ axis, m/s
$v_{y,i,k}$	Velocity of turbine i along the $\hat{y}$ axis at time-step k along the prediction horizon, m/s
$\hat{x}$	Axis aligned with the predominant free stream wind direction
x_i	Position of turbine i along the $\hat{x}$ axis, m
$\bar{\mathbf{x}}_i$	State vector trajectory of turbine i along the prediction horizon
$x_{i,k}$	Position of turbine i along the $\hat{x}$ axis at time-step k along the prediction horizon, m
$\mathbf{x}_{i,k}$	State vector of turbine i at time-step k along the prediction horizon
$\mathbf{x}_{i,s}$	State vector of turbine i at the candidate stationary point
$\hat{y}$	Axis perpendicular to the predominant free stream wind direction
y_i	Position of turbine i along the $\hat{y}$ axis, m
$y_{i,k}$	Position of turbine i along the $\hat{y}$ axis at time-step k along the prediction horizon, m

Greek letters

γ_i	Yaw angle of the nacelle of turbine i measured relative to the $\hat{x}$ axis, rad
$\gamma_{i,k}$	Yaw angle of turbine i at time-step k along the prediction horizon, rad

$\gamma_{i,s}$	Yaw angle of turbine i at the candidate stationary point, rad
$\Delta\mathbf{u}_{i,k}$	Change in the input vector of turbine i relative to greedy reference values
σ_{∞}	Parameter used to fluctuate the free stream wind velocity vector relative to its mean value

Acronyms

DEMPC	Distributed economic model predictive control
EMPC	Economic model predictive control
FOWFSim-Dyn	Dynamic Floating Offshore Wind Farm Simulator
MPC	Model predictive control
NREL	National Renewable Energy Laboratory
RMSE	Root-mean-square-error
YITuR	Yaw and induction-based turbine repositioning

Chapter 7

Acronyms

DEMPC	Distributed economic model predictive control
FOWFSim	Floating Offshore Wind Farm Simulator
FOWFSim-Dyn	Dynamic Floating Offshore Wind Farm Simulator
LES	Large eddy simulation
YITuR	Yaw and induction-based turbine repositioning

Acknowledgments

I am first grateful to my research supervisor Prof. Ryozo Nagamune for the opportunities and guidance that he has offered to me over the past three years. Second, I would like to extend my appreciation to the supervisory committee members Prof. Anasavarapu Srikantha Phani, Prof. Rajeev Jaiman, and Prof. Bhushan Gopaluni, the university examiners Prof. Guy Dumont and Prof. Martin Ordonez, and the external examiner Prof. Jan-Willem van Wingerden for offering their time, support, and useful feedback, which has led to substantive improvements to this thesis.

I am also indebted to my close friends Moe Kabbara, Corey Yeadon, and Patrick MacInnis, and to my lab mates at the Control Engineering Laboratory Pouya, Sara, Eduardo, Alejandrina, Yue, Christopher, Marcin, Jacob, Nafise, Jihoon, and Naho for their support and encouragement.

Finally, I am thankful for the financial support provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Province of British Columbia.

Chapter 1

Introduction

This thesis presents various aspects of research aimed at maximizing power production in floating offshore wind farms. The current chapter introduces the reader to the main problem that causes inefficiency in wind farms, and then describes the solution approach of interest for mitigating this problem. Subsequent chapters then describe the different steps taken to implement and assess the potential, feasibility, and challenges of the proposed solution. Each chapter is authored such that it may be read independently of the rest of the text. The current chapter therefore also provides an outline of the thesis while discussing the purpose and contributions of each chapter in the context of the overarching objective.

1.1 The wake effect

Wind turbines are clustered into wind farms to maximize energy extraction from regions with high-resource winds. One consequence of this siting approach is that machines that are in close proximity interact aerodynamically and experience diminished performance. Referring to Figure 1.1, viscous interaction along the blades of a wind turbine generates a region of airflow downstream of its rotor that is characterized by reduced wind speeds; this region is referred to as a *wake*. Any wind turbines located downstream and aligned with the wake produced by an upstream machine therefore produce less power than they would in the absence of the wake. This phenomenon is termed the *wake effect*, and may lead to reductions

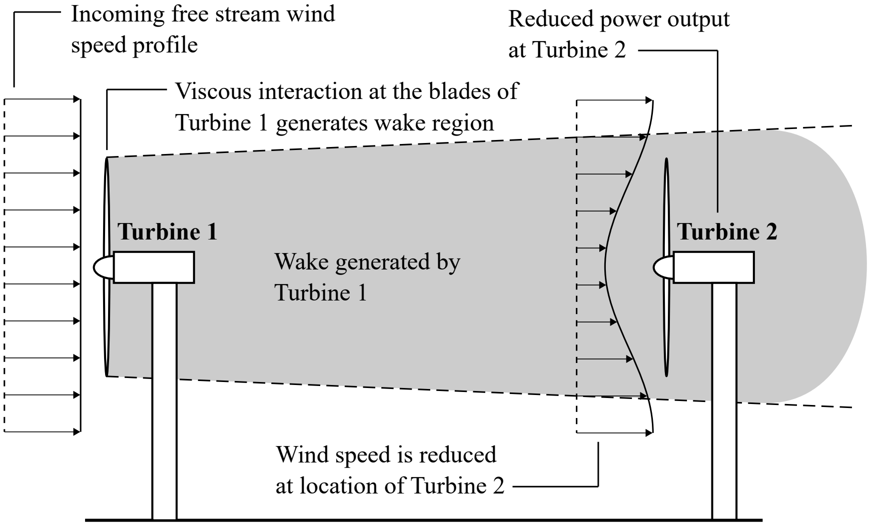

Incoming free stream wind speed profile
Reduced power output at Turbine 2
Viscous interaction at the blades of Turbine 1 generates wake region
Turbine 1
Turbine 2
Wake generated by Turbine 1
Wind speed is reduced at location of Turbine 2

Figure 1.1: Schematic demonstrating the phenomenon of aerodynamic coupling between wind turbines that are aligned with the wind.

of up to 60 % in the power output of downstream turbines [141], while decreasing annual energy production by 20 to 30 % [30].

1.2 Yaw and induction-based turbine repositioning (YITuR)

One solution to mitigating the wake effect is *wind farm control*, which simply describes the process of operating individual wind turbines in a manner that raises power production from the collective. The primary objective of this thesis is to investigate the potential of a specific wind farm control approach that is relevant to floating offshore wind farms and is referred to as *yaw and induction-based turbine repositioning* (YITuR).

As shown in Figure 1.2, by varying the nacelle yaw angle of a wind turbine, the direction of the aerodynamic thrust force acting on its rotor may be altered as desired. Further, by adjusting other parameters, such as the pitch angles of the blades and shaft rotation speed, the magnitude of the aerodynamic thrust force may be manipulated as well. Floating offshore wind turbines that are anchored to the seabed using slack mooring lines possess some finite range of mobility along the

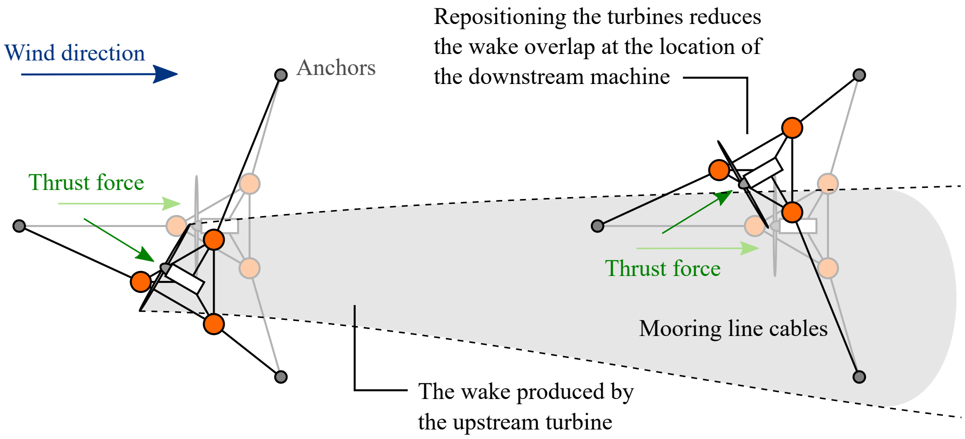

Wind direction
Anchors
Repositioning the turbines reduces the wake overlap at the location of the downstream machine
Thrust force
Thrust force
Mooring line cables
The wake produced by the upstream turbine

Figure 1.2: Schematic demonstrating the use of yaw and induction based turbine repositioning (YITuR) to minimize wake overlap in floating offshore wind farms.

ocean surface. Variations of thrust forces thus permit relocation of floating platforms in real-time. The permissible range of this motion is determined by the equilibrium established between the aerodynamic thrust forces and restoring mooring line tensions acting on each platform. Given this repositioning mechanism, individual floating wind turbines may be relocated in real-time to minimize the overlap between the rotors of downstream machines and the wakes produced by upstream counterparts.

1.3 Research question and thesis outline

The research question addressed in this thesis may be stated as follows: *Does the YITuR wind farm control technique offer sufficient economic incentive to be implemented in practice? And if so, what challenges must be overcome along the road to industrial implementation?*

This question is answered through a series of steps. The efforts and results associated with each step are outlined in subsequent chapters. An overview of the role of each chapter is provided in Figure 1.3. The chapters are listed in the order that is most suitable for comprehension; however, each chapter is written without dependence on content or knowledge contained in other parts of the thesis.

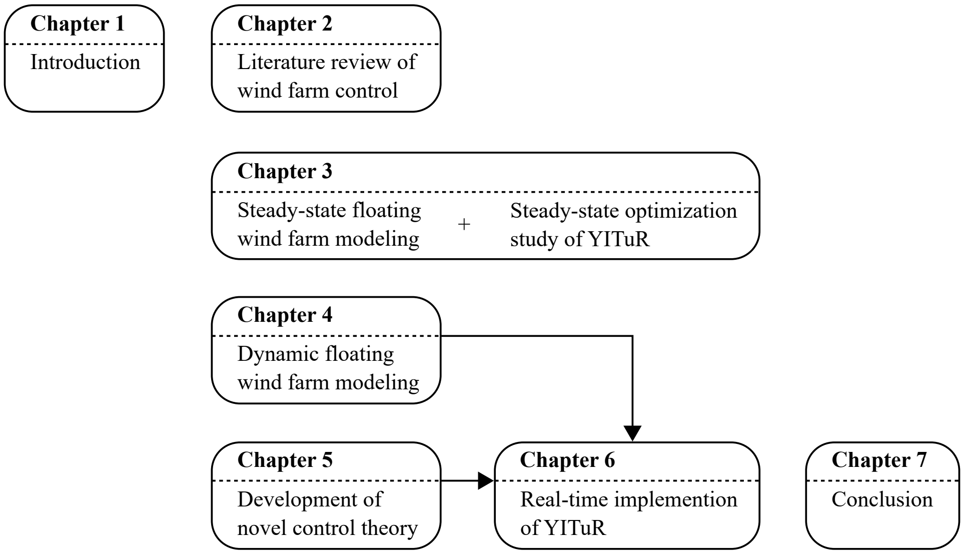

Chapter 1
Introduction
Chapter 2
Literature review of wind farm control
Chapter 3
Steady-state floating wind farm modeling
+
Steady-state optimization study of YITuR
Chapter 4
Dynamic floating wind farm modeling
Chapter 5
Development of novel control theory
Chapter 6
Real-time implemention of YITuR
Chapter 7
Conclusion

Figure 1.3: Layout of the thesis and the role of each chapter.

The reader may therefore skip specific chapters that are not of interest without relinquishing any information that is vital for their apprehension. The purpose and contributions of each chapter in the context of the objective of the thesis are listed below:

- **Chapter 2:** The second chapter provides a comprehensive review of wind farm control literature. The current status of the wind energy sector is first discussed, followed by detailed descriptions of the wake effect and various wind farm control techniques. Next, vital information from existing studies, such as modeling methodologies and reported wind farm efficiency gains, is accumulated in review tables while the individual studies are summarized. Finally, the collected data are used to offer insight into the gaps in existing research practices. This work has been published in the *Journal of Wind Engineering and Industrial Aerodynamics* [102].

- **Chapter 3:** Prior to dedicating effort to the development of dynamic models and controllers, it is necessary to establish, using a relatively simplified analysis, whether turbine repositioning is a beneficial wind farm control approach. The third chapter thus presents a steady-state assessment of the potential benefits of YITuR. This step involves the development of the first steady-state floating offshore wind farm model that couples platform displacement with wake aerodynamics. This simulator is name the *Floating Offshore Wind Farm Simulator* (FOWFSim). Then, optimization studies are carried out for different wind settings and mooring system design parameters to assess the efficiency gains that result from YITuR. The analysis demonstrates that turbine repositioning may lead to significant gains in power production under specific conditions, while also establishing its limitations. This work has been published in *Ocean Engineering* [103].

- **Chapter 4:** In order to assess the performance of a YITuR controller in real-time, the fourth chapter extends FOWFSim to capture floating wind farm dynamics; hence the name FOWFSim-Dyn. To this end, the first parametric dynamic floating offshore wind farm model is developed. Parametric wake models offer computational expediency by utilizing assumptions concerning flow behaviour which simplify the governing equations of fluid flow. The trade-off, however, is reduced accuracy in comparison to computational fluid dynamics solvers. FOWFSim-Dyn predicts floating wind turbine motion along the ocean surface coupled with wake transport. This simulator is also the first parametric tool to capture the effects of time-varying wind speed and direction on wake behaviour. The model is validated under steady-state conditions using published experimental results.

- **Chapter 5:** The fifth chapter develops novel theory in the field of distributed economic model predictive control[1] (DEMPC). The turbine repositioning optimal control problem is non-convex since there exist multiple trajectories that individual turbines may follow to reduce wake overlap. One major gap among existing DEMPC algorithms is that, due to non-convexity, they cannot offer a guarantee that the optimal decisions made by the individual turbines will converge. In other words, they offer no guarantee that individual floating turbines will make decisions that ultimately lead to a rise in total power production. As a result, the development of novel DEMPC theory capable of providing such a guarantee is necessary for delivering a generalized

[1] The reasons for selecting DEMPC for turbine repositioning are clarified in Chapter 6.

algorithm that implements YITuR.

- **Chapter 6:** The sixth chapter pieces together the contributions of Chapter 4 and Chapter 5 by using the novel DEMPC algorithm to implement YITuR in real-time in the simulated environment of FOWFSim-Dyn. The DEMPC algorithm maximizes wind farm power production by solving dynamic optimization problems at each time-step. In order to increase the speed of decision-making, feed-forward neural networks are used to estimate the dynamic behaviour of floating wind turbine platforms during the optimization process of DEMPC. These neural networks are tuned using data generated by FOWFSim-Dyn. Therefore, an additional contribution of this chapter is the first use of neural networks to predict floating turbine dynamics. Finally, the performance of the DEMPC algorithm and the benefits of real-time YITuR are assessed for different wind farm sizes and time-varying wind conditions.

- **Chapter 7:** The final chapter concludes the thesis by summarizing the major findings from each preceding research step. Additionally, recommendations regarding future research steps are offered.

Chapter 2

A Quantitative Review of Wind Farm Control with the Objective of Wind Farm Power Maximization

This chapter presents a review of control strategies for maximizing power production within wind farms. Discussions focus on three notable concepts; power derating, yaw-based wake redirection, and turbine repositioning. Existing works that have examined the potential of these concepts via optimization studies, numerical simulation, experimentation, as well as those that have developed and evaluated control algorithms, are reviewed thoroughly and quantitatively. Criteria for this review process include the evaluation methods employed, simulated wind conditions, controller properties such as model dependency and communication architecture, and the resulting relative rise in wind farm efficiency. The data collected from existing literature is then utilized to draw conclusions regarding the influence of each of these criteria on the potential and performance of wind farm controllers. Appropriate recommendations for future modeling and controller design research are then offered based on these conclusions.

2.1 Introduction

This introductory section summarizes the status of the wind energy industry and outlines the phenomenon of aerodynamic interaction among wind turbines and its effect on wind farm efficiency. Possible solutions for mitigating the wake effect are discussed prior to tackling the main subject of this review, which is wind farm control. A brief overview of existing literature surveys on this subject is also presented as build-up to the objectives of the current chapter.

2.1.1 Status of wind energy

The environmental and economic significance of wind power in the 21st century is not only palpable to members of scientific and green-tech communities, but is now mainstream knowledge. What is more, the initial goals and ambitions that have propelled the wind energy industry since the start of the century are gradually being realized. For instance, the primary incentive for investment and technological progress in the wind energy sector has been to reduce green house gas (GHG) emissions while continuing to meet global electricity demands [13]. From 2000 to 2016, the global share of annual electricity production from wind power increased from 0.20 to 3.84 %[1] [87, 167]. Based on electricity production methods in the United States, the recent end of this range corresponds to 712 million metric tons of avoided CO_2 emissions [57], and accounts for 1.97 % of total global CO_2 emissions from 2015 [143].

In addition to mitigating environmental concerns, investment in wind energy offers a multitude of social, political, and economic benefits; the diversification of national energy portfolios, which strengthens energy security and prevents global conflicts over natural resources; the reduction of poverty through greater accessibility to low-cost energy; and innovation and job creation [13]. On this final point, the global employment count in the wind energy sector rose from 0.75 to 1.16 million between 2012 and 2016 [89]. The global wind energy sector therefore employed more individuals than the traditional power generation industries (*i.e.* coal, oil, and gas) of the United States, which collectively employed 1.1 million individuals in

[1] Data on global electricity production from wind were obtained from IRENA [87]. Data on overall global electricity production were obtained from Enerdata [167].

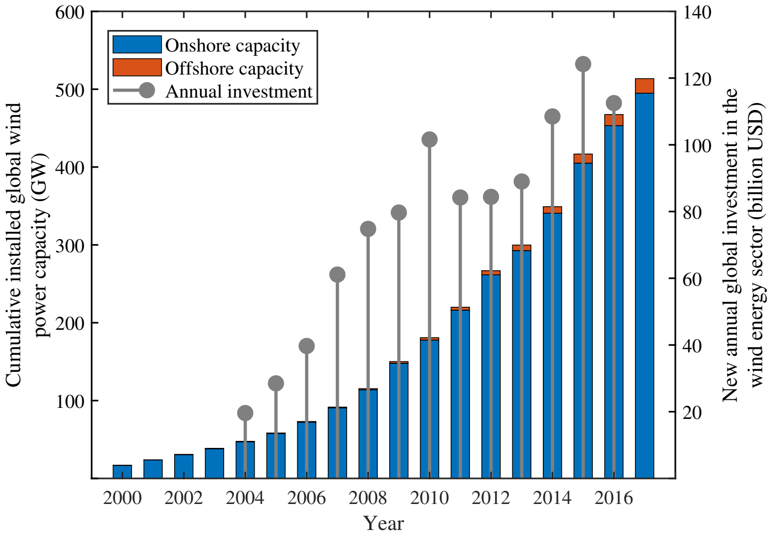


Figure 2.1: Trends in installed global wind power capacity and global investment in the wind energy sector since the start of the century. Data have been obtained from the International Renewable Energy Agency (IRENA) [87], [90].

2016 [173].

These environmental and economic benefits, along with government mandates and support, have generated the substantial levels of growth and investment in the wind energy sector that are shown in Figure 2.1. Cumulative installed global wind power capacity has increased from 16.9 to 513.5 GW from 2000 to 2017 [87]. Meanwhile, new annual global investments in the wind energy sector have risen from 19.6 to 112.5 billion USD from 2004 to 2016 [90]. These trends are indicative of the considerable efforts being made to mature and proliferate wind power technology on a global scale.

Figure 2.1 also reveals growing interest in locating wind farms offshore, for which the motivations are plentiful; environmental impact is limited since the construction of roads for transport is unnecessary; visual and auditory concerns are mitigated as turbines are located far from residential areas; obtaining land permits is less problematic; and most major cities are located close to shore, which permits shorter transmission lines [121]. Offshore wind is also stronger and less turbulent due to the absence of obstructions and uneven terrain [8]. Consequently, offshore wind turbines accumulate less fatigue damage and deliver more electricity relative to their onshore counterparts. On this final point, the global average capacity factor (the ratio of energy produced to the maximum producible amount over a given period) for offshore wind power was 30 % greater than that of onshore wind power in 2017 [90].

Despite these benefits, offshore wind power lacks competitiveness against onshore wind and fossil fuel-based alternatives due to its infancy and harsh operating conditions [30]. Investment trends and the levelized cost of energy (LCOE) for onshore and offshore wind power are compared in Figure 2.2. In 2017, the global average investment cost for offshore wind power per kW of installed capacity was 4,239 USD/kW, while that for onshore wind power was 65.2 % lower at 1,477 USD/kW [88]. Likewise, the global average LCOE for offshore wind power at this time was 0.14 USD/kWh; the LCOE for onshore wind power was 57.1 % lower at 0.06 USD/kWh [88], and that of coal-fired power plants ranged from 0.0764 to 0.097 USD/kWh in 2015 [174]. Increased competitiveness is therefore imperative for the proliferation of offshore wind power, and it is a priority within the wind energy research community [24, 161].

2.1.2 Increasing wind farm efficiency

One approach to raising the competitiveness of a wind farm is to mitigate efficiency losses resulting from aerodynamic coupling between individual wind turbines [24]. The concept of aerodynamic coupling is depicted in Figure 2.3, which shows two wind turbines aligned with the free stream wind. Viscous interaction between the blades of turbine 1 and the incoming wind generates a downstream region of airflow that is characterized by low velocity and increased turbulence intensity [115]. This downstream region is referred to as a *wake*. Turbine 2, which is aligned with the wake that is generated by turbine 1, therefore produces less power and accumulates greater fatigue damage over time [115]. The overall process described in Figure 2.3 is commonly referred to as the *wake effect*, and may reduce power production from individual downstream turbines by up to 60 %, while diminishing power production from an entire wind farm by as much as 54 % [141]. Annu-

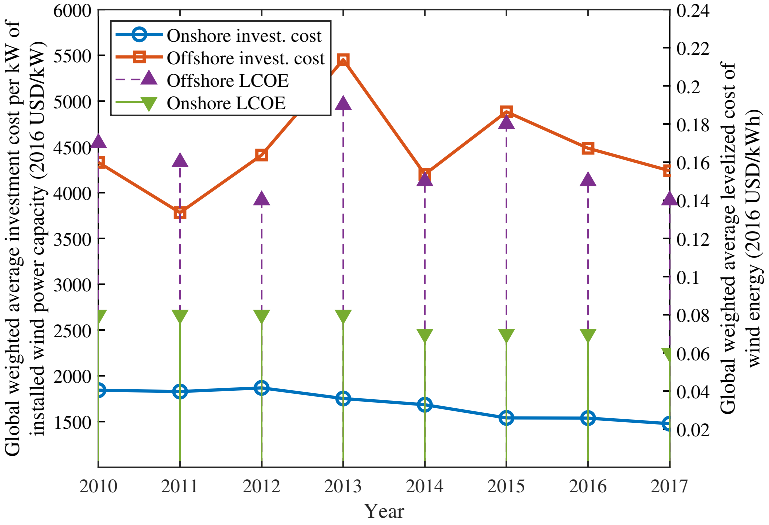


Figure 2.2: Trends in levelized energy and investments costs for onshore and offshore wind power since 2010. Data have been obtained from IRENA [88].

ally, the wake effect may result in cumulative revenue losses ranging from 20 to 30% [30].

Thus far, three methods have either been proposed or employed to mitigate losses resulting from the wake effect. The first approach involves siting wind turbines such that there exists appropriate spacing between two adjacent machines in the predominant wind direction [30]. Large inter-turbine spacing permits sufficient distance over which the process of turbulent mixing re-energizes wakes and reduces velocity deficits [115]. Although this method has been widely employed within the wind energy industry, it conflicts with the fundamental rationale behind wind farms; that is, to cluster wind turbines in close proximity to minimize capital and operational expenses [121]. Furthermore, the 20 to 30 % annual revenue losses alluded to earlier are observed with current industry-standard inter-turbine spacings of seven to ten rotor diameters in the predominant wind direction [30]. There is therefore growing interest in alternative methods that will further mitigate such losses.

The second solution for increasing wind farm power production is *layout opti-*

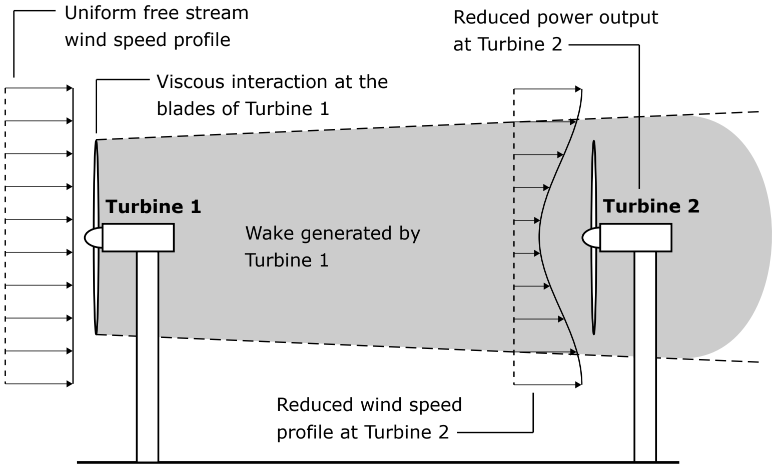


Figure 2.3: The phenomenon of aerodynamic coupling between two wind turbines aligned with the free stream wind.

mization. Engineering wake models are used to estimate wind farm power production as a function of historical wind velocity data spanning the entire year [161]. Optimization algorithms are then used to determine ideal turbine siting such that wind farm annual energy production is maximized [161]. Despite the large body of academic work that has been dedicated to layout optimization, the largest operational offshore wind farms possess gridded layouts with appropriate inter-turbine spacing in the predominant wind direction[2]. The absence of optimized wind farm layouts within the industry may be due to the novelty of this field of research[3], the low fidelity of engineering wake models [24], or a wind farm developer's consideration of additional factors when designing wind farms, which may include electrical connectivity costs, maintenance procedures, and local terrain properties [30].

The third strategy for alleviating revenue losses resulting from the wake effect, which is the focus of the current chapter, is *wind farm control*. In the broadest sense, a wind farm controller uses the available degrees of freedom of individual wind turbines to manipulate the wind field such that a wind farm level objective is achieved [24]. This wind farm level objective may be power maximization, or

[2] The reader may investigate the London Array, Greater Gabbard, Bard 1, Anholt, Walney, Thorntonbank, Sheringham Shoal, Thanet, Centruica Lincs, and Horns Rev 2 wind farms.

[3] Research into wind farm layout optimization was initiated in 1994 by Mosetti *et al.* [133].

power set-point tracking with load reduction [24]. Contrary to the aforementioned solutions to the wake problem, wind farm control does not conflict with the design objectives of wind farm developers as it requires no adjustment to turbine siting. Additionally, wind farm control may be implemented within existing operational wind farms, whether they possess a gridded or optimized layout. The demonstrated potential of such control systems has motivated a 2.7 million USD project, managed by the United Kingdom's Carbon Trust, involving the implementation of wind farm control trials within operational wind farms [34].

2.1.3 Objective

In 2015, Knudsen *et al.* [107] reviewed wind farm control literature with objectives of power maximization and power tracking. The various optimization and control algorithms that have been implemented in the existing literature were briefly discussed, and results related to power gains for these studies were presented. Major challenges related to wind farm control were also highlighted. The 2017 tutorial by Boersma *et al.* [24] provided greater detail regarding the wake effect, wake modeling, and wind farm control from a practical standpoint. Discussions covered a wide array of topics including actuation, sensors, control strategies, and observer design for wind farm controllers, as well as wind turbine modeling and control, and wake model theory, applicability, and fidelity. These review articles provide adequate information for engineers and researchers to gain familiarity with the field of wind farm control. What is lacking in this field is a quantitative assessment and comparison of published studies that identifies the benefits and drawbacks of specific control techniques and evaluation methods.

The objective of the current chapter is to provide such a quantitative review of wind farm control literature with focus on wind farm power maximization. The methods and results presented in existing studies are examined and compared based on a set of relevant criteria, and then used to draw conclusions regarding the effectiveness of control strategies in addition to the influence of evaluation techniques and simulated wind conditions. Furthermore, the current chapter quantitatively reviews recent publications involving field tests and high-fidelity simulations that have not been covered in existing review articles. It is our intention to first provide engineers and researchers with a readable and focused introduction to wind farm control for power maximization, to then offer a recent and comprehensive overview of progress in this field, and to finally present definitive conclusions regarding controller design and evaluation that will establish appropriate directions for future research.

The remainder of this chapter is organized as follows; Section 2.2 provides an introduction to various wind farm control concepts and controller properties; Section 2.3 reviews commonly employed aerodynamic models used to simulate wind farms; Section 2.4 represents the major contribution of the current chapter, and provides a detailed review of wind farm control literature; Section 2.5 delivers a quantitative overview that pieces together the major findings from reviewed works, and draws conclusions regarding the current state of research in the field; and Section 2.6 concludes the chapter with a summary of major findings and recommendations for future paths of research in wind farm control.

2.2 Wind farm control

This section introduces various wind farm control concepts that have been proposed and investigated within existing literature. Additionally, properties that have been used to distinguish between wind farm controllers are discussed.

2.2.1 Control concepts

Three notable wind farm control concepts have thus far been proposed for mitigating the effects of aerodynamic coupling within wind farms. These concepts include *power de-rating*, *yaw-based wake redirection*, and *turbine repositioning*; all of which are depicted in Figure 2.4.

Power de-rating

The oldest and most prevalent approach within existing literature, which was first proposed in 1988 by Steinbuch [169], is power de-rating (also commonly referred to as *axial induction-based control*). The axial induction factor is a measure of the momentum deficit experienced by air flowing past a wind turbine. It is directly related to the power and thrust coefficients of a wind turbine rotor. Referring to

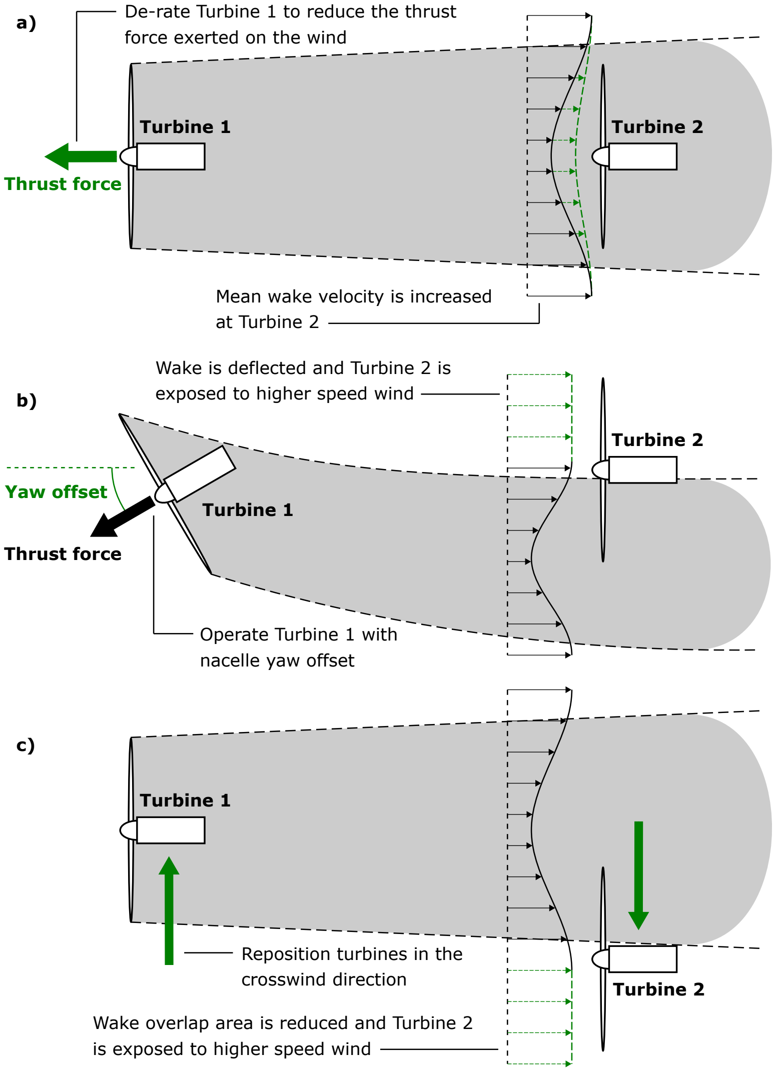

a)
De-rate Turbine 1 to reduce the thrust force exerted on the wind
Turbine 1
Turbine 2
Thrust force
Mean wake velocity is increased at Turbine 2
b)
Wake is deflected and Turbine 2 is exposed to higher speed wind
Turbine 2
Yaw offset
Turbine 1
Thrust force
Operate Turbine 1 with nacelle yaw offset
c)
Turbine 1
Reposition turbines in the crosswind direction
Turbine 2
Wake overlap area is reduced and Turbine 2 is exposed to higher speed wind

Figure 2.4: Overview of three notable wind farm control techniques proposed for mitigating the wake effect. The schematic shows a top-view of two wind turbines aligned with the incoming free stream wind. Green elements highlight changes in turbine operation and wake conditions associated with each control concept.

Figure 2.4a, de-rating turbine 1 reduces its axial induction factor, which diminishes power output and also decreases the magnitude of the overall thrust force that is exerted on the wind by its rotor [95]. This process lessens the momentum deficit within the wake that is generated [95]. Turbine 2, which is located further downstream and is aligned with this wake, is exposed to greater wind speeds as a result, and outputs electrical energy at a higher rate in relation to a baseline scenario in which turbine 1 had not been de-rated [95]. This concept is theoretically valid since, with the appropriate scale of power de-rating, the power loss associated with de-rating turbine 1 is outweighed by the power gain of turbine 2 [95].

Yaw-based wake redirection

A wind turbine operating with nacelle yaw offset deflects the downstream path of the generated wake in the crosswind direction [93, 181]. This phenomenon results from unsteady and asymmetric aerodynamic loads that are generated along the blades of a yawed rotor [29]. Due to misalignment of the rotor plane with the incoming wind, blades that are located in the top and bottom halves of the rotor plane experience different aerodynamic loads instantaneously [29]. As a result, load projections along the rotor plane are not in balance. This imbalance imparts forces that cause the wind to gain momentum in the crosswind direction [29]. Vortices generated in the wind field by these loads have also been shown to contribute to wake deflection [60].

The phenomenon of wake deflection forms the basis of the concept of yaw-based wake redirection. Referring to Figure 2.4b, operating turbine 1 with nacelle yaw offset causes the generated wake to be redirected in the direction of yaw [29]. The overlapping area between this deflected wake and the rotor of turbine 2 is reduced as a result, and a portion of the rotor of turbine 2 is exposed to higher speed wind that is less disturbed by the wake effect [58]. At the optimal nacelle yaw offset, turbine 2 experiences a power output rise that exceeds the power loss corresponding to the operation of turbine 1 with yaw misalignment [58].

Turbine repositioning

The final notable concept that has recently begun to gain attention is repositioning of floating offshore wind turbines. Regardless of the mechanism employed, the concept of turbine repositioning involves altering the locations of individual wind turbines within a wind farm in real-time with the aim of minimizing wake overlap [24]. Referring to Figure 2.4c, shifting the locations of the two turbines in opposite crosswind directions decreases the wake overlap at turbine 2, and raises its power output due to exposure to higher speed wind [58]. While the concepts of power de-rating and yaw-based wake redirection have received considerable attention from the wind farm control community, publications investigating the potential of turbine repositioning are limited. This trend stems from the narrow applicability of turbine repositioning to the floating offshore wind power sector [24], which is yet in an early stage of development [30]. Additionally, there is currently an absence of wind farm simulation tools capable of incorporating non-stationary wind turbines.

Active technologies that have been proposed for controlling the position of floating offshore wind turbines include under-water thrusters, which are commonly employed by the offshore oil and gas industry [165], winch mechanisms that are attached to floating offshore platforms and anchored to the seabed [153], and potentially thermally-actuated sewing thread artificial muscles that utilize electrical current to alter tension within mooring lines [114]. An alternative passive solution is *yaw and induction-based turbine repositioning* (YITuR), which involves using nacelle yaw and blade pitch control to manipulate the magnitude and direction of aerodynamic forces for the purpose of controlling floating platform positions [80, 81]. Due to these changes in yaw angles and thrust force magnitudes, the wind velocity field will also be manipulated in the process. A wind farm controller based on YITuR must therefore perform dynamic optimization in consideration of both turbine motion and wind field changes.

Other concepts

Additional wind farm control concepts include wake redirection via individual blade pitch control and nacelle tilt. Using nacelle tilt to redirect wakes employs the same underlying physics that permit yaw-based wake redirection, except wakes are redirected in the vertical direction rather than the crosswind direction [58]. Although redirecting wakes using nacelle tilt has shown greater capacity for wind farm power maximization relative to using nacelle yaw [58], the nacelle tilt degree of freedom is unavailable in modern wind turbine designs [62]. It is worth mentioning that if an optimal nacelle tilt angle is identified across a spectrum of potential wind speeds and atmospheric turbulence intensities, nacelle tilt actuation may not be necessary for wind farm power maximization. Standard utility-scale wind turbines currently possess a fixed shaft tilt offset to increases clearance between blades and the turbine tower [29]. This shaft tilt offset may be fixed at a predetermined optimal value for maximizing wind farm power production.

Individual blade pitch control may be used to redirect wakes by generating an uneven distribution of drag and lift forces across the area spanned by a wind turbine rotor [62]. The difficulty in realizing such a controller is that individual blade pitch control is conventionally used by turbine-level controllers for providing independent aerodynamic rotor braking and reducing asymmetrical blade loads [29]. As a result, an effective wind farm controller based on individual blade pitch control must simultaneously consider these turbine-level objectives in addition to wake redirection for wind power maximization [62]. This requirement yields an interesting yet challenging problem which has consequently subsided research on this specific control concept.

2.2.2 Controller properties

Properties of importance when assessing and designing wind farm controllers include their closed-loop structure, which governs requisite measurements and control signals, their dependency upon analytical or estimated wake models, the algorithms they use for characterizing the wind field, and the scope of permitted communication among wind farm level control units. Understanding these properties and their effects is necessary for justifying differences in the performances of different wind farm controllers.

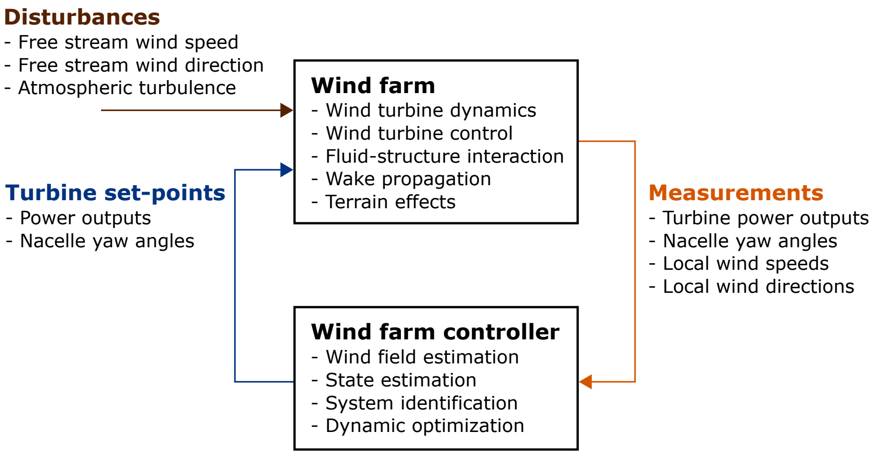


Figure 2.5: The closed-loop feedback structure of a general control system for wind farm power maximization. Listed measurements and turbine set-points do not encapsulate all possible signals; they are examples of commonly used data for wind farm control.

Closed-loop structure

Based on the studies reviewed in this chapter, a general closed-loop structure for wind farm controllers has been formulated and is presented in Figure 2.5. The plant to be controlled is the wind farm, whose dynamic behavior is a function of several underlying physical subsystems; wind turbine dynamics and control, which govern the motions of individual wind turbine components in response to aerodynamic and hydrodynamic loads; fluid-structure interaction, which influences the magnitudes of these loads, as well as their local effects on the wind field; and wake transport, which affects temporal and spatial changes in the wind field that result from mass and momentum conservation within a turbulent flow field. This flow domain contains mixing co-flowing regions and is subject to terrain effects.

The velocity and turbulence intensity of the free stream wind are regarded as disturbances acting on the plant since these parameters are undisturbed by wind turbines and are therefore considered as uncontrollable inputs. In general, wind farm measurements include local wind velocities and magnitudes of control degrees of freedom associated with individual wind turbines. Among studies pertaining to wind farm power maximization, these degrees of freedom are commonly turbine power outputs (or axial induction factors, which are directly related to the power coefficients of ideal actuator discs) and nacelle yaw angles, depending on the control concepts being implemented. Few studies in this research area are concerned with turbine-level control degrees of freedom such as blade pitch angles, tip-speed ratios, or generator torque values. It is preferential for wind farm controllers to compute power output set-points while individual wind turbine controllers determine the optimal combinations of these turbine-level degrees of freedom that deliver the desired power output at minimal mechanical loading.

Regardless of specific wind turbine measurements, wind farm controllers solve dynamic optimization problems in order to identify optimal turbine set-points that dynamically manipulate the wind field in such a way that the wake effect is mitigated and power losses are minimized. These set-points, which commonly encompass power outputs and nacelle yaw angles of individual wind turbines, are then tracked by wind turbine-level controllers. In addition to dynamic optimization, wind farm control systems may encompass wind field estimation algorithms for conditioning wind field measurements [10], correcting wake model inaccuracies [69], and estimating wake model parameters [72, 186]. These topics are briefly reviewed in Section 2.2.2.

Model dependency

In solving a dynamic optimization problem, wind farm controllers may utilize *model-based* or *model-free* approaches. In the case of model-based control, for which the general feedback structure is shown in Figure 2.6, dynamic optimization is performed on a wind farm model (which may be dynamic or steady) at each time step in order to determine optimal turbine set-points. The wind farm model may be analytically derived based on integral relations of fluid mechanics and assumptions regarding wake expansion [98], or its parameters may be estimated using measured data [72]. Machine learning methods such as neural networks may also be used for predicting wake behavior and wind turbine power production [132, 186]. Furthermore, model predictions may be corrected using state estimators such as Kalman filters [49]. Model-predictive control is commonly employed for model-based wind farm controller design due to its compatibility with constrained dy-

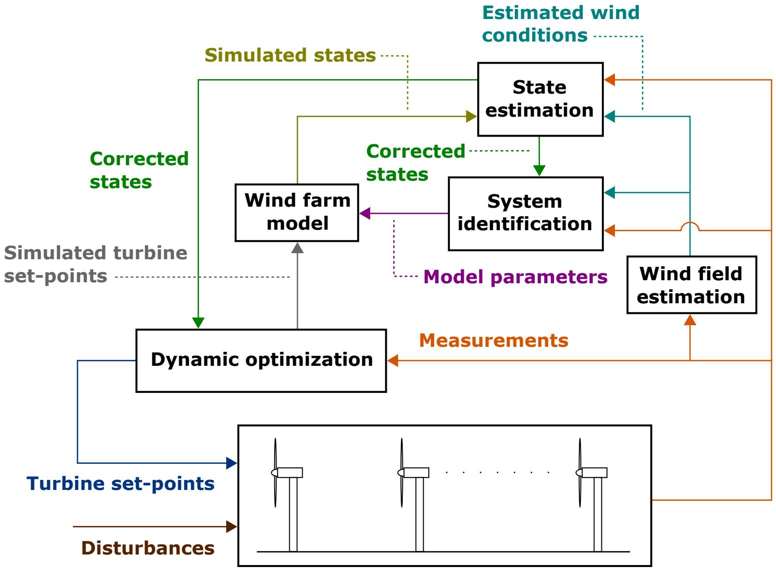


Figure 2.6: The closed-loop feedback structure of a model-based wind farm controller with wake model parameter and state estimation, and wind field estimation functions.

namic optimization.

Contrarily, model-free control involves dynamic optimization driven purely by measured data with no dependency on analytical or estimated wind farm models [24]. Such algorithms iteratively or continuously probe systems and compute future control actions based on observed changes in some objective function. Common implementations within wind farm control literature include gradient-ascent schemes, game theory, and extremum-seeking control. The rising popularity of such controllers within the wind farm control community is often attributed to the challenge of accurately modeling aerodynamic interaction within wind farms using control-oriented computationally inexpensive wake models [122]. Additionally, using higher-fidelity computational modeling techniques such as large eddy simulations (LES) is infeasible for online dynamic optimization due to the excessive computational resources required [77]. Model-free control should not be conflated with *data-driven* control. A data-driven controller, which may be model-free or model-based, computes control actions either directly from measured data or via a model estimated using such data [155].

Wind field estimation algorithms

Whether a wind farm controller uses a Kalman filter to correct wake model inaccuracies or system identification techniques to estimate wake model parameters, accurate measurements of wind speed and direction across wind farms are necessary [33]. The challenge presented here is that these measurements cannot be used directly and require conditioning. For instance, readings from wind vanes and anemometers mounted atop wind turbine nacelles are distorted by turbulence and the effects of moving blades on the nearby wind field [82]. In the absence of LiDAR technology and dedicated wind masts, appropriate algorithms are required for estimating wind conditions at different wind farm locations [10]. These algorithms are represented by the *wind field estimation* block in Figure 2.6. Although not the focus of the current chapter, wind field estimation is a broad and active area of research with significant implications in the fields of both wind turbine and wind farm control.

One recent effort implementing wind field estimation at the wind farm level is presented by Annoni *et al.* [10], who estimated the wind direction at the location of any wind turbine within a wind farm using wind vane measurements from the turbine itself and from those of its neighbors. Knowledge of local wind direction is crucial for identifying which pairs of wind turbines interact aerodynamically. Another example by Bottasso *et al.* [28] used blade bending moments to estimate wind speeds within different regions along a wind turbine rotor. This technique was then implemented experimentally in wind tunnel tests by Schreiber *et al.* [158] and Campagnolo *et al.* [33] to estimate the centerline location of a wake that was overlapping with a downstream turbine rotor. Estimates of this centerline wake location may be used to tune or correct parametric wake models intended for simulating wake redirection resulting from yaw misalignment.

Regardless of the means of obtaining wind speed and direction estimates, this information may ultimately be used for tuning wake model parameters as suggested by Gebraad *et al.* [72], training neural networks as performed by Yan [186], or correcting wake model inaccuracies using state-estimators as presented by Gebraad *et al.* [69]. Data obtained from wind field estimation may also be used for joint parameter-state identification. Doekemeijer *et al.* [49] designed state and parameter-state estimators for the medium-fidelity *Wind Farm Simulator* (WFSim) using various forms of Kalman filters. These estimators corrected wind speed predictions across the flow domain while identifying WFSim parameters. They evaluated their algorithms by simulating a 3×3 wind farm using the high-fidelity *Simulator for Wind Farm Applications* (SOWFA); the results from these simulations were treated as true measurements. Upon implementing Kalman filters, they reported convergence between wind speeds predicted by the two models within 300 sec. Beyond this point, errors between calibrated WFSim and SOWFA predictions were predominantly less than 1 m/s across the flow domain through which the free stream wind speed was 8 m/s.

Communication architecture

The final property of interest is the communication architecture of a wind farm-level controller, which may be *centralized* or *distributed*. A comparison of these architectures is provided in Figure 2.7. In a centralized control system, the wind farm controller receives measured wind velocity and control degree of freedom magnitudes from all wind turbines, and likewise transfers control set-points to all machines [122]. In a distributed wind farm control system, there is no centralized wind farm controller. Instead, the set-points of individual wind turbines are regulated by their respective distributed control units [70]. In the specific arrangement shown in Figure 2.7b, each turbine's wind farm-level control unit is permitted to communicate with those of the turbine's immediate neighbors. Alternatively, communication may be extended to an arbitrary collection of neighbors [122], or it may be further narrowed to solely a single downstream or upstream neighbor [70].

The motivation for developing distributed wind farm controllers is to reduce the size and complexity of the dynamic optimization problem being solved by any given control unit. Reducing the number of optimization variables enhances computational time-efficiency for each control unit; however a global optimal solution is less likely to be reached in comparison to using centralized control systems [70]. An additional benefit to developing distributed controllers is that they deliver a

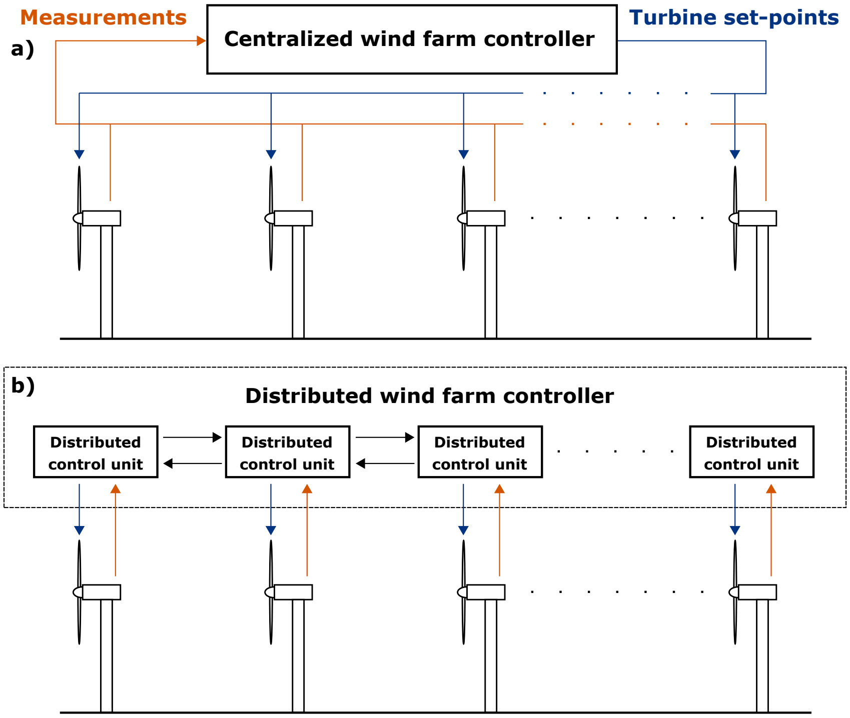


Figure 2.7: Comparison between the closed-loop feedback structures of (a) centralized and (b) distributed wind farm controllers.

generalized solution that may be implemented in wind farms of any size without altering computational efficiency and structure.

2.3 Wind farm modeling

Evaluating the performance of control systems on operational or experimental wind farms is an expensive and time-intensive task. Appropriate models for simulating wind farm aerodynamics are therefore necessary at the early stages of controller development [24]. Typically, such wake models are categorized along a spectrum ranging from low to high fidelity based on their numerical accuracy in predicting wind farm and wind turbine power outputs [24]. The intention of this section is not to review wake modeling literature in detail, but rather to provide the reader with sufficient knowledge regarding the simplifying assumptions, computational requirements, and accuracy of simulation tools that are commonly used to evaluate wind farm controllers. A summary of these wake models is provided in Table A.1. More comprehensive reviews of existing wind farm models are presented by Boersma *et al.* [24], Göçmen *et al.* [75], and Vermeer *et al.* [179].

2.3.1 Low-fidelity wake models

Low-fidelity wake models are based on integral relations of fluid mechanics, where the rates of change of fluid momentum and mass must be conserved across a specified control volume [56]. Such models are often referred to as *kinematic*, *engineering*, or *parametric wake models*, and they are best suited for offline static optimization and online dynamic optimization due to their simplicity and economical computational requirements [24]. The primary drawback of these models is that they provide no detail regarding velocity and pressure gradients within control volumes [56]. The phenomenon of wake expansion, which is the physical result of turbulent mixing between flow regions internal and external to wakes, is dependent upon velocity and pressure gradients within the fluid domain [115]. As a result, wake models based on integral relations rely upon parametric assumptions concerning the rate of wake expansion or the velocity distribution within wakes [24].

The earliest and simplest implementation of integral relations for constructing a wake model was presented by Jensen [91] in 1983. The methodology and assumptions inherent to this model are shown in Figure 2.8. Assuming uniformly distributed incoming flow that is subject to a uniform axial thrust force (actuator disc theory [121] is invoked here), the wind speed immediately downstream of the rotor is reduced by an amount that respects linear momentum conservation across the rotor. Upon defining a cylindrical control volume whose length spans from the rotor to some arbitrary downstream distance and whose diameter is equal to the wake diameter at this downstream location, the uniform wind speed at this location may also be found via linear momentum conservation. A major assumption necessary for this final step is a linearly expanding wake with a known expansion constant. A second assumption is zero radial flow through the cylindrical boundary of the control volume.

The *Jensen wake model* has since been modified and extended for improved

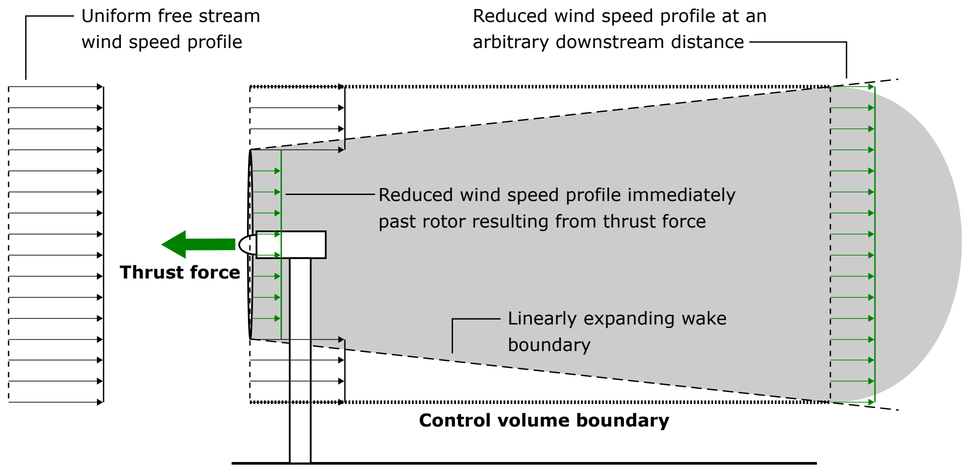

Uniform free stream wind speed profile
Reduced wind speed profile at an arbitrary downstream distance
Reduced wind speed profile immediately past rotor resulting from thrust force
Thrust force
Linearly expanding wake boundary
Control volume boundary

Figure 2.8: The control volume analysis used for computing momentum conservation in parametric wake models that are based on integral relations from fluid mechanics.

accuracy and the capability to capture additional aerodynamic phenomena. Katić *et al.* [98] modeled the interaction of multiple wakes by assuming that the kinetic energy deficit at any point in the wind field is equal to the sum of kinetic energy deficits of all overlapping wakes. As a result, the velocity deficit at any point is equal to the root-sum-square of pertinent wake velocity deficits. This extended model was named the *Park wake model*. Despite its simplicity and the absence of ground effects and wake meandering, the Park wake model has been shown to predict experimental wind farm power production with relative errors ranging from 0.1 to 15.9 % across a range of wind directions [98].

Gebraad *et al.* [72] incorporated the effects of nacelle yaw on the evolution of wakes. An analytical derivation by Jiménez *et al.* [93] was used to calculate the magnitude and direction of wake centerline deflection as a function of the nacelle yaw angle and axial induction factor. Additionally, while the Jensen wake model assumed uniform velocity distributions within wakes, Gebraad *et al.* [72] segmented wakes into three concentric zones to better predict velocity distributions. Finally, the parameters of this wake model were identified using high-fidelity simulation results obtained from SOWFA. This newly developed simulation tool was named the *Flow Redirection in Steady-state* (FLORIS) model and was recently released as standalone software by the National Renewable Energy Laboratory (NREL) for controller design and evaluation research. Recent releases of this software have incorporated Gaussian velocity profiles [17] and vorticity effects [124]. FLORIS has been shown to match wind farm power predictions from SOWFA with relative errors ranging from 0.1 to 5.3 % across various wind directions and turbine yaw settings [72].

Gebraad and van Wingerden [67] also introduced the *Flow Redirection and Induction Dynamics* (FLORIDyn) model, which extended FLORIS by including dynamic wake transport effects. As the nacelle yaw angle or axial induction factor of an upstream turbine are altered, and the generated wake is altered accordingly, a certain period of time must pass before this change in wake behavior is transported to a downstream machine. The FLORIDyn model considers this spatial and temporal delay by transporting steady-state wake solutions in the downstream direction. The absence of partial differential equations results in a dynamic control-oriented simulation tool for dynamic wind farm optimization.

The process of wake transport used in FLORIDyn is visualized in Figure 2.9. Assuming that a wind turbine has been facing the free stream wind with zero nacelle yaw for a long period of time, the steady-state wake profile will exist as shown in Figure 2.9a. Then, assuming that the wind turbine instantaneously yaws counterclockwise at time t, a deflected steady-state wake profile will result. In FLORIDyn, this deflected wake profile does not instantaneously take form downwind of the turbine. Instead, the downstream region is gradually updated with the deflected wake profile according to the distance traveled by the wake over an elapsed time-span. The wake profiles shown in Figure 2.9b and Figure 2.9c show the distance traveled by the deflected wake after elapsed time-spans of Δt_1 and Δt_2. Simulation tools that utilize transport delays in such a manner are categorized as *quasi-dynamic* wake models.

Gebraad [67] validated FLORIDyn against dynamic SOWFA simulations conducted over a 600 sec time-span with turbine yaw settings that were adjusted mid-simulation. FLORIDyn results deviated from SOWFA turbine power predictions with root-mean-square error (RMSE) values ranging from 7.9 to 11.7 % across different wind turbines, whereas FLORIS results deviated from SOWFA's with RMSE values ranging from 11.1 to 14.4 %. Approximating wake dynamics by transport-

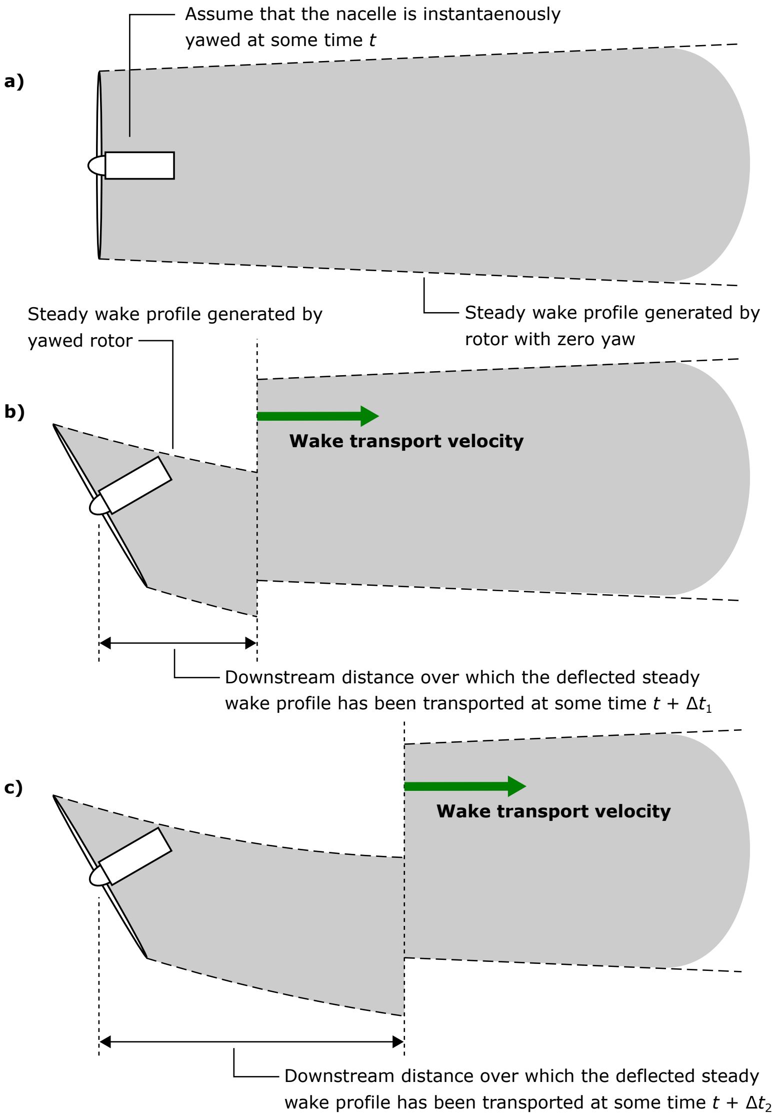

Assume that the nacelle is instantaenously yawed at some time t
a)
Steady wake profile generated by yawed rotor
Steady wake profile generated by rotor with zero yaw
b)
Wake transport velocity
Downstream distance over which the deflected steady wake profile has been transported at some time $t + \Delta t_1$
c)
Wake transport velocity
Downstream distance over which the deflected steady wake profile has been transported at some time $t + \Delta t_2$

Figure 2.9: A demonstration of steady-state solutions being transported in the downstream direction as a simplified means of approximating dynamic wake transport.

ing steady-state wake solutions in the wind direction therefore reduced absolute simulation error by as much as 6.5 %[4].

Our recent work [101] extended FLORIS to model platform motion in floating offshore wind farms. The modified simulation tool was named the *Floating Offshore Wind Farm Simulator* (FOWFSim). The wind farm was represented by a system of particles that were subject to aerodynamic thrust and mooring line forces. Similar to FLORIS, aerodynamic thrust forces were computed using actuator disc theory. Mooring line tensions were determined by solving the differential equations describing a static catenary that is either fully elevated or partially resting along the seabed. Cable stretch and seabed contact friction were also considered in this calculation. Finally, steady-state wake aerodynamics were captured using FLORIS.

2.3.2 High-fidelity wake models

High-fidelity wake models rely upon differential relations of fluid mechanics. In particular, they solve various forms of the turbulent Navier-Stokes equations, which are a set of partial differential equations that describe fluid momentum and mass conservation based on assumptions regarding turbulence modeling [56]. Unlike integral relations, solutions to differential relations provide details regarding velocity and pressure gradients within fluid domains [56]. As a result, phenomena that are dependent upon this information, such as turbulent mixing, wake expansion, and ground effects, are inherently captured by the partial differential equations of motion; no parametric assumptions are necessary. The primary drawback of higher-fidelity models is increased computational cost, since simulations may require from 10^3 to beyond 10^6 states, which render these tools less practical or infeasible for optimization and control [24].

High-fidelity wind farm models generally employ LES, which solve temporally and spatially-filtered forms of the three-dimensional Navier-Stokes equations that only capture eddies of relevant scale. One prominent high-fidelity simulation tool that is commonly used for wind farm controller evaluation is SOWFA, which is publicly available via NREL [40]. SOWFA couples an LES-based flow field

[4] Potential reduction in absolute simulation error was obtained from *Figure 5a* by calculating the maximum reduction in RMSE obtained for any single turbine's power production results [67].

model with NREL's *Fatigue, Aerodynamics, Structures, and Turbulence* (FAST) code, which models flexible multi-body wind turbine dynamics using the principal of virtual power (or Kane's method). The aerodynamic interaction between the wind and turbine blades is modeled using actuator line potential flow theory, which captures additional flow phenomena such as blade root and tip vortices in comparison to LES-based tools that model turbines using actuator disc theory; albeit at a higher computational cost with greater mesh density requirements. Wang *et al.* [182] compared SOWFA turbine power predictions to wind tunnel experimentation results and calculated relative discrepancies ranging from 0.2 to 14.0 % across various turbines with different yaw settings.

Other three-dimensional LES-based simulators developed specifically for wind farm applications include *SP-Wind* [7] and *University of Texas at Dallas-Wind Farm* (UTD-WF) [123]. SP-Wind reduces the computational cost of simulations by modeling wind turbines using actuator disc theory, whereas UTD-WF offers both actuator disc and actuator line modeling options. Martínez-Tossas *et al.* [123] compared LES-based wind farm modeling using actuator disc and actuator line models and showed that power predictions from the two methods relatively differ by less than 1.0 %, and that differences in velocity profiles are less pronounced in far-wake regions. Additional general-purpose computational fluid dynamics (CFD) solvers that have been used in wind farm control literature include EllipSys3D [166] and STAR-CCM+ [168].

2.3.3 Medium-fidelity wake models

Medium-fidelity models solve simplified forms of the turbulent Navier-Stokes equations in order to reduce computational complexity at the cost of neglecting certain flow physics. WFSim, developed by Boersma *et al.* [23, 25], solves a two-dimensional form of the unsteady turbulent Navier-Stokes equations along a horizontal plane located at the hub height of the wind turbines within a wind farm. Actuator disc theory is used to model thrust forces exerted on the wind by turbine rotors, and a mixing length turbulence model is used to compute momentum transfer via Reynolds stresses. The assumption of two-dimensional flow does not capture ground effects and wake rotation, which has been shown to influence wake centerline deflection [60, 72]. Nevertheless, Boersma *et al.* [23, 25] validated WFSim against SOWFA simulation results and showed that the medium-fidelity wake model provides a reasonable approximation of wake centerline deflection, which renders the tool useful for wake-redirection control studies. It was additionally shown that WFSim predicts spatially-averaged wake velocity profiles with up to an 82.1 % variance accounted for (VAF) match to SOWFA results.

The *Ainslie wake model* [4] solves the steady turbulent Navier-Stokes equations in cylindrical coordinates with the primary assumption that flow within a wake is axisymmetric with zero circumferential velocities. Additional properties of this model include the assumption of thin shear layers, which eliminate viscous terms from the momentum conservation equations, and the usage of an eddy viscosity turbulence model. Wake meandering is also accounted for by using a correction factor. Similar to WFSim, the Ainslie model captures neither wind shear nor wake rotation, and is not capable of modeling wake redirection due to yaw misalignment. When validated against wind tunnel experimental results, this medium-fidelity wake model predicted wake centerline velocity deficits with relative errors ranging from 0.5 to 8.4 % across different instances in time.

FarmFlow, which was developed by the Polytechnic University of Madrid [45], solves the parabolized form of the steady turbulent three-dimensional Navier-Stokes equations with turbulent transport coefficients computed using the k-ε turbulence model. A number of assumptions were made to render the flow problem parabolic, particularly the omission of streamwise diffusion[5], which eliminated the influence of downstream fluid conditions on upstream points in the fluid domain. This type of flow problem with one-directional influence, similar to an elementary boundary layer problem, is solvable using a computationally-inexpensive marching procedure [154]. The assumption of negligible streamwise diffusion is only valid in flows where boundary layer separation in the streamwise direction is insignificant [154]. Upon validation against wind tunnel experimental results, FarmFlow predicted maximum wake velocity deficits with relative errors ranging from 0.2 to 26.9 % across different tip-speed ratios and downstream distances [45].

[5] In the Navier-Stokes equations, streamwise diffusion refers to second-order velocity gradients in the wind direction.

2.4 Review of wind farm control literature

This section provides a detailed review of studies that have contributed to wind farm control research. Each of the three notable wind farm control concepts discussed in Section 2.2.1 are reviewed separately. Furthermore, referenced works are grouped into two categories; those that assess the steady-state potential of wind farm control concepts by conducting static optimization studies, numerical simulations, or experiments; and those that develop dynamic wind farm controllers capable of responding to real-time changes in wind conditions. Two subsections are therefore dedicated to each control concept; the first section concerns its steady-state potential, while the second details implemented controllers and dynamic optimization algorithms.

Summaries of all referenced studies are presented in Table A.2–Table A.6, while only works with significant findings are reviewed in detail to maintain brevity. These tables list properties of significance that are used to draw quantitative conclusions regarding controller performance and evaluation. These properties include optimization or control algorithms, input parameters, models used for optimization or control, models used for evaluation of optimal settings or controller performance, simulated wind conditions, and the resulting relative increase in wind farm power efficiency. Each listing under the *Relative efficiency gain* heading is accompanied by an endnote that highlights the origin of the result in the cited publication. The listings in all tables are sorted in order of increasing fidelity of the model or method used for evaluating controller performance or optimization results.

2.4.1 Steady-state potential of power de-rating

Studies evaluating the steady-state potential of power de-rating are summarized in Table A.2. A large number of studies have used low and medium-fidelity wake models (particularly the Park wake model) to determine optimal wind turbine set-points for power de-rating under steady wind conditions. These works may be distinguished based on the optimization scheme used and the turbine inputs that are varied, which may consist of tip-speed ratios λ and/or collective blade pitch angles β, axial induction factors a, thrust coefficients C_T, power coefficients C_P, turbine power output targets P, or generator torque values T_{gen}. The general consensus that may be drawn from literature based on parametric modeling is that power de-rating offers the potential to increase the instantaneous efficiency of a wind farm by up to 25 %, with no risk of efficiency loss. As higher fidelity wake models are employed however, and as additional aerodynamic phenomena such as turbulence and time-varying wind velocities are considered, the range of potential wind farm efficiency gains narrows or attains negative values (*i.e.* wind farm efficiency decreases relative to baseline control scenarios[6]).

Annoni *et al.* [9] delineated the discrepancies observed between results obtained using steady parametric wake models and high-fidelity LES-based solvers. They used both FLORIS and SOWFA to simulate a two-turbine wind farm subject to constant mean wind speed; SOWFA simulations considered a turbulence intensity of 6 %. A grid search optimization scheme applied to the steady Park model was used to calculate the optimal axial induction factor for the upstream turbine. This value was then implemented in FLORIS directly, while being converted into collective blade pitch angle adjustments for SOWFA. Simulation results showed that while the parametric FLORIS model predicted a wind farm efficiency gain of 24.82 %, SOWFA results indicated an efficiency loss of 9.35 %; this discrepancy was attributed to two phenomena. First, when an upstream turbine is de-rated, the increased kinetic power in the wind is predominantly concentrated along the wake boundary, which lies beyond the area swept by wind turbine rotors located directly downstream. As a result, these downstream wind turbines do not benefit when their upstream neighbors are de-rated. Parametric wake models fail to capture this phenomenon due to their assumption of uniform velocity distribution within simulated wake zones. Second, reducing the thrust force exerted on the wind by an upstream wind turbine decreases the turbulence intensity within its wake, which in turn diminishes the rates of wake expansion and momentum recovery. Parametric wake models fail to reproduce this observation as they rely upon wake expansion constants that are independent of turbine thrust forces.

Dilip and Porté-Agel [48] further validated the second conclusion drawn by Annoni *et al.* [9]. They used an LES-based solver coupled with a rotating actua-

[6] The baseline control scenario that is universally used in wind farm control literature is referred to as *greedy control*. This control scenario involves each turbine maximizing its own power production locally without consideration of its effects on the wind field or its neighboring turbines.

tor disc model to simulate a two-turbine wind farm subject to constant mean wind speed and different turbulence intensities. The collective blade pitch angle of the upstream turbine was varied by increments of 1 deg in the direction of both feather and stall. It was shown that power de-rating by pitching to stall (*i.e.* increasing thrust) was preferential over pitching to feather (*i.e.* decreasing thrust), since pitching to stall increased wake turbulence and resulted in faster wake expansion and momentum recovery further downstream. Despite this observation, conclusions regarding the potential of power de-rating as a wind farm control technique echoed those of Annoni *et al.* [9]. At the lower simulated turbulence intensity of 4.3 %, a maximum wind farm efficiency gain of 2.8 % was observed, while this value reduced to 0.5 % at the higher turbulence intensity of 6.3 %. The larger level of atmospheric turbulence offset the benefits of pitching the blades of the upstream turbine in the direction of stall. This outcome upheld the long-established understanding that downstream wake behavior is primarily influenced by atmospheric turbulence, not turbulence generated by viscous interaction [115].

Other studies employing CFD to assess power de-rating potential offer similar conclusions regarding its efficacy. Santoni *et al.* [156] used UTD-WF and the steady Park model to simulate a three-turbine wind farm subject to constant mean wind speed with an unspecified level of atmospheric turbulence. The steady Park model was used to determine the optimal tip-speed ratios of the turbines for maximizing the wind farm power output. The wind farm's performance with these optimal set-points was then evaluated using both the steady Park model and UTD-WF. The former parametric model predicted a wind farm efficiency gain of 5.7 %, while a value of 0.2 % was obtained using UTD-WF. Nilsson *et al.* [141] used EllipSys3D, an LES-based simulation tool that models wind turbine rotors using actuator disc theory, to simulate power production from the Lillgrund wind farm subject to constant mean wind speed and a turbulence intensity of 5.7 %. The wind plant contained 48 wind turbines organized in a gridded layout. The front row turbines were de-rated by altering their collective blade pitch angles by increments of 2 deg. It was shown that these adjustments resulted in wind farm efficiency losses ranging from 2.2 to 7.7 %. Across all simulated blade pitch angles, the power output rise of the second-row wind turbines was superseded by the power loss associated with pitching the blades of the front row turbines.

Kazda *et al.* [99] used MULTI3, an in-house CFD code based on the Reynolds-Averaged Navier-Stokes (RANS) equations and the k-ω turbulence model, to simulate aerodynamic interaction among four wind turbines within the Mont Crosin wind farm. A steady mean wind speed was simulated with a turbulence intensity of 13 %. The four wind turbines were situated in a single row, except the two trailing machines were shifted in the crosswind direction by an unspecified amount. This study was then unique in the sense that not all wind turbine rotors were aligned with the simulated wind direction. When assessing the effect of power de-rating within the upstream pair of turbines, altering the collective blade pitch angle and tip-speed ratio of the leading machine raised the pair's efficiency by 2.5 %. When considering all four machines however, the wind farm efficiency increased by 9.7 %. This large rise in efficiency may be attributed to the crosswind misalignment between the trailing pair of turbines and the leading turbine, which offers credence to conclusions drawn by Annoni *et al.* [9]. The trailing pair of turbines benefited from the increased kinetic power along the wake boundary of the leading turbine, since they were not directly aligned with the rotor of this leading machine.

Similar to high-fidelity simulations, experimental wind tunnel tests using scaled wind turbines have shown less potential for power de-rating in comparison to low-fidelity parametric simulations. Corten and Schaak [43] examined a 3×8 wind farm containing scaled wind turbines with rotor diameters of 25 cm subject to constant mean wind speed. The effect of power de-rating was assessed by adjusting the collective blade pitch angles of the leading two columns of turbines. A wind farm efficiency gain of 4.6 % was produced, with the most significant rise in power output corresponding to the second column of three turbines. Although this efficiency gain is sizable relative to predictions obtained from high-fidelity simulations reviewed in this chapter, it was acknowledged that the low Reynolds number of the blades may have inflated power output measurements. McTavish *et al.* [127] showed experimentally that reduced Reynolds numbers in scaled wind tunnel tests decrease the rate of wake expansion, which produces smaller wake diameters at the locations of downstream turbines. Coupling this phenomenon with observations reported by Annoni *et al.* [9], the added kinetic power attained from power de-rating, which is concentrated along wake boundaries, will be clustered closer to the swept rotor areas of downstream machines. The relative efficacy of power de-rating is therefore likely to be overestimated in scaled wind tunnel tests.

Other experiments have yielded similar efficiency gains. Machielse *et al.* [118] performed experiments on the same wind tunnel setup with a single row of seven scaled wind turbines. A maximum wind farm power gain of 3.46 % was obtained by pitching the blades of the leading two turbines by 2.5 deg in the direction of feather. Adaramola and Krogstad [2] performed wind tunnel experiments on a two-turbine wind farm containing scaled wind turbines with rotor diameters of 90 cm subject to constant mean wind speed. Maximum wind farm efficiency gains of 3.26 and 3.81 % were attained by increasing the tip-speed ratio and collective blade pitch angle of the upstream turbine, respectively. The Reynolds number based on blade chord length was an order of magnitude lower than typical values present among utility-scale wind turbines.

Campagnolo *et al.* [31] experimented with a wind farm containing three wind-aligned scaled turbines with rotor diameters of 1.1 m subject to constant mean wind speed with a turbulence intensity of 2 %. They reported a wind farm efficiency gain of 0.9 % when considering only the leading two turbines, and no improvement when including the third turbine in their calculations. The poor performance of the third wind turbine was attributed to possible wall interference. Bartl and Sætran [16] conducted wind tunnel experiments using two scaled wind turbines with rotor diameters of 90 cm subject to constant mean wind speed and turbulence intensity that decayed from 10 to 5 % across the flow domain. They observed no improvement in wind farm efficiency when power de-rating was implemented using either tip-speed ratio or blade pitch adjustments. These findings were attributed to the observation reported by Annoni *et al.* [9] regarding the concentration of added kinetic power beyond the rotor swept area of the downstream turbine.

Schepers and van der Pijl [157] and Boorsma [26] conducted field tests on two 2.5 MW wind turbines located in a research wind farm operated by the Energy Research Center of the Netherlands (ECN). The former study examined the effects of noise reduced operation on the total power output of two machines. This method of operation decreased the collective blade pitch angle and tip-speed ratio of the upstream turbine, which reduced its axial induction factor, albeit not by an amount that was optimal for wind farm efficiency gains. Results showed that power losses from the upstream turbine were recovered by the downstream machine, which validated power de-rating as a control concept, yet no demonstrable rise in wind farm power output was observed. The latter work varied the collective blade pitch angle of the upstream machine to simulate a reduction in axial induction, yet also produced inconclusive results. The absence of efficiency gains was attributed to several challenges in conducting field tests; varying wind direction, which rendered power de-rating ineffective as a wind farm control concept when two turbines were not nearly aligned with the wind; data collection from different periods in time, which introduced errors resulting from differing atmospheric conditions; and time-varying wind conditions that introduced transience in power and velocity measurements, despite the partitioning of data into bins based on wind speed and direction.

2.4.2 Controllers developed for power de-rating

A list of studies that have developed wind farm controllers based on the concept of power de-rating is provided in Table A.3. Abbes and Allagui [1], Gionfra *et al.* [73], Heer *et al.* [83], Kim *et al.* [106], and Kazda *et al.* [100] developed model-based power de-rating controllers that solved static offline optimization problems using low and medium-fidelity wake models. These controllers were then evaluated using similar wake models. Static optimization algorithms were implemented at specific wind speeds and directions to compute optimal wind farm set-points, which were then transferred to and dynamically tracked by individual wind turbine controllers. These control systems increased wind farm efficiency by up to 4 % across simulations considering turbulence and time-varying wind velocities. These efficiency gains are substantially lower than the potential gains predicted by static optimization studies discussed in Section 2.4.1, which reached 25 % and also employed low and medium-fidelity wake models. Furthermore, Kim *et al.* [106] demonstrated that substantial levels of atmospheric turbulence could yield efficiency losses of up to 2.88 %. Therefore, in addition to the shortcomings of power de-rating reviewed in Section 2.4.1, the dynamic tracking of optimal set-points by individual wind turbine controllers further deteriorates the potential of this wind farm control concept, even with the presumption that these optimal turbine set-points are known for all wind speeds and directions without the need for online optimization.

Alternatively, model-free controllers have also been implemented to compute optimal wind farm set-points using real-time dynamic optimization. Marden *et al.* [122] developed a centralized game-theoretic dynamic optimization algorithm for power de-rating. At each time-step, this algorithm perturbed the axial induction factors of individual wind turbines relative to preassigned baseline values using some probability distribution function. If the wind farm power output increased in response to these perturbations, then the baseline power output set-points were set equal to the perturbed values; otherwise, the baseline values were maintained. A distributed game-theoretic algorithm was also implemented in which each control unit had limited access to information from neighboring turbines only. This algorithm assigned moods of either content or discontent to each turbine, and optimum operation was reached when all turbines were content. In order for any turbine to maintain a content mood, it would have to sustain high power production, while its neighbors also remained content. The centralized controller was evaluated by simulating the Horns Rev wind farm (8×10) using the steady Park wake model. Under steady wind conditions and no turbulence, this controller increased the relative wind farm efficiency by $34.05\,\%$. At least 10^3 iterations were necessary however, which is indicative of the limitations that such an algorithm will encounter under time-varying wind conditions. A wind-aligned row of three turbines was also optimized using both controllers, and it was observed that the distributed controller required on the order of 10^5 iterations, while the centralized algorithm required only several hundred.

Zhong and Wang [188] developed distributed discrete adaptive filtering algorithms for power de-rating. These algorithms were fundamentally similar to the game-theoretic algorithms proposed by Marden *et al.* [122] in that they randomly perturbed wind turbine axial induction factors based on some probability distribution function and updated baseline axial induction factors only if the perturbations increased the power output of a cluster of turbines. The difference between the algorithms was that Zhong and Wang [188] updated probability distribution functions using an adaptive filtering step that rejected noise and disturbances. For a simulated 5×5 wind farm subject to constant mean wind speed, wind directions that varied every 200 iterations, and added noise for modeling turbulence, the proposed algorithm raised the wind farm efficiency by 3.87 % in 200 iterations. Simulations were conducted using the steady Park wake model and the effects of time-varying wind speed and dynamic wake transport were therefore not considered. In comparison, the game-theoretic algorithm proposed by Marden *et al.* [122] was also examined by Zhong and Wang [188], and resulted in an efficiency loss of 5.92 %, which was attributed to sensitivity of this algorithm to environmental disturbances.

Gebraad and Wingerden [70] developed a distributed gradient-ascent maximum power-point tracking controller for power de-rating. This controller optimized the axial induction factor of each turbine by considering only the power output fluctuations of the turbine itself plus that of its immediate downstream neighbor. Specifically, iterative measurements were used to numerically estimate the gradient of the power output of a pair of turbines with respect to the axial induction factor of the leading machine. Appropriate adjustments in this axial induction factor were then calculated based on the estimated gradient. A distributed quasi-Newton maximum power-point tracking algorithm was additionally developed to improve convergence. Simulations of the Princess Amalia wind farm (6×10) were conducted using the dynamic Park wake model (*i.e.* the steady Park wake model with time delay added for wake transport effects). With mean wind speeds varying between 8 and 10 m/s, the gradient-ascent controller raised the wind farm efficiency by an average of 4 %; the effect of turbulence was not considered however. Simulations were also conducted with constant mean wind speed to compare the proposed algorithms to the centralized game-theoretic controller presented by Marden *et al.* [122]. Results showed that the gradient-ascent and quasi-Newton controllers reached optimal solutions in approximately 30 minutes, while the game-theoretic approach required over 80 hours to achieve the same outcome. This finding is unsurprising since the game-theoretic controller randomly varied axial induction factors, while the maximum power-point tracking methods relied upon computed gradients.

Ahmad *et al.* [3] presented a centralized multi-resolution simultaneous perturbation stochastic approximation control algorithm for power de-rating. A standard simultaneous perturbation stochastic approximation algorithm randomly and simultaneously perturbs all wind turbine axial induction factors to determine wind farm power output gradients, and then updates the axial induction factors using a gradient ascent scheme. The proposed multi-resolution controller implemented this standard algorithm repeatedly with wind turbines being clustered into smaller groups after each iteration. These groups were referred to as resolutions, with larger groups being designated lower resolution values, and vice-versa. The solution obtained from optimizing at a lower resolution then served as the initial guess for optimization at a higher resolution. The performance of this controller was assessed by simulating the Horns Rev wind farm (8×10) using the dynamic Park wake model. Over the course of a ten-hour simulation, the mean wind speed and direction were varied using a series of step functions. That is, the simulated wind was steady for hour-long intervals, and was instantaneously altered at the end of each interval. The proposed algorithm monotonically raised wind farm efficiency by 32 % over the first eight hours of the simulation. It is important to note however that adjusting wind conditions in such a stepwise fashion is not tantamount to simulating stochastic time-varying wind behavior that is encountered in the field. All algorithms reviewed thus far must therefore be evaluated under more realistic turbulent wind conditions with varying direction.

Johnson and Fritsch [94] offered insight into the effects of stochastic time-varying wind conditions on the efficacy of wind farm control. They developed a centralized extremum seeking controller for power de-rating. The controller perturbed wind turbine axial induction factors with a continuous periodic signal in order to observe changes in wind farm power production. Depending on whether the variation in wind farm power output was in or out of phase with perturbation signals, the mean values of the axial induction factors were raised or lowered toward their optimal values. This controller was evaluated by simulating a wind-aligned row of three wind turbines using the dynamic Park wake model. Under steady mean wind conditions and a turbulence intensity of 2 %, the extremum seeking controller increased wind farm efficiency by 3.8 %. Increasing the turbulence intensity to 8 % then generated a 13.2 % loss in wind farm efficiency. Yang *et al.* [187] implemented a nested-loop extremum seeking controller for power de-rating, where optimal generator torque values were calculated consecutively starting from the most downstream turbine to the leading machine. Simulations were conducted on a wind-aligned row of three wind turbines using the dynamic SimWindFarm simulation tool. Under steady mean wind conditions with a turbulence intensity of 5 %, the extremum seeking controller increased wind farm efficiency by 1.3 %. These studies did not consider the effects of time-varying mean wind speed and direction. Nonetheless, the meager gains predicted in these studies highlight the fact that adapting to stochastic time-varying wind conditions is a major challenge for wind farm controllers.

Ciri *et al.* [41] developed a nested extremum seeking controller for power de-rating similar to that implemented by Yang *et al.* [187]. Key differences between the two studies were that Ciri *et al.* [41] used a discrete-time version of extremum seeking control with faster convergence, and they used the high-fidelity simulation tool UTD-WF with rotating actuator disc theory for controller evaluation. A wind-aligned row of three wind turbines was simulated under constant mean wind speed and a turbulence intensity of 8 %, and a relative wind farm efficiency gain of 7.8 % was reported. Ciri *et al.* [41] also tested an individual extremum seeking controller with the objective of maximizing turbine power outputs locally. Even though this algorithm did not constitute a wind farm controller, it interestingly raised wind farm efficiency by 7.6 % relative to a greedy scenario wherein generator torques were fixed at ideal design values. It was explained that these ideal design torque values were intended for non-waked wind turbine rotors, and that individual extremum seeking control was able to tune generator torque values for optimal local operation in the presence of rotor wake overlap.

A small number of studies have developed dynamic optimization algorithms that are model-based. Vali *et al.* [175] designed a centralized linear adjoint-based model predictive controller[7] for power de-rating that used WFSim to simulate wake aerodynamics. The term *adjoint-based* refers to additional steps taken to efficiently estimate the gradient of wind farm power output with respect to axial induction fac-

[7] An explanation of model predictive control is provided for readers who are unfamiliar with the topic. Traditional regulators such as proportional-integral-derivative control rely upon past and present information and measurements to determine control inputs at the current time-step. If one possesses a model of the system that is being controlled, then the future behavior of this system in response to any hypothetical future control input may be simulated. With this capability, a dynamic optimization problem may be solved over some future time horizon to determine the optimal future response of the system, as well as the corresponding future control signals. This process describes the basic principle of model predictive control, according to which a dynamic optimization problem is solved online at each time-step in order to determine optimal future control inputs. The objective of optimization may involve maximizing or minimizing an objective function with or without constraints; hence the relevance of model predictive control in maximizing wind farm power production.

tors. Lagrange multipliers were used to remove constraints from the optimization problem, while WFSim was linearized around an appropriate operating point at each time-step. The performance of the controller was assessed by simulating a 2×3 wind farm, again using WFSim. With constant mean wind speed and step changes in wind direction, the model predictive controller raised wind farm efficiency by up to 23.59 % during major portions of the 30-minute simulation, even though there existed roughly a three-minute period directly after specific wind direction changes where wind farm efficiency was reduced by a maximum of 3.6 %. This study did not consider the effects of turbulence nor varying wind speed.

Shu *et al.* [164] developed a centralized nonlinear model predictive controller for power de-rating with dynamic wind farm aerodynamics modeled by combining the Frandsen [64] and Larsen [110] wake models. Similar to the dynamic Park wake model, these wake models are based on integral relations from fluid mechanics; however they also consider wind shear effects and wake meandering due to atmospheric turbulence. A wind speed estimator was also designed to correct for mismatch between measured power outputs and those predicted by the wake model. The model predictive controller was validated by simulating a wind-aligned row of ten wind turbines using the same combined wake model that was used for model predictive control. Over the course of a four-minute simulation, during which the mean wind speed varied between 7 and 12 m/s, the controller raised wind farm efficiency by an average of 1.96 %. This study did not consider the effects of turbulence or time-varying wind direction. It is important to keep in mind the limitations of model predictive controllers that rely upon low fidelity wake models. As indicated in Section 2.4.1, these simulators tend to overestimate efficiency gains resulting from power de-rating due to inherent simplifications regarding velocity distributions and rates of wake expansion. Such model predictive controllers must therefore be evaluated using higher-fidelity wake models in order to ensure their effectiveness.

Goit and Meyers [76], Goit *et al.* [77], and Munters and Meyers [138] developed centralized model predictive controllers for power de-rating that solved dynamic optimization problems using SP-Wind, an LES-based CFD tool that models wind turbine rotors as actuator discs. The performance of their controllers was also evaluated using SP-Wind. These studies were unique in the sense that they utilized a high-fidelity simulation tool to dynamically compute optimal wind farm set-points. All other studies cited in this chapter made use of either low or medium-fidelity simulators, or relied upon model-free dynamic optimization algorithms. Large wind farm efficiency gains of up to 21.19 % were computed; however it was indicated in all cases that the computational cost of running LES is impractically high for online dynamic optimization. Instead, the authors observed patterns in the optimized flow field and proposed open-loop dynamic induction controllers that would replicate these patterns without the need for online dynamic optimization. Specifically, Munters and Meyers [138] proposed sinusoidal variations in the thrust coefficients of upstream turbines in order to replicate vortex rings that were shed by rotors within the optimized flow field. This dynamic induction controller increased power output from a 4×4 wind farm subject to turbulent wind with constant mean speed by a maximum of 5 %.

Van der Hoek *et al.* [176] recently conducted field tests of a power de-rating controller on two wind-aligned rows of five wind turbines and one wind-aligned row of six machines at the Goole Fields wind farm. Optimization studies were carried out offline using FarmFlow to determine optimal blade pitch angle set-points for different wind directions. These set-points were tracked by the collective blade pitch controllers of the turbines during field experiments. Data was collected over a year-long period during which optimal blade pitch settings were only implemented when the wind direction was near alignment with the rows of turbines. Upon applying various filters and corrections, including the imposition of limits on wind direction, blade pitch angle, and yaw angle variability, efficiency gains ranging from -15.9 to 11.1 % were reported for a single row of five turbines across a spectrum of wind speeds. Using measured wind speed distribution data, an increase of 3.3 % in annual energy production was predicted.

The results reported by van der Hoek *et al.* [176] indicate greater potential for power de-rating than was demonstrated in previous field tests by Schepers and van der Pijl [157] and Boorsma [26]; however a few factors may have inflated measured gains. First, wind turbines in a single row were not exactly aligned; therefore downstream wind turbines may have benefited from added kinetic power concentrated along the boundaries of the leading turbine's wake. This phenomenon was discussed when reviewing the work of Kazda *et al.* [99]. Second, the wind turbines were spaced between 2.3 and 3.1 rotor diameters apart. These distances are less that half the typical spacing of seven diameters that is used in numerical studies. A shorter distance between adjacent wind turbines decreases the scale of wake recovery; wind farm control strategies are therefore rendered relatively more effective. Important sources of error were reported by van der Hoek *et al.* [176]; FarmFlow is less accurate at inter-turbine spacings of five rotor diameters; despite data-filtering, wind turbines spent a substantial portion of ten-minute time intervals operating with yaw offset; and changes in turbulence intensity resulting from wind farm control affect wake recovery in ways that are not captured by wake models. All of these factors may have served to underestimate potential gains associated with power de-rating in field tests.

2.4.3 Steady-state potential of yaw-based wake redirection

A list of studies that have evaluated the steady-state potential of yaw-based wake redirection is provided in Table A.4. Several works have used parametric wake models (chiefly, modified versions of the FLORIS wake model) to perform static optimization studies that demonstrate the potential of this control concept. These studies may be distinguished based on the optimization algorithms used, and the specific modification made to the FLORIS wake model. The turbine parameter that is varied in these studies has universally been the yaw angle γ. The resulting wind farm efficiency gains that have been predicted range from 2.85 to 17.7 %, which represents a slight improvement relative to the potential of power de-rating evaluated using low and medium-fidelity wake models. Unlike the case with power de-rating however, higher-fidelity simulations and experiments have shown large potential for yaw-based wake redirection.

Fleming *et al.* [58] used SOWFA to simulate a two-turbine wind farm subject to constant mean wind speed and direction, a hub-height turbulence intensity of 4.9 % [40], and wind shear. Upon modeling several scenarios with the yaw angle of the upstream turbine being varied by increments of 5 deg, it was observed that a counter-clockwise yaw misalignment of 25 deg generated a wind farm efficiency gain of 4.6 %. Yawing the upstream nacelle clockwise by 25 deg resulted in an efficiency loss of 4.0 %. Gebraad *et al.* [72] attributed this asymmetry to the phenomenon of wake rotation. A clockwise spinning rotor exerts a counter-clockwise torque on the incoming wind, which causes the wake immediately downstream to rotate counter-clockwise. Rotation of the wake causes lower speed wind closer to the ground, which exists due to wind shear, to deflect in the crosswind direction. Hence, even in the absence of yaw misalignment, a spinning rotor generates asymmetric bias in the wind field that supplements wake deflection in one direction, and curtails it in another.

More recently, Fleming *et al.* [60] extended upon the above findings of Gebraad *et al.* [72]. Using SOWFA simulations of a single wind turbine, they showed that asymmetry in wake redirection is also substantially influenced by vortices generated due to yaw misalignment. Moreover, when a second turbine was introduced further downstream, its wake was deflected even with zero yaw offset. This phenomenon was referred to as *secondary steering*, and was attributed to two processes; the interaction between the rotating wake of the second turbine and the vortex generated by the first turbine; and the fact that yawing the upstream turbine generated partial overlap along the rotor of the second machine. Simulations were then conducted involving three rows of four wind turbines aligned with constant mean speed wind with a turbulence intensity of 6 %. Implementing yaw offset on the leading turbine in each row resulted in a wind farm efficiency gain of 12.5 %. This improvement relative to previous work by Fleming *et al.* [58] leads to two major conclusions; wake redirection is relatively more beneficial in larger wind farms; and vortices generated by the leading turbines combine as they are transported downstream to have profitable impact on trailing machines. This study also indicated that the low-fidelity FLORIS wake model actually underestimated the benefits of wake steering by not capturing vortex-related phenomena. This an interesting distinction between wake redirection and power de-rating, for which low-fidelity wake models tend to overestimate efficiency gains.

Churchfield *et al.* [39] used a dynamic LES-based CFD tool to simulate a single row of five turbines within the Fishermen's Atlantic City Wind Farm. Constant mean wind speed and direction, wind shear, and a turbulence intensity of 5 % were modeled. In addition to a baseline scenario with no yaw offsets, three simulations were performed during which the yaw angles of the four leading turbines were set to 15, 20, and 25 deg. At the largest yaw offset, a relative efficiency gain of 13.77 %

was observed, with diminishing returns apparent in the trend of reported data. This benefit was accompanied with a penalty of increased blade-root flap-wise bending moment fluctuations however. While the leading turbine experienced a reduction of 50 % in the root-mean-square (RMS) value of its blade bending moments, this RMS value increased by 15 to 45 % for the remaining turbines, with the largest value corresponding to the trailing turbine with no yaw-offset. These observations were attributed to the fact that yaw-based wake redirection causes partial wake overlap along the rotors of downstream turbines, which generates fluctuations in aerodynamic loads as individual blades traverse the rotor swept area. It was also suggested that individual blade pitch control would be an effective approach for mitigating this drawback.

Miao *et al.* [130] used the commercial CFD package STAR-CCM+ to simulate a two-turbine wind farm by solving the unsteady RANS equations with the k-ω turbulence model. Unlike CFD solvers designed specifically for wind farm applications, which use either actuator disc or line theories to model rotor aerodynamic loads, this study considered three-dimensional wind turbine blade geometries within the fluid domain and solved for aerodynamic forces directly via the equations of fluid motion. The vicinity of each wind turbine rotor was therefore discretized using a high-density rotating mesh, and a sliding mesh technique was used to link data transfer between rotating and stationary mesh regions. With constant mean wind speed and direction, wind shear, and turbulence intensity ranging from approximately 3 to 23 % across the fluid domain, yawing the upstream turbine by 30 deg yielded a wind farm efficiency gain of 2.11 %. The triviality of this efficiency gain relative to predictions by other simulators was attributed to the tendency of the k-ω turbulence model to predict excessive turbulence dissipation. This condition resulted in faster wake recovery, which would have reduced the relative benefit of any wind farm control strategy.

Adaramola and Krogstad [2] and Campagnolo *et al.* [31], whose wind tunnel experimental works on power de-rating were reviewed in Section 2.4.1, also investigated the potential of yaw-based wake redirection. Adaramola and Krogstad [2] used the same two scaled wind turbines with rotor diameters of 90 cm. After yawing the upstream rotor from 0 to 40 deg, they measured a peak efficiency gain of 12 % at a yaw angle of 30 deg. Campagnolo *et al.* [31] conducted their wake redirection experiments on two scaled wind turbines with rotor diameters of 1.91 m subject to constant mean wind speed and a turbulence intensity of 8 %. The yaw angle of the upstream turbine was varied from -20 to 20 deg, while the downstream turbine was simultaneously displaced in the crosswind direction. They observed a wind farm efficiency gain of 7.0 % when the upstream rotor was yawed by -20 deg and the two turbines were aligned with the wind. When the downstream turbine was displaced by 0.45 rotor diameters, the efficiency gain increased to 25.5 %, since repositioning the downstream turbine further contributed to reducing wake overlap. This result may be interpreted as a promotion for turbine repositioning.

Bastankhah and Porté-Agel [19] conducted wind tunnel experiments on wake redirection wherein the objective was to examine the effects of yaw angle distribution across a wind farm on overall power production. The wind farm consisted of a wind-aligned row of five scaled devices with rotor diameters of 15 cm. In a brute-force fashion, different combinations of yaw offsets ranging from 0 to 30 deg were implemented, and a maximum wind farm efficiency gain of 17 % was observed. This peak corresponded to a distribution in which the yaw offset was greatest at the leading turbine, while gradually dropping to zero at the trailing machine. An analytical wake model was also used to ascertain the effects of turbine spacing and turbulence intensity on the benefits of wake redirection. The results were in agreement with others presented in this review. Increasing turbine spacing allowed for more time and space for wakes to re-energize; thus reducing the relative gains obtained from wake redirection. Greater turbulence intensity values yielded the same outcome since larger turbulent kinetic energy quickened turbulent diffusion and wake recovery.

Wagenaar *et al.* [181] conducted field tests on a single row of three 2.5 MW wind turbines located at ECN's experimental wind farm. Incremental clockwise yaw offsets of the upstream turbine ranging from 0 to 16 deg were introduced, while only measured data pertaining to the wind direction aligned with the row of turbines were gathered and averaged. Analysis showed that the wind farm performance was highly scattered and sensitive to data selections, particularly to wind speed. At 4 deg of yaw offset, wind farm efficiency gains of 32.9 and 11.1 % were observed corresponding to wind speeds of 4 and 8 m/s, respectively, while a 16 deg yaw offset resulted in efficiency losses of 26.3 and 11.7 % at the same wind speeds, respectively. Additionally, efficiency results corresponding to a reference row of turbines without any yaw offset showed variability across different wind speeds and yaw angle offsets of the leading turbine. In light of these features of the collected data, no conclusions were drawn regarding the potential of yaw-based wake redirection.

Howland *et al.* [86] performed wake redirection field tests on a row including five 1.8 MW Vestas V80 wind turbines and one 2.0MW model. An analytical wake steering model based on Prandtl's lifting line theory [163] was calibrated using five years of data obtained from the test site. This model was then used to solve an analytical gradient ascent optimization problem to obtain the set of optimal yaw angles for maximizing farm efficiency. Data collected over a ten-day period showed a maximum relative efficiency gain of 47 % at wind directions close to alignment with the row of turbines. In these cases, operation without wind farm control resulted in partial overlap along the rotors of downstream machines. Introducing wake redirection then virtually eliminated the wake effect and resulted in substantial power gains. Conversely, when the incoming wind was directly aligned with the row of turbines, a smaller relative efficiency gain of 7 % was reported. Implementation of the optimal yaw angles also served to decrease the variability (*i.e.* the standard deviation in one-minute averaged data) of power measurements by 73 %. This outcome was attributed to the fact that wake redirection reduced the likelihood of downstream wind turbines operating below cut-in speeds as a result of the wake effect. Reducing power output variability was noted to contribute to reduced intermittency and improved reliability of wind energy in the power grid.

2.4.4 Controllers developed for yaw-based wake redirection

Studies that have developed wind farm controllers based on the concept of yaw-based wake redirection are summarized in Table A.5. Gebraad *et al.* [72] implemented a model-based version of the centralized game-theoretic algorithm presented by Marden *et al.* [122]. The developed controller was model-based in the sense that the game-theoretic algorithm was not driven by measured data, but rather by the steady FLORIS wake model. At each time-step, the algorithm used the steady FLORIS model to determine the optimal set of nacelle yaw angles across the wind farm for maximizing power production given the immediate wind speed and direction. This controller was evaluated using SOWFA by simulating two rows of three turbines that were subject to constant mean speed wind with wind shear and a turbulence intensity of 6 %. A relative wind farm efficiency gain of 13.03 % was reported for a scenario in which the wind was aligned with the turbine rows. For this wind condition, it was also shown that fatigue damage, computed via damage equivalent loading (DEL), generally decreased for the leading two turbines in each row (*i.e.* the turbines whose rotors were subject to yaw misalignment). Blade flap-wise bending DEL was reduced by 11.2 to 18.4 % for different machines, for instance. Of the two trailing turbines, the machine toward which wakes were redirected experienced minor changes in DEL. The other trailing turbine experienced significant rise in fatigue damage; 12.7 % for blade flap-wise bending DEL, 18.0 % for low-speed shaft torsional DEL, and 19.1 % for yaw bearing moment DEL.

Gebraad *et al.* [69] used the FLORIDyn wake model to develop a nonlinear model predictive controller for yaw-based wake redirection. A Kalman filter was introduced in order to account for sensor noise and power fluctuations resulting from turbulence; the Kalman filter would also correct mismatch between predictions from FLORIDyn and measured data. The controller was evaluated by simulating a wind-aligned row of two wind turbines using FLORIDyn. Under steady wind conditions, the model predictive controller produced an increase of 0.19 % in the wind farm efficiency. Such a small relative power gain was associated with the large spacing between the modeled wind turbines; this spacing was ten rotor diameters, as opposed to the more commonly simulated seven rotor diameter spacing. Increasing turbine spacing allows for more time and space over which wakes are re-energized by the free stream wind [115], which diminishes the relative benefits of any wind farm control strategy. The use of the Kalman filter generated substantial reductions in RMSE values between turbine power outputs predicted by SOWFA and FLORIDyn. RMSE values ranged from 13.4 to 18.4 % without the filter, and from 7.6 to 11.5 % upon inclusion. This result promotes the capability of dynamic control-oriented wake models to adapt to time-varying wind conditions.

Park and Law [146] developed a cooperative game-theoretic controller for combined power de-rating and yaw-based wake redirection using Bayesian-ascent dynamic optimization. The Bayesian-ascent controller operated in a fashion similar to previously described game-theoretic controllers, where axial induction factors and yaw angles were perturbed, and the measured wind farm power output influenced decisions at the next iteration. One major modification was that a Gaussian process (a non-parametric, probabilistic machine learning approach) was used to construct an approximated objective function (*i.e.* an approximated model that predicted wind farm power production) in real-time based on measured data. This approximated objective function was then used to solve a real-time optimization problem at each time-step to determine optimal axial induction factors and yaw angles for future time-steps. The controller was evaluated by simulating a 10×10 wind farm using a newly derived steady wake model[8]. Under steady wind conditions with measurement noise, the cooperative controller raised wind farm efficiency by $24.5\,\%$ after 50 iterations.

Park *et al.* [148] later tested this control algorithm in wind tunnel experiments consisting of a 2×3 layout of scaled turbines with rotor diameters of $1.5\,\mathrm{m}$. At constant mean wind speed, the controller monotonically increased the wind farm efficiency by $33.22\,\%$ after 40 iterations. Campagnolo *et al.* [32] also performed experimental wind tunnel tests during which they implemented the gradient-ascent maximum power-point tracking algorithm presented by Gebraad and Wingerden [70]; except this algorithm was applied to the concept of yaw-based wake redirection rather than power de-rating. Experiments were conducted on a wind-aligned row of three scaled wind turbines with rotor diameters of $1.1\,\mathrm{m}$ subject to constant mean wind speed. Implementation of the control algorithm increased the average wind farm efficiency by approximately $20\,\%$ after 200 seconds of operation. These two studies represent the only experimental evaluations of model-free wind farm control systems to date, and their results are encouraging. Nonetheless, these algorithms must be evaluated using larger scaled models with Reynolds numbers based on blade chord length that are representative of utility-scale wind turbines.

Similar to their studies of power de-rating, Munters and Meyers [137, 139] developed centralized model predictive controllers for combined power de-rating

[8] Similar to the Park wake model, this newly derived model relied upon momentum integral relations from fluid mechanics. However, rather than assuming uniform velocity distribution within wakes, this new model computed the radial velocity distribution using a Gaussian exponential function and performed numerical integration to determine the effective wind speed at some downstream wind turbine rotor.

and yaw-based wake direction. They used the LES-based CFD tool SP-Wind to perform online dynamic optimization, which rendered their controller impractical for real-world implementation. As with their earlier work however, they utilized observed patterns in the optimized flow field to propose open-loop controllers that would replicate such flow patterns. Specifically, Munters and Meyers [139] proposed two simplified open-loop controllers; one involving steady yaw offset that would deliver mean deflections of wake centerlines, and one implementing dynamic wake steering that would engender wake meandering. Upon implementing these open-loop controllers on a 4×4 wind farm subject to turbulent wind with constant mean wind speed, a relative efficiency gain of $14\,\%$ was obtained using the steady yaw offset open-loop controller, while a relative loss of $2\,\%$ was observed using dynamic wake steering.

Fleming *et al.* [59] conducted field tests of wake redirection at the Longyuan Rudong Chaojiandai offshore wind farm, which consists of 25 4 MW wind turbines arranged in a gridded layout. For the purposes of this study however, wake steering was only implemented via a single turbine that shadowed one out of three other machines at any given time depending on the immediate wind direction. The FLORIS wake model was used in conjunction with an optimization algorithm to compute a lookup table of optimal yaw angles as a function of wind direction. These lookup tables established the set-points that were tracked by the yaw controller of the regulated turbine. Based on collected data corresponding to the wind direction aligned with a pair of two turbines that were spaced seven rotor diameters apart, a wind farm efficiency gain of $5.56\,\%$ was reported. Meanwhile, over a 60 deg range of variation in the wind direction, efficiency gains ranged from -12.16 to $19.96\,\%$, with losses occurring at a minority of wind directions. These are promising results given the inconclusive outcomes that have typically been reported from field tests. Even more encouraging is the fact that the optimal yaw angles implemented in these field tests were calculated using an engineering wake model, the use of which would lead to highly efficient model-based wind farm control.

More recently, Fleming *et al.* [61] examined the performance of the above controller on two wind turbines within an unspecified commercial wind farm. Across an 80 deg spectrum of wind directions surrounding alignment with the row, wind farm efficiency gains ranging from -7.9 to $10.2\,\%$ were reported. The following additional difficulties associated with field testing were pointed out; biases in yaw angle set-point tracking that could not be accounted for due to intellectual property protection; the inability of FLORIS to capture terrain effects, which ultimately led to discrepancies relative to field measurements; the fact that FLORIS was not tuned to predict near-wake region behaviour, which was problematic due to the close spacing of the wind turbines; and significant atmospheric instability, which resulted from data being collected during the summer season. All of these factors may justify the limited range of efficiency gains recorded in comparison to the previous field test by Fleming *et al.* [59].

2.4.5 Steady-state potential of turbine repositioning

Studies that have examined the potential of turbine repositioning are listed in Table A.6. At the time of writing, no work has implemented a dynamic control algorithm involving this wind farm control concept. Fleming *et al.* [58] simulated a two-turbine wind farm using SOWFA and investigated the effect of incrementally relocating the downstream turbine in the crosswind direction. Constant mean wind speed and direction, with wind shear and a turbulence intensity of 4.9 % [40] were modeled. Their results showed that shifting the downstream turbine by 130 m (*i.e.* one rotor diameter) perpendicular to the wind direction increased the power output from the pair by 41.0 % relative to a baseline simulation with no crosswind shift. The computed trend in blade flap-wise bending DEL demonstrated that accumulated fatigue damage rose as turbine relocation increased, and peaked at 50 m of crosswind shift. The relative rise in bending DEL began to drop beyond this point, and was either negligible or negative (*i.e.* indicating load reduction) when the downstream machine had been shifted by an entire rotor diameter. Finally, as with yaw-based wake redirection, it was shown that there exists a preferred direction of turbine relocation due to the asymmetric bias in wake deflection that was discussed in Section 2.4.3.

Rodrigues *et al.* [153] proposed a mechanism for repositioning floating offshore wind turbines in real-time which consisted of tensioned mooring lines feeding into winches mounted on-board floating platforms. By regulating the lengths of three mooring lines, each platform could be relocated within a movable range that was outlined by an equilateral triangle with side lengths of approximately 330 m. Multiple optimization studies with different wind directions were performed on a 25-turbine floating offshore wind farm using the steady Park wake model with constant wind speed. The optimization goal was to determine the magnitude and direction of turbine repositioning, within the aforementioned movable range surrounding the installation locations of the floating platforms, that would maximize wind farm power output. These optimization results were then validated using FarmFlow simulations. For a wind farm with floating platforms installed in a gridded 5×5 layout, turbine relocation yielded wind farm efficiency gains ranging from -11.49 to 18.11 % across a 330 deg spectrum of simulated wind speeds.

A practical method for repositioning floating offshore wind turbines was proposed by Han *et al.* [81], and is illustrated in Figure 2.10. A floating offshore wind turbine implementing a greedy control algorithm, which aims to locally maximize power production, will operate at its optimal axial induction factor with zero yaw offset. This operating point corresponds to the faded turbine in Figure 2.10. Variations in the axial induction factor and nacelle yaw angle of this turbine alter the effective magnitude and direction of the thrust force applied to its rotor. The equilibrium established between this altered thrust force and the restoring forces generated by the mooring lines permits relocation of the floating turbine anywhere within a limited movable range. Unlike winch-based repositioning methods, this proposed strategy for relocating floating wind turbines using yaw and axial induction requires no additional hardware, while employing control degrees of freedom that are available in currently operational wind turbines.

In our recent work [101], the potential of the repositioning method proposed by Han *et al.* [81] was assessed. Optimization studies were carried out on a 3×6 floating offshore wind farm with semi-submersible type platforms subject to constant wind speed and direction. FOWFSim was used for both optimization and evaluation. Optimization was performed for various mooring line lengths ranging from 835 to 925 m in order to assess the potential of turbine repositioning as the movable range of the platforms was expanded. We reported that YITuR increased wind farm efficiency by approximately 16.5 % for mooring lines lengths beneath 875 m. Beyond this point, wind farm efficiency gains increased dramatically, with an increase of 53.5 % corresponding to the largest cable length of 925 m. It was

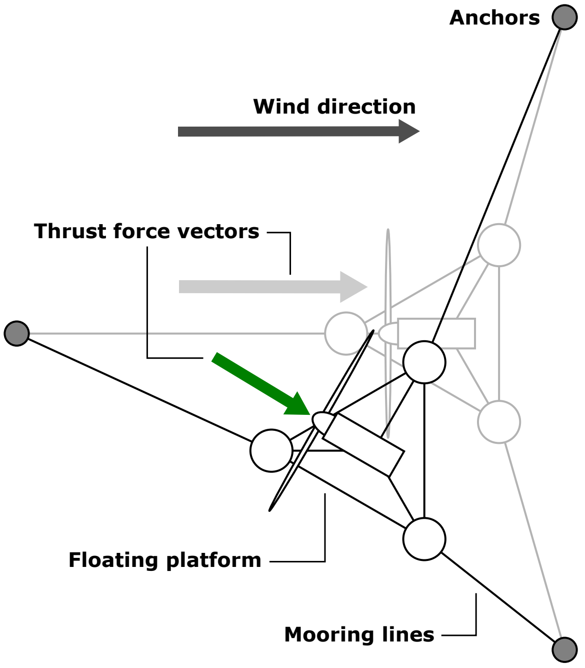


Figure 2.10: Schematic demonstrating the concept of yaw and induction-based turbine repositioning (YITuR).

noted that despite such potential, the practicality of permitting a movable range of nearly 90 m, which was the case for the largest cable length examined, must be carefully assessed.

2.5 Quantitative overview of published results

This section pieces together information that has been presented throughout the literature review from Section 2.4 with the aim of drawing conclusions regarding the impacts of controller properties, evaluation techniques, and simulated wind conditions on the potential of wind farm control concepts and algorithms.

2.5.1 Effects of evaluation techniques on predicted controller potential

The influence of evaluation techniques on the predicted potential of each wind farm control concept is assessed using the bivariate histograms shown in Figure 2.11–

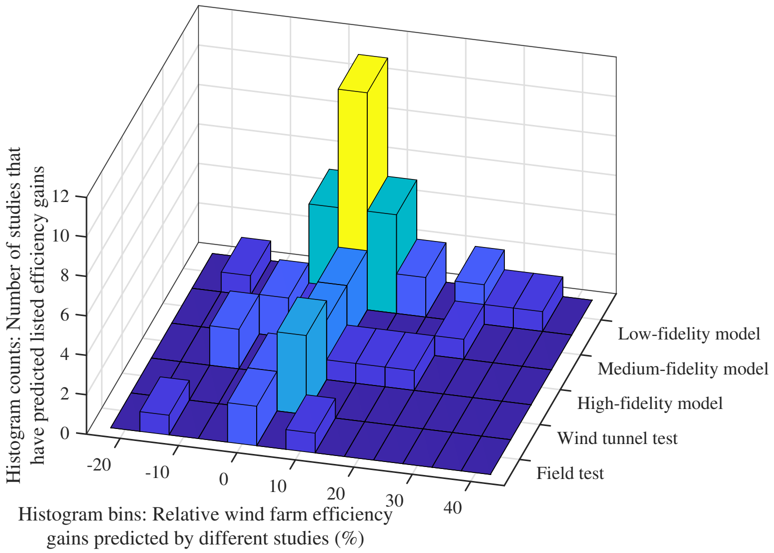


Figure 2.11: Bivariate histogram demonstrating the quantity of studies assessing power de-rating that have reported different wind efficiency gains using various evaluation models and techniques.

Figure 2.13. These plots demonstrate the number of studies that have used a particular evaluation method or tool to obtain relative wind farm efficiency gains corresponding to 5 % incremented bins. Focusing on Figure 2.11, which relates to power de-rating, the bulk of studies that have utilized low and medium-fidelity wake models have predicted efficiency gains ranging up to 25 %, while few works have attained values as high as 35 %. As the fidelity of the evaluation method is increased however, the range of predicted gains narrows to -10 to 10 % in the case of high-fidelity simulations, 0 to 5 % for wind tunnel tests, and 0 % or less for field tests.

The few high-fidelity simulations that predicted gains up to the 20 % bin correspond to the works of Goit and Meyers [76] and Munters and Meyers [138], who used a high-fidelity LES-based solver to dynamically optimize power production using power de-rating. As discussed in Section 2.4.2, such a control approach is impractical given the computational cost of running CFD simulations for online optimization. Upon implementing a more practical control strategy involving sinusoidal power de-rating, the wind farm efficiency gain predicted by Munters and Meyers [138] reduced to 5 %, which bears greater similarity to predictions by the remaining studies that employed high-fidelity simulations. The field tests that measured efficiency gains up to the 10 % bin were performed by van der Hoek *et al.* [176], and we suggested that close turbine spacing and crosswind misalignment among rows of turbines resulted in inflated power de-rating potential. The latter comment was also made regarding the work of Kazda *et al.* [99].

The discrepancy between results obtained via low and high-fidelity simulation tools has been attributed to the concentration of kinetic power gain in the vicinity of wake boundaries, and the effects of rotor thrust force changes on wake turbulence, neither of which are captured by lower-fidelity wake models. These phenomena may also justify the negligible efficiency gains observed among field tests; although the inconclusiveness of field test results has also been attributed to the difficulty in gathering consistent data for comparative assessment under stochastic time-varying wind conditions. Finally, while wind tunnel tests have shown promising results in comparison to those obtained via field tests, the presence of low Reynolds numbers and tunnel effects in such tests may have inflated the measured potential benefits. Despite these conclusions, power de-rating should not be ruled out as an effective wind farm control concept. Since the kinetic power gain from power de-rating is concentrated along wake boundaries, potential benefits may arise in scenarios where neighboring wind turbine rotors are not directly aligned with the wind. This phenomenon may have caused the efficiency gains reported by Kazda *et al.* [99] and van der Hoek *et al.* [176]. Moreover, studies with inconclusive results concerning power gains have shown that power losses from upstream de-rated turbines are at the very least recovered by their downstream neighbors. Power de-rating may therefore be employed for load mitigation without excessive power loss.

Shifting focus to Figure 2.12, it is evident that the predicted potential of yaw-based wake redirection is less sensitive to the fidelity of the evaluation method employed. All simulation tools and wind tunnel tests have indicated potential efficiency gains ranging up to the 35 % bin. The reported value in the 45 % bin for high-fidelity simulation once again corresponds to the work of Munters and Meyers [137], who employed an LES-based wake model for online optimization. The range of efficiency gains obtained via field tests has been mostly limited to the -5 to 10 % bins, which is foreseeable given the aforementioned challenges in han-

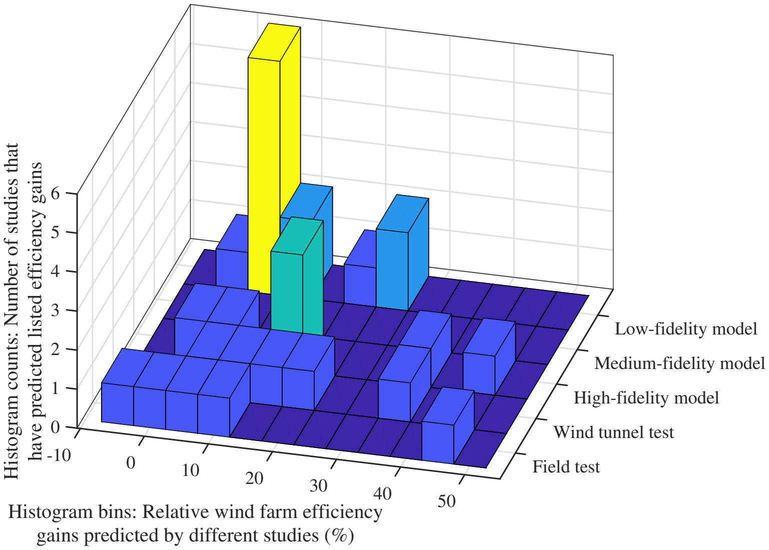


Figure 2.12: Bivariate histogram demonstrating the quantity of studies assessing yaw-based wake redirection that have reported different wind efficiency gains using various evaluation models and techniques.

dling field test data. The large efficiency gain in the 45 % bin for field testing was reported by Howland *et al.* [86], who calibrated their analytical wake model using five years of historical site-specific wind farm data. Their promising results should arouse much interest in the use of historical data for wind farm control.

On the whole, wake redirection field tests indicate a significant improvement relative to those concerning power de-rating. Furthermore, it is worth observing that significant losses in wind farm efficiency resulting from wake redirection have been seldom reported. There has also been general agreement between studies employing low-fidelity models and those relying upon high-fidelity simulations or field tests. This consensus may be attributed to the tendency of low-fidelity models to actually underestimate the potential of wake redirection, as reported by Fleming *et al.* [60]. Similarly positive results have been observed in the context of turbine repositioning, for which the histogram of existing results is plotted in Figure 2.13. Although research concerning the potential of this control concept is limited, available data has indicated potential gains exceeding 20 % regardless of the simulation tool employed.

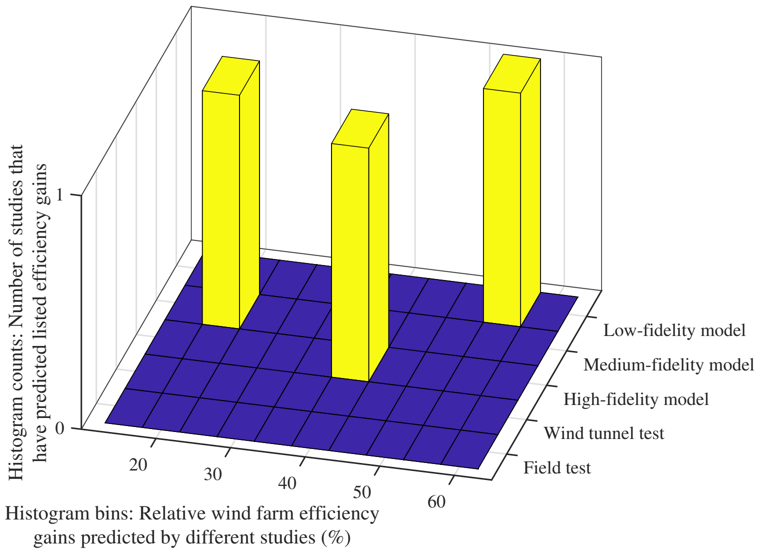


Figure 2.13: Bivariate histogram demonstrating the quantity of studies assessing turbine repositioning that have reported different wind efficiency gains using various evaluation models and techniques.

2.5.2 Effects of simulated wind conditions on predicted controller potential

Most studies that have been reviewed in this chapter have evaluated the potential and performance of wind farm controllers under steady wind conditions; that is, constant mean wind speed and direction with zero turbulence intensity. The few studies that have considered more realistic wind behavior have shown that time-varying wind conditions significantly deteriorate the performance of any wind farm controller. This outcome has held true regardless of the model used for controller evaluation, or the wind farm control technique in question. The explanation for this observed trend is that wind farm controllers, whether model-based or model-free, must adapt to time-varying wind conditions, which are stochastic and unpredictable disturbances. Throughout the period of time over which this adaptation occurs, a wind farm controller will be operating sub-optimally as it solves a dynamic optimization problem to compute an optimal setting. Furthermore, as wind conditions vary, the optimal set-points that must be tracked by individual wind turbine controllers also fluctuate, which further increases the duration of sub-optimal operation due to the time required for individual turbines to reach their optimal set-points.

Particular attention has been attributed to the influence of turbulence, and all findings indicate performance decay with rising turbulence intensity. Dynamic low-fidelity simulations conducted by Johnson and Fritsch [94] demonstrated that extremum seeking control for power de-rating resulted in efficiency gains dropping from 3.8 to $-13.2\,\%$ when the turbulence intensity of the incoming flow was raised from 2 to $8\,\%$. Likewise, dynamic medium-fidelity simulations performed by Kim *et al.* [106] involving power de-rating predicted a decrease in wind farm efficiency gains from 2.92 to $-2.88\,\%$ when the turbulence intensity of the flow was increased from 7.5 to $21\,\%$. Dynamic high-fidelity simulation results reported by Munters and Meyers [139] presented a similar outcome. Their proposed simplified controllers for wake redirection were implemented on a wind farm subject to both steady wind and airflow with a turbulence intensity of $8\,\%$. Wind farm efficiency gains dropped from 97 to $14\,\%$ as turbulence was introduced when testing their static controller, and 79 to $-2\,\%$ with dynamic wake steering.

The effects of time-varying mean wind speed and direction on controller performance have not been investigated as thoroughly as the impacts of turbulence. Nonetheless, the tendency of controller performance to diminish under these wind conditions holds. Studies by Gebraad and Wingerden [70] and Shu *et al.* [164] have shown that both model-based and model-free controllers are capable of adapting to time-varying mean wind speeds, thereby increasing wind farm efficiency averaged over the simulation period by $4\,\%$ in the former case, and $1.96\,\%$ in the latter case. However, compared to other low-fidelity evaluations involving constant mean wind speeds, which have yielded efficiency gains as large as $25\,\%$, the time-varying results reported by Gebraad and Wingerden [70] and Shu *et al.* [164] fall significantly short.

To date, no study has evaluated the performance of a wind farm controller with the wind direction varying continuously and stochastically over time. Several studies have incrementally altered the wind direction using step functions, and then allowed for their controllers to successfully adapt to the discrete changes [3, 175, 188]; however such an analysis is not tantamount to gradual and continuous variations in the direction of the wind. Simulation results presented by Vali *et al.* [175], who tested a model predictive controller for power de-rating, indicated efficiency losses as large as 3.6 % occurring at specific points in time corresponding to step changes in the wind direction. It is reasonable to conclude from this outcome that continuous variations in wind direction that are of substantial magnitude and frequency would therefore engender similar losses throughout the operation of wind farm control.

2.5.3 Effects of controller properties on performance

Given the scarcity of studies that have evaluated wind farm controller performance using either high-fidelity simulation tools or experimental techniques, no decisive conclusions may be drawn concerning the impacts of controller properties on performance. The following comments are nonetheless presented based on the limited findings that have been observed in this review.

Effects of model dependency

With regards to the model dependency of wind farm controllers, available results indicate that both model-based and model-free dynamic optimization algorithms offer potential to improve wind farm efficiency. The model-based game-theoretic wake redirection controller evaluated by Gebraad *et al.* [72], which performed online optimization of the FLORIS wake model, yielded an efficiency gain of 13.03 % when subject to turbulent wind simulated by SOWFA. The model-based lookup table wake redirection controller tested in the field by Fleming *et al.* [59], which utilized optimal set-points determined via offline optimization of the FLORIS wake model, generated efficiency gains up to 19.96 % across the majority of the spectrum of tested wind directions. The model-free gradient-ascent wake redirection controller evaluated by Campagnolo *et al.* [32] via wind tunnel experimentation produced an efficiency gain of at least 20 % after 200 seconds of operation. The game-theoretic controller tested via wind tunnel experimentation by Park *et al.* [148] also yielded impressive efficiency gains up to 33.22 % after 40 iterations. This controller was model-based yet data-driven in the sense that it utilized a Gaussian machine learning process for online dynamic optimization. Even the simplified open-loop control algorithms for power de-rating and wake redirection proposed by Munters and Meyers [138, 139] led to efficiency gains of 5 and 14 %, respectively.

These open-loop controllers may be categorized as model-free, yet not data-driven, since control actions are determined independently of any analytical or estimated model, or power measurements.

It is difficult to draw any deeper conclusions regarding the role of model dependency based on the magnitudes of efficiency gains reported in these studies. Each work utilized a different method of evaluation, and even studies involving wind tunnel tests are not directly comparable due to the differences in Reynolds numbers that were effective during experimentation. Even so, data-driven control approaches involving nonparametric machine learning or system identification are, by definition, more robust and practical for field implementation. Such controllers are capable of adapting to time-varying wind conditions while computing optimal wind farm set-points based on estimated models, rather than via random or gradient-based probing. Examples include the game-theoretic controller proposed by Park *et al.* [148], which utilized a Gaussian process for machine learning, the game-theoretic controller implemented by Gebraad *et al.* [72], which tuned the parameters of the FLORIS wake model using measured data, and the model predictive controller implemented by Gebraad *et al.* [69], which used a Kalman filter to correct states of the FLORIDyn model based on wind farm power measurements.

Effects of communication architecture

The effects of communication architecture on the performance of wind farm controllers also evade decisive conclusions due to the limited quantity of available results. Studies that have developed and evaluated both centralized and distributed wind farm controllers have yielded conflicting findings regarding the potencies of their algorithms. One the one hand, the distributed game-theoretic power de-rating controller developed by Marden *et al.* [122] required an iteration count that was three orders of magnitude greater than that of their centralized control algorithm to compute the optimal set-points of the simulated wind farm. Conversely, the distributed adaptive filtering power de-rating controller developed by Zhong and Wang [188] required 20 to 30 fewer iterations to determine optimal wind farm set-points compared to their centralized adaptive filtering algorithm. Both studies simulated wind farms of considerable size (an 8×10 wind farm in the former case, and a 5×5 wind farm in the latter work); thus the quantity of regulated wind turbines was likely not a contributing factor to the observed discrepancy between the iteration counts of the two works.

Gebraad and Wingerden [70] compared their distributed gradient-ascent controllers for power de-rating to the centralized game-theoretic controller proposed by Marden *et al.* [122]. The distributed gradient-ascent algorithms reached optimal operation in 30 minutes, while the centralized game-theoretic controller required 80 hours of simulation time to approach the same operational conditions. Campagnolo *et al.* [32] implemented one of the distributed gradient-ascent controllers developed by Gebraad and Wingerden [70] for wake redirection via wind tunnel experimentation. Their three-turbine scaled wind farm was optimized within approximately 3.3 minutes. In this case, wind farm size definitely contributed to the required iteration count for optimal operation, since the wind farm simulated by Gebraad and Wingerden [70] consisted of 60 wind turbines, which necessitated 30 minutes of simulation time for optimal conditions to be reached.

The results listed above show that the algorithmic operations of wind farm controllers contribute significantly to their required iteration counts for reaching optimal operation. The quantity of wind turbines within a regulated wind farm also impacts the time-efficiency of controllers. As a result, further research involving direct comparisons between centralized and distributed controllers is necessary for pinpointing the precise benefits and drawbacks of each communication architecture. The work by Gebraad and Wingerden [70] did validate a major preconception regarding communication architecture; in addition to optimizing a large number of parameters, a centralized controller must allow sufficient time for changes in the aerodynamic field, which result from control actions, to transport across the entire wind farm [146]. Centralized algorithms are thereby rendered unpractical for implementation in large wind farms from the standpoints of both computational cost and time-efficiency [70]. Distributed controllers circumvent this concern since their computational cost and required transport time remain unchanged irrespective of the wind farm size.

2.6 Conclusions and recommendations for future research

This chapter provided a review of three notable and prevalently examined wind farm control concepts for power maximization; power de-rating, yaw-based wake redirection, and turbine repositioning. Discussions focused on the potential of these concepts as well as the performance of control algorithms in the context of their properties, the evaluation methods used, and the simulated wind conditions. The current section summarizes the major conclusions drawn from the findings of published works, while offering recommendations on potential future research goals and practices that would be necessary for progress in the field of wind farm control.

With regards to the potential of each concept, power de-rating possesses the weakest capability of increasing wind farm power production. The added kinetic power generated via de-rating concentrates along the boundaries of wakes and does not sufficiently overlap with immediate downstream rotors. Power de-rating does remain a viable control concept for load mitigation and power set-point tracking however. Both yaw-based wake redirection and turbine repositioning have demonstrated substantial potential for raising wind farm efficiency, and this conclusion has been verified via high-fidelity simulations (and experimentation in the case of wake redirection).

In terms of evaluation techniques, lower-fidelity simulators do not capture velocity distributions and generated turbulence within wakes; both of which have contributed to over-prediction of efficiency gains associated with power de-rating controllers. On the other hand, predicted wind farm power output gains associated with wake redirection and turbine repositioning are less sensitive to these aerodynamic phenomena. This conclusion is verified by the commonality in simulation results obtained from models of varying fidelities.

On the experimental side, the low Reynolds numbers that are effective in wind tunnel experiments may be inflating measured efficiency gains for all control concepts. Meanwhile, field tests have provided the most conservative evaluations due to the stochastic time-varying nature of atmospheric wind, as well as the difficulties in extracting meaningful conclusions from the resulting measurements. In general, it has been shown that added layers of realism in terms of simulated wind conditions tend to deteriorate the performance of wind farm controllers. Added measurement noise resulting from turbulence and time-varying mean wind speed and direction reduces the length of time over which wind farms operate at optimal conditions, since wind farm controllers are continuously searching for optimal set-points in an environment with stochastic and unpredictable disturbances.

Further research is necessary for properly assessing the pros and cons of specific wind farm controller properties such as model dependency and communication architecture. Available data suggests that model-free, model-based, and even open-loop controllers all demonstrate potential for power maximization; however data-driven controllers involving system identification, machine learning, and wind field estimation are best suited for adapting to time-varying wind conditions. Published results also indicate that distributed controllers offer greater time-efficiency and computational expedience relative to centralized algorithms. This advantage stems from the lower wake transport periods and fewer optimization variables involved, respectively.

Based on these conclusions, the following recommendations may assist in propelling research within the wind farm control community:

- **Researching turbine repositioning:** The limited quantity of work investigating this control concept has demonstrated significant potential for wind farm power maximization. This field of research should receive considerable attention, particularly focusing on the practicality and safety concerns associated with relocating floating offshore wind turbines in real-time. Existing parametric wake models and LES-based CFD tools should also be extended to capture turbine motion as a degree of freedom.

- **Simulating time-varying wind direction:** This review has shown that no study has investigated the effects of continuously and stochastically time-varying wind direction on controller performance. This outcome has no doubt resulted from the difficulty in simulating fluctuating wind direction using models and wind tunnels. Nonetheless, both low-fidelity simulations (see Ahmad *et al.* [3] and Vali *et al.* [175]) and field tests have demonstrated indirectly that time-varying wind direction, much like unsteady mean wind speed and turbulence, serves to diminish controller performance; efforts should thus be exerted into assessing control algorithms under realistic wind scenarios.

- **Incorporating wind field and state estimation techniques:** Few of the studies reviewed in this chapter utilized state estimators and wind field estimation techniques. These methods have been used obtain undistorted wind field measurements within experiments conducted by Annoni *et al.* [10] and Schreiber *et al.* [158], and to correct wind field predictions from control-oriented wake models within high-fidelity simulations performed by Gebraad *et al.* [69] and Doekemeijer *et al.* [49]. Future wind farm control research should therefore incorporate such estimation techniques, and preferably assess their robustness experimentally.

- **Using machine learning for data-driven control:** Machine learning techniques such as convolutional or recurrent neural networks have successfully been employed to model wake phenomena [132, 186]. Furthermore, the use of nonparametric Gaussian processes has proven to be highly successful in wind tunnel tests conducted by Park *et al.* [148]. The applicability of these techniques, along with classical forms of system identification, to wind farm control should be further investigated.

- **Assessing joint wind farm and wind turbine-level control:** One major drawback associated with yaw-based wake redirection and turbine repositioning is the increase in fatigue damage encountered by rotors that are partially shadowed. Fleming et al. [58] demonstrated via SOWFA simulations that individual blade pitch control could reduce such accumulated fatigue damage. The performance of wind farm controllers operating in conjunction with turbine-level control systems, which include individual blade pitch regulation, should therefore be examined in greater detail.

- **Evaluating dynamic induction and wake steering controllers:** Works by Munters and Meyers [138, 139] have shown high potential for dynamic induction and wake steering controllers. Their open-loop regulators were designed based on detailed understanding of wind field phenomena that engendered wind farm efficiency gains. As a result, they required neither wake models, nor model-free online dynamic optimizers. Future research should investigate other variations of such open-loop controllers, as well as evaluate existing forms in wind tunnel experiments and field tests.

- **Using site-specific historical data for wind farm control:** The recent promising field test results by Howland *et al.* [86] on wake redirection have cast light on the merits of using site-specific historical data for wind farm control. While this study utilized static set-points determined via offline optimization, this approach should be further investigated in conjunction with online dynamic optimization algorithms and state-estimators that are capable of adapting to time-varying wind conditions.
- **Understanding wake phenomena using CFD:** The most significant leaps toward developing effective wind farm controllers have stemmed from CFD simulations of wake aerodynamics. For instance, simulations conducted by Annoni *et al.* [9] led to an understanding of the physical limitations of power de-rating, while the findings of Fleming *et al.* [60] and Munters and Meyers [138, 139] shed light on aerodynamic phenomena that may be exploited to further enhance wind farm control concepts. It is important that these research objectives remain highly active.

Chapter 3

Real-time Relocation of Floating Offshore Wind Turbine Platforms for Wind Farm Efficiency Maximization: An Assessment of Feasibility and Steady-state Potential

This chapter examines the steady-state potential and feasibility of *yaw and induction-based turbine repositioning* (YITuR), which is a wind farm control concept that passively repositions floating offshore wind turbines using existing turbine control degrees of freedom. To this end, the *Floating Offshore Wind Farm Simulator* (FOWFSim) is developed to model steady-state wind farm power production while considering floating platform relocation. Optimization studies are carried out with different floating wind farm design parameters and configurations. The objective is to determine sets of optimal wind turbine operating parameters that relocate floating turbines such that wind farm efficiency is maximized. Results show that the potential of YITuR is starkly limited by wind farm design parameters. In particular, anchors should be placed adequately far from floating platform neutral positions, mooring lines should be sufficiently long, and only specific mooring system orientations permit substantial gains in wind farm efficiency. With specific combinations of these parameters, simulation results show that the efficiency of a 7×7 floating offshore wind farm may be raised by 42.7 % when implementing YITuR in comparison to greedy operation.

3.1 Introduction

Clustering wind turbines in close proximity to form wind farms offers the primary benefit of capital expense reduction resulting from shared grid connection and project management costs [29]. From an operational standpoint however, clustering engenders aerodynamic interaction between individual wind turbines that diminishes the efficiency of wind farms as a whole [30]. As upstream machines extract the wind's kinetic energy, viscous interaction along their blades decelerates incoming flow and generates wakes with reduced average velocities [115]. Downstream wind turbines that are aligned with these wakes produce less electrical power as a result [30]. This phenomenon is referred to as the *wake effect* and may result in annual wind farm revenue losses ranging from 20 to 30 % with turbines spaced seven to ten rotor diameters apart [30]. Further, instantaneous wind farm efficiencies may be reduced by as much as 54 % [141].

One approach to mitigating the wake effect involves appropriately siting individual turbines within wind farms; that is, either increasing inter-turbine spacing [30], or utilizing wake models and historical wind velocity data to compute optimal wind farm layouts [161]. Alternatively, *wind farm control* employs the degrees of freedom of individual wind turbines to manipulate the aerodynamic field such that the wake effect is mitigated and wind farm power output is maximized [66]. Consequently, wind farm control requires no modification to turbine siting, which enables wind farm developers to prioritize layout design around terrain, grid connectivity, installation, transportation, and land permitting considerations. Additionally, wind farm control strategies may be implemented in operational wind farms, as demonstrated via recent efforts by the United Kingdom's Carbon Trust [34].

Two wind farm control concepts that have received considerable attention over the past decade include *power de-rating* (also referred to as *axial induction-based control*) and *yaw-based wake redirection*. The former concept de-rates upstream turbines to reduce the net thrust force exerted upon the wind, which in theory increases the average velocity within generated wakes that influence downstream machines [95]. Evaluations of power de-rating have reported differing results, with larger efficiency gains corresponding to parametric wake models [3, 9, 47, 122, 131, 160, 171], and negligible gains or losses associated with computational fluid dynamics (CFD) simulations [9, 48, 141] and field tests [26, 157]. Annoni *et al.* [9] identified the practical limitations of power de-rating by demonstrating via large-eddy simulations (LES) that the added kinetic power obtained from reducing the thrust force of an upstream turbine is primarily concentrated along the boundary of the generated wake, which occupies space beyond the swept-areas of rotors located directly downstream.

Yaw-based wake redirection operates upstream wind turbines with nacelle yaw misalignment, which generates unsteady and asymmetric blade loads that deflect generated wakes in the crosswind direction [93]. Vortices generated by these asymmetric loads also contribute to wake deflection farther downstream [60]. These processes result in reduced wake overlap along rotors that are located downstream from turbines operating with yaw offset [58]. This control concept has been proven effective via CFD simulations [39, 58, 72], wind tunnel experimentation [2, 31, 32, 148], and field tests [59].

A third wind farm control concept, broadly referred to here as *turbine repositioning*, minimizes wake overlap by relocating wind turbines in real-time [24]. This strategy is applicable to floating offshore wind turbines with non-taught mooring lines that allow for horizontal semi-submersible platform or spar buoy motion along the ocean surface [81]. The few studies that have evaluated the potential of turbine repositioning have reported promising results. Fleming *et al.* [58] simulated a two-turbine wind farm using the LES-based Simulator for Wind Farm Applications (SOWFA). They incrementally shifted the downstream turbine in both crosswind directions and reported a relative wind farm efficiency gain of 41 % at approximately one rotor diameter of relocation relative to a baseline simulation with no repositioning.

Rodrigues *et al.* [153] performed optimization studies using the parametric Park wake model [98] to maximize power production from a 5×5 floating offshore wind farm. The proposed mechanism for turbine repositioning consisted of taught mooring lines whose lengths would be regulated by winches mounted onboard floating platforms. By regulating the lengths of its three mooring lines, a single floating wind turbine could be repositioned within an equilateral-triangular area with a side length of $330\,\mathrm{m}$ surrounding its neutral position. Upon validating their optimization results using FarmFlow (a three-dimensional parabolized turbulent Navier-Stokes solver), a maximum relative wind farm efficiency gain of $18.11\,\%$ was reported across a $330\,\mathrm{deg}$ span of simulated wind directions.

Han *et al.* [81] proposed a passive mechanism that utilizes the aerodynamic force acting on wind turbine rotors to reposition floating structures; this concept is demonstrated in Figure 3.1. A floating offshore wind turbine that operates under greedy settings (*i.e.* maximizing power production locally) maintains zero yaw offset and optimal axial induction; the resulting position of such a turbine corresponds to the faded schematic in Figure 3.1. Introducing yaw misalignment and varying axial induction (*i.e.* by pitching blades or increasing/decreasing the tip-speed ratio) alter the magnitude and direction of the thrust force acting on the turbine's rotor. The result is platform motion until the altered aerodynamic force reaches equilibrium with restoring mooring lines forces. Appropriate manipulation of the yaw angle and axial induction factor of a floating wind turbine therefore permits platform relocation within a movable range that is limited by mooring line system properties. This relocation technique is referred to as *yaw and induction-based turbine repositioning*, and will be abbreviated as *YITuR* for convenience.

The objective of the current work is to evaluate the steady-state potential of YITuR by performing optimization studies under the assumption of idealized steady wind conditions. This task first requires the development of a floating offshore wind farm model that captures steady-state wake behaviour and turbine displacement. This tool is referred to as the *Floating Offshore Wind Farm Simulator* (FOWFSim). While assessing the effectiveness of YITuR, the influence of wind farm design parameters such as the quantity of turbines, inter-turbine spacing, mooring line length and orientation, and anchor location are investigated. A previous conference publication [101] briefly introduced FOWFSim and presented effi-

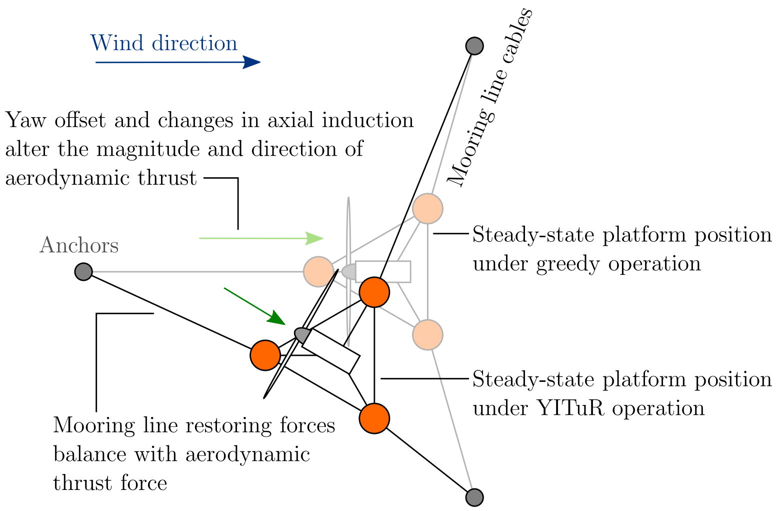

Wind direction
Mooring line cables
Yaw offset and changes in axial induction alter the magnitude and direction of aerodynamic thrust
Anchors
Steady-state platform position under greedy operation
Steady-state platform position under YITuR operation
Mooring line restoring forces balance with aerodynamic thrust force

Figure 3.1: Schematic demonstrating the concept of yaw and induction-based turbine repositioning (YITuR). The faded diagram corresponds to a floating offshore wind turbine subject to greedy operation (*i.e.* $a_{\text{greedy}} = 1/3$ and $\gamma_{\text{greedy}} = 0\,\text{deg}$).

ciency gain results for a single wind farm configuration. The current chapter details the mathematical model behind FOWFSim and presents comprehensive simulation results and analyses concerning multiple wind farm configurations and properties.

This chapter is organized as follows; Section 3.2 describes the structure and theory behind FOWFSim; Section 3.3 defines the optimization problem that is solved to determine optimal turbine operating parameters for maximizing wind farm efficiency; Section 3.4 presents the results of optimization studies carried out with different wind farm design parameters while providing detailed explanations regarding the observed phenomena; and Section 3.5 concludes the chapter by summarizing major findings and offering recommendations for future research concerning floating offshore wind farm control.

3.2 Mathematical model - FOWFSim

The current section describes in detail the structure and theory behind FOWFSim, which is ultimately used to model steady-state floating offshore wind farm performance. The FLOw Redirection In Steady-state (FLORIS) wake model presented by Gebraad *et al.* [72] is combined with a newly-derived steady-state two-dimensional platform displacement model. The result is a simulation tool that, for given mooring system, turbine, and platform properties, determines the position and power output of every turbine within a floating wind farm in response to steady wind conditions and turbine operating parameters. It is assumed that all wind turbines within a given wind farm are identical; therefore rotor, platform, and mooring system dimensions are common to all machines and structures. Further, discussions of the model are focused on floating wind turbines with semi-submersible platforms; the model may be readily extended to simulate spar buoys by simply altering the locations of mooring line fairleads.

3.2.1 Wind farm description

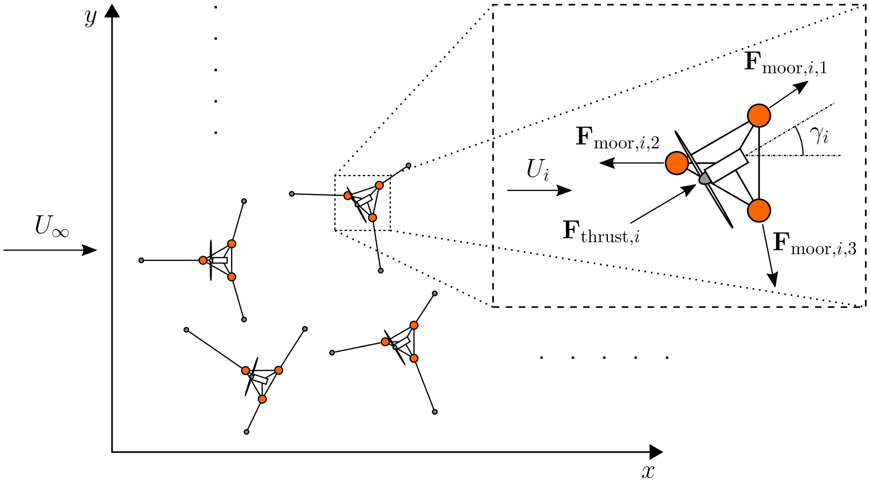
As shown in Figure 3.2, a floating wind farm is modelled as a system of N particles distributed along the two-dimensional ocean surface. Each particle represents a floating wind turbine, and the turbines are numbered according to the set $\mathscr{F} = \{1, 2, \cdots, N\}$ in an ascending order based on their downstream locations. If two turbines are located at the same downstream distance, the structure with the most negative crosswind location is numbered first. The subscript $i \in \mathscr{F}$ is used to indicate a particular wind turbine. The most upstream turbine is therefore labeled as $i = 1$, the most downstream machine is denoted by $i = N$, and all intermediary turbines are numbered accordingly.

Assuming steady wind conditions, the positive x direction is aligned with the free stream wind (denoted as U_∞ in Figure 3.2), and the y direction is chosen according to the right-hand rule such that the z axis points upward and normal to the ocean surface. The resulting outputs of the system $\mathbf{x}$ and $\mathbf{y}$ represent the locations of all turbines along the horizontal x-y plane as follows:

$$\mathbf{x} := \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}, \tag{3.1}$$

$$\mathbf{y} := \begin{bmatrix} y_1 & y_2 & \cdots & y_N \end{bmatrix}^{\mathrm{T}}, \tag{3.2}$$

where x_i and y_i denote the location of turbine i along the ocean surface. Also, the



y
x
U_∞
U_i
$\mathbf{F}_{\mathrm{moor},i,1}$
$\mathbf{F}_{\mathrm{moor},i,2}$
$\mathbf{F}_{\mathrm{moor},i,3}$
$\mathbf{F}_{\mathrm{thrust},i}$
γ_i

Figure 3.2: Schematic demonstrating the representation of a floating offshore wind farm as a system of N particles distributed along the two-dimensional ocean surface. The loads and local wind velocity U_i that are incident on a single turbine i are also shown.

position vector $\mathbf{r}_i$ of turbine i is defined as follows:

$$\mathbf{r}_i := \begin{bmatrix} x_i \\ y_i \end{bmatrix}. \tag{3.3}$$

These positions are functions of the inputs to the system $\mathbf{a}$ and γ, which respectively symbolize the axial induction factors and yaw angles of all turbines as follows:

$$\mathbf{a} := \begin{bmatrix} a_1 & a_2 & \cdots & a_N \end{bmatrix}^{\mathrm{T}}, \tag{3.4}$$

$$\gamma := \begin{bmatrix} \gamma_1 & \gamma_2 & \cdots & \gamma_N \end{bmatrix}^{\mathrm{T}}, \tag{3.5}$$

where a_i and γ_i denote the operating parameters of turbine i. All yaw angles are measured as positive counterclockwise from the x axis. At $\gamma_i = 0\,\mathrm{deg}$, turbine i is therefore facing the incoming wind with zero yaw offset.

Each turbine is subject to an aerodynamic thrust force vector $\mathbf{F}_{\mathrm{thrust},i}$ and three mooring line force vectors $\mathbf{F}_{\mathrm{moor},i,j}$ for $j \in \mathscr{M}$, where $\mathscr{M} = \{1,2,3\}$ represents the set of indices that identify the three mooring lines attached to any given floating platform. The magnitude of $\mathbf{F}_{\text{thrust},i}$ is a function of the effective wind speed U_i at turbine i, which is dependent upon the location of turbine i as well as the positions and operating parameters of all machines that are situated upstream of turbine i. Conversely, $\mathbf{F}_{\text{moor},i,j}$ is independent of the aerodynamics of the wind farm and is solely a function of the position of turbine i.

It is important to acknowledge the limitations and assumptions behind FOWF-Sim. First, it neglects floating platform motion in the z direction (*i.e.* platform heave), as well as rotations about the x and y axes (*i.e.* platform roll and pitch). Semi-submersible platforms are designed to exert large buoyancy and heave drag forces in order to limit these motions. Their effects on overall wind farm power production are neglected. Second, platform rotation about the z axis (*i.e.* platform yaw) is neglected. It is assumed that platform yaw resulting from mooring line nonlinearity has negligible effect on steady-state platform displacement. Third, the impacts of ocean waves and currents are neglected. Rotor translational motion resulting from wave-induced platform oscillation is assumed not to significantly affect average wind power production. Steady ocean currents would induce a bias in floating turbine positions; however this effect is neglected in the present study since it is a first examination of YITuR.

3.2.2 Wind farm power output and efficiency

The efficiency of a wind farm η_{farm} as the ratio of wind farm power output P_{farm} to hypothetical wind farm power output $P_{\text{farm},\infty}$ in the absence of wakes is expressed as follows:

$$\eta_{\text{farm}} = \frac{P_{\text{farm}}}{P_{\text{farm},\infty}}. \tag{3.6}$$

The power outputs P_{farm} and $P_{\text{farm},\infty}$ are calculated as follows:

$$\begin{aligned} P_{\text{farm}} &= \sum_{i \in \mathscr{F}} P_i, & (3.7) \\ P_{\text{farm},\infty} &= N \frac{1}{2} C_{P,\max} \rho A U_\infty^3, & (3.8) \end{aligned}$$

where ρ is the density of air, A is the rotor swept area, and U_∞ is the steady free stream wind speed. The variable $C_{P,\max}$ is the optimal power coefficient of the simulated wind turbines under greedy operation, which entails optimal axial induction and no yaw offset (*i.e.* $a_{\text{greedy}} = 1/3$ and $\gamma_{\text{greedy}} = 0\,\text{deg}$) [121]. The power output P_i of turbine i is calculated as follows [29]:

$$P_i = \frac{1}{2} C_{P,i} \rho A U_i^3, \tag{3.9}$$

where U_i is the wind speed at the location of turbine i. The power coefficient $C_{P,i}$ of turbine i is calculated according to actuator disc theory with corrections applied due to yaw offset as follows [128]:

$$C_{P,i} = 4a_i\left(1 - a_i\right)^2 \eta_P \left(\cos\gamma_i\right)^{p_P}, \tag{3.10}$$

where η_P is the power conversion efficiency and p_P is a parameter required for fitting the power coefficient to experimental or numerical data. The values for these parameters have been obtained from existing implementations of FLORIS [72].

3.2.3 Wind farm aerodynamics - FLORIS

In order to calculate effective wind speeds at all turbine locations within a wind farm, the FLORIS wake model as outlined by Gebraad *et al.* [72] is directly employed. This parametric wind farm simulation tool utilizes integral momentum conservation relations from fluid mechanics to estimate centerline deflection and velocity deficit within wakes [72]. Specifically, the Park wake model [91, 98] is used to compute velocity deficits, and an analytical derivation by Jiménez *et al.* [93] serves as a basis for wake deflection calculations. The first strength of the FLORIS wake model lies in the fact that its parameters have been estimated using simulation results from SOWFA. The second is that, rather than assuming uniform velocity distribution within wakes, FLORIS divides each wake into three concentric zones in order to better capture velocity gradients in relation to its predecessors. Recent releases of the FLORIS wake model have implemented Gaussian velocity profiles within wakes [17] and curled wake effects [124] in order to better predict wind farm aerodynamic behaviour. The original version of FLORIS is used in the current work due to the unavailability of these releases when simulations were conducted. Further, the wake model currently employed in FOWFSim may be readily replaced by an alternative simulation tool without altering other computational processes.

The FLORIS model is implemented in a generalized manner wherein every turbine may be affected by the wakes generated by any of its upstream counterparts. This assumption is necessary since the locations of floating platforms are unknown variables, and specific combinations of turbine operating parameters may bring any two turbines near alignment with the wind. The only simplifying assumption that may be imposed given the nature of movable platforms is that turbines operate independently of their downstream neighbors. The resulting computational structure of FOWFSim is shown in Figure 3.3. When computing the local wind speed U_i at turbine i, the FLORIS wake model accepts as inputs the free stream wind speed U_∞ and the operating parameters and positions of all turbines located upstream of turbine i. In other words, U_i is a function of a_k, γ_k, x_k, and y_k for $k \in \mathscr{U}_i$, where the set $\mathscr{U}_i = \{1, 2, \cdots, i-1\}$ numbers all wind turbines that are located upstream of turbine i. The turbine displacement and mooring line models then determine the steady-state location of turbine i. This computation is repeated for each turbine starting from the most upstream machine and proceeding downstream. The following subsections provide greater detail on the remaining two functions shown in Figure 3.3.

3.2.4 Floating turbine displacement model

The steady-state position of turbine i is determined by the equilibrium established between all forces that act upon it. As shown in Figure 3.2, these entities are the aerodynamic thrust force $\mathbf{F}_{\text{thrust},i}$ and the net mooring line force $\mathbf{F}_{\text{moor},i}$. The equilibrium between these forces is described as follows:

$$\mathbf{F}_{\text{thrust},i} + \mathbf{F}_{\text{moor},i} = 0, \tag{3.11}$$

where each of these vectors contains the x and y components of the respective forces acting on turbine i. Since $\mathbf{F}_{\text{thrust},i}$ is a function of U_i, and since both U_i and $\mathbf{F}_{\text{moor},i}$ are nonlinearly dependent upon floating platform positions, the above

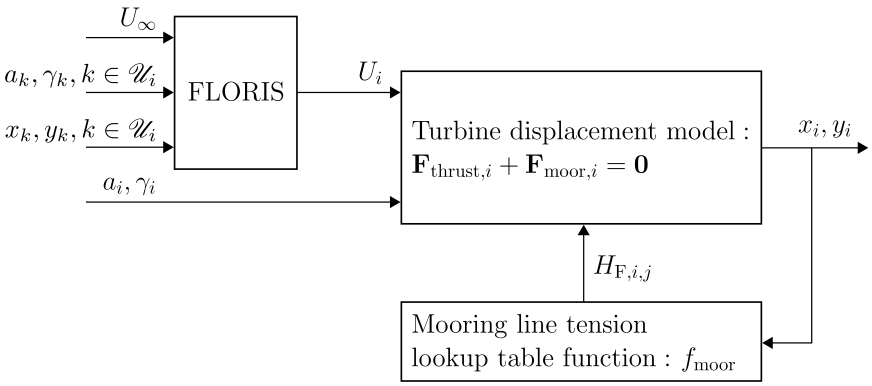

U_∞
$a_k, \gamma_k, k \in \mathscr{U}_i$
$x_k, y_k, k \in \mathscr{U}_i$
FLORIS
U_i
a_i, γ_i
Turbine displacement model :
$\mathbf{F}_{\text{thrust},i} + \mathbf{F}_{\text{moor},i} = \mathbf{0}$
x_i, y_i
$H_{\text{F},i,j}$
Mooring line tension
lookup table function : f_{moor}

Figure 3.3: Block diagram showing the computation modules of FOWFSim along with information transfer routes.

equation must be numerically solved with respect to x_i and y_i using a Newton-Raphson algorithm. Since the position and effective wind speed associated with a given turbine are independent of the positions and operating parameters of its downstream counterparts, Eq. (3.11) is solved using a sweeping algorithm starting from the leading turbine to the trailing machine.

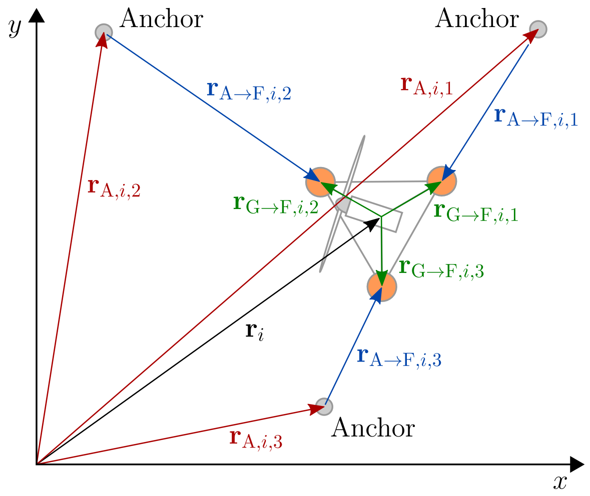
Similar to power production, aerodynamic thrust forces are also modeled according to actuator disc theory [121]. The thrust force vector acting the rotor of turbine i is calculated as follows [72]:

$$\mathbf{F}_{\text{thrust},i} = \frac{1}{2} C_{\text{thrust},i} \rho A \left(U_i \cos \gamma_i \right)^2 \begin{bmatrix} \cos \gamma_i \\ \sin \gamma_i \end{bmatrix}. \tag{3.12}$$

The above equation computes the magnitude of the thrust force based on the wind velocity projected normal to a turbine's rotor, and then computes the thrust force vector components based on the yaw angle of the turbine's nacelle. The thrust coefficient $C_{\text{thrust},i}$ of turbine i is calculated as follows [72]:

$$C_{\text{thrust},i} = 4a_i \left(1 - a_i\right). \tag{3.13}$$

The net mooring line force acting on turbine i is calculated as a summation of



y
Anchor
Anchor
$\mathbf{r}_{\mathrm{A}\to\mathrm{F},i,2}$
$\mathbf{r}_{\mathrm{A},i,1}$
$\mathbf{r}_{\mathrm{A}\to\mathrm{F},i,1}$
$\mathbf{r}_{\mathrm{A},i,2}$
$\mathbf{r}_{\mathrm{G}\to\mathrm{F},i,2}$
$\mathbf{r}_{\mathrm{G}\to\mathrm{F},i,1}$
$\mathbf{r}_{\mathrm{G}\to\mathrm{F},i,3}$
$\mathbf{r}_i$
$\mathbf{r}_{\mathrm{A}\to\mathrm{F},i,3}$
$\mathbf{r}_{\mathrm{A},i,3}$
Anchor
x

Figure 3.4: Schematic demonstrating the position vectors that are used to describe the mooring system for turbine *i*.

all three connected mooring line forces as follows:

$$\mathbf{F}_{\mathrm{moor},i} = \sum_{j\in\mathscr{M}} \mathbf{F}_{\mathrm{moor},i,j}, \tag{3.14}$$

where the force $\mathbf{F}_{\mathrm{moor},i,j}$ exerted by mooring line j on turbine i (refer to Figure 3.2) is computed as follows:

$$\mathbf{F}_{\mathrm{moor},i,j} = -H_{\mathrm{F},i,j}\frac{\mathbf{r}_{\mathrm{A}\to\mathrm{F},i,j}}{L_{\mathrm{A}\to\mathrm{F},i,j}}. \tag{3.15}$$

The vectors that are relevant to mooring force calculations are described in Figure 3.4. The position vector $\mathbf{r}_{\mathrm{A}\to\mathrm{F},i,j}$ measures from the j^{th} anchor of turbine i to its j^{th} fairlead as follows:

$$\mathbf{r}_{\mathrm{A}\to\mathrm{F},i,j} = \mathbf{r}_i + \mathbf{r}_{\mathrm{G}\to\mathrm{F},i,j} - \mathbf{r}_{\mathrm{A},i,j}, \tag{3.16}$$

where $\mathbf{r}_{\mathrm{G}\to\mathrm{F},i,j}$ is a constant vector that symbolizes the location of the j^{th} fairlead of turbine i relative to the center of its floating platform, and $\mathbf{r}_{\mathrm{A},i,j}$ is also a constant vector that measures the location of the j^{th} anchor of turbine i relative to the origin of the coordinate system.

The horizontal tension in the j^{th} cable that is connected to turbine i is a function of the horizontal distance $L_{\text{A}\rightarrow\text{F},i,j}$ between the cable's respective anchor and fairlead, which is calculated as follows:

$$L_{\text{A}\rightarrow\text{F},i,j} = \|\mathbf{r}_{\text{A}\rightarrow\text{F},i,j}\|, \tag{3.17}$$

where $\|\cdot\|$ represents the Euclidean norm. The magnitude of the horizontal cable tension $H_{\text{F},i,j}$ is then obtained from the solution of the suspended cable problem defined in Section 3.2.5. Since platform motion in the z direction is neglected, the vertical distance between the anchor and fairlead of any mooring line remains unchanged. The only independent variable influencing $H_{\text{F},i,j}$ is therefore $L_{\text{A}\rightarrow\text{F},i,j}$. This characteristic is exploited and $H_{\text{F},i,j}$ is computed using a lookup table function as follows:

$$H_{\text{F},i,j} = f_{\text{moor}}\left(L_{\text{A}\rightarrow\text{F},i,j}\right). \tag{3.18}$$

The lookup table function $f_{\text{moor}}(\cdot)$ is constructed for each new set of mooring line properties prior to running wind farm simulations or conducting optimization studies.

3.2.5 Mooring line tensions

The horizontal tension generated within a mooring line is dependent upon the general shape of its cable, which is in turn a function of the location of its fairlead. Figure 3.5 identifies three different zones in which a fairlead may be located that determine the general shape of a mooring line cable. Zone 1 is active when the horizontal distance between the fairlead and anchor is small enough such that the corresponding cable is partially resting on the seabed while its suspended portion is vertical with no catenary profile. Zone 2 is active when the fairlead is located beyond the boundary of zone 1, where the cable is still partially resting on the seabed and its suspended portion possesses a catenary profile, up to a horizontal position where no portion of the cable rests on the seabed and the slope of the cable profile at the anchor is zero (*i.e.* no vertical force at the anchor). The blue line in Figure 3.5 is an example of a cable operating within zone 2. The green and orange lines represent the boundaries of zone 2. Zone 3 is then active when no portion

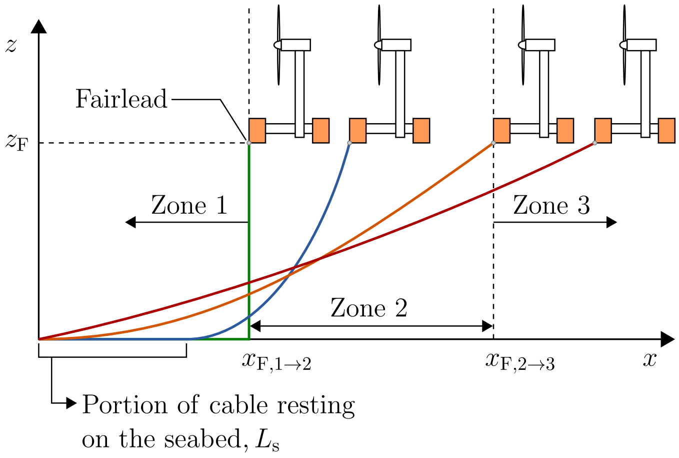


Figure 3.5: Schematic demonstrating the three zones in which mooring line fairleads may be located. The zone that is active determines the cable profile and tension.

of the cable rests on the seabed and the slope of the cable profile at the anchor is greater than zero, which results in vertical pull on the anchor. The red line in Figure 3.5 is an example of a cable operating within zone 3.

The horizontal tension in the cable is equal to zero when the fairlead is located within zone 1. Finding the cable tension when zones 2 and 3 are active requires solving the problem of an elastic cable suspended under its weight. Solutions to a similar problem have been presented by Jonkman [97] with no derivation. Since the solutions contained in the current work differed from those of Jonkman [97], details concerning the setup and derived expressions are presented. While zone 2 is active, the cable suspension problem is defined according to Figure 3.6a. The suspended cable portion (*i.e.* the blue line) is lifted by a vertical force V_F at the fairlead which counteracts the weight of the cable F_w. The horizontal tension throughout the catenary profile is uniform and equal to the fairlead horizontal force H_F. The cable portion resting on the seabed (*i.e* the orange line) is subject to a horizontal anchor force H_A, a static friction force F_s, and a horizontal reaction force of magnitude H_F exerted by the tension of the suspended portion of the cable. The resulting differential equations that describe the equilibrium of forces for this scenario are formed as follows:

$$\frac{dV}{ds} = \begin{cases} 0 & , \quad s < L_\mathrm{s} \\ \frac{w}{1+\frac{T}{A_\mathrm{c}E}} & , \quad s \geq L_\mathrm{s} \end{cases} , \tag{3.19}$$

$$\frac{dH}{ds} = \begin{cases} \frac{\mu_\mathrm{s} w}{1+\frac{H}{A_\mathrm{c}E}} & , \quad s < L_\mathrm{s} \\ 0 & , \quad s \geq L_\mathrm{s} \end{cases} , \tag{3.20}$$

where H, V, and T are the horizontal, vertical, and total tensions at any point along the cable, s is a general descriptor of arc length along the cable measured from the origin depicted in Figure 3.6a, w is the weight of the cable per unit length in ocean water, μ_s is the coefficient of static friction between the cable and seabed, and A_c and E are the cross-sectional area and elastic modulus of the cable, respectively. The total tension T is calculated as follows:

$$T = \sqrt{H^2 + V^2}. \tag{3.21}$$

If the fairlead is located within zone 3, as shown in Figure 3.6b, the cable assumes a fully catenary profile with uniform horizontal tension H_F throughout. The vertical fairlead force V_F must then counteract the weight of the cable F_w and the vertical anchor force V_A. The resulting differential equation describing force equilibrium in the vertical direction is therefore set up as follows:

$$\frac{dV}{ds} = \frac{w}{1+\frac{T}{A_\mathrm{c}E}}. \tag{3.22}$$

The solutions to these differential equations are obtainable analytically and, based on the zones defined in Figure 3.5, their solutions are grouped within $f_\mathrm{moor}(\cdot)$ as follows:

$$f_\mathrm{moor}(x_\mathrm{F}) = \begin{cases} 0 & , \quad x_\mathrm{F} \leq x_{\mathrm{F},1\to 2} \\ f_1(x_\mathrm{F}) & , \quad x_{\mathrm{F},1\to 2} < x_\mathrm{F} \leq x_{\mathrm{F},2\to 3} \\ f_2(x_\mathrm{F}) & , \quad x_\mathrm{F} > x_{\mathrm{F},2\to 3} \end{cases} , \tag{3.23}$$

where x_F is the horizontal position of the fairlead relative to the anchor, which is equal to $L_{\mathrm{A}\to\mathrm{F},i,j}$, $x_{\mathrm{F},1\to 2}$ and $x_{\mathrm{F},2\to 3}$ are the fairlead locations that separate the

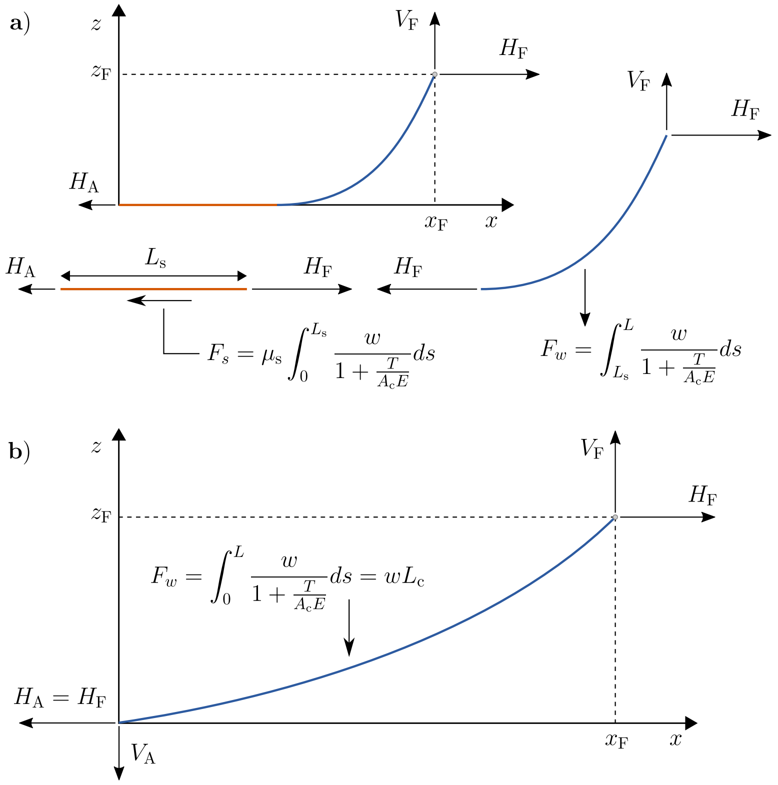

a)
z
V_{F}
z_{F}
H_{F}
V_{F}
H_{F}
H_{A}
x_{F}
x
H_{A}
L_{s}
H_{F}
H_{F}
$F_s = \mu_{\mathrm{s}} \int_0^{L_{\mathrm{s}}} \frac{w}{1+\frac{T}{A_{\mathrm{c}}E}} ds$
$F_w = \int_{L_{\mathrm{s}}}^{L} \frac{w}{1+\frac{T}{A_{\mathrm{c}}E}} ds$
b)
z
V_{F}
z_{F}
H_{F}
$F_w = \int_0^{L} \frac{w}{1+\frac{T}{A_{\mathrm{c}}E}} ds = wL_{\mathrm{c}}$
$H_{\mathrm{A}} = H_{\mathrm{F}}$
V_{A}
x_{F}
x

Figure 3.6: Schematic demonstrating the forces acting on different cable portions when the fairlead of the mooring line is located within (a) zone 2 and (b) zone 3.

different zones identified in Figure 3.5, and $f_1(\cdot)$ and $f_2(\cdot)$ are solutions to the suspended cable problems corresponding to zones 2 and 3, respectively.

First, the problems defined in Figure 3.6 are solved to yield the following expressions for the transition locations between the different zones (refer back to Figure 3.5 for a description of these locations):

$$x_{\mathrm{F},1\to 2} = L_{\mathrm{c}} - z_{\mathrm{F}}, \tag{3.24}$$

$$x_{\mathrm{F},2\to 3} = \frac{H_{\mathrm{F},2\to 3}}{w}\left[\frac{wL_{\mathrm{c}}}{A_{\mathrm{c}}E} + \sinh^{-1}\frac{wL_{\mathrm{c}}}{H_{\mathrm{F},2\to 3}}\right], \tag{3.25}$$

where L_{c} is the length of the unstretched cable, z_{F} is the vertical location of the fairlead relative to the anchor, which is constant in all simulations since vertical platform motion is neglected, and $H_{\mathrm{F},2\rightarrow3}$ is the horizontal cable tension when the fairlead is located at $x_{\mathrm{F},2\rightarrow3}$ and is calculated as follows:

$$H_{\mathrm{F},2\rightarrow3} = \frac{wL_{\mathrm{c}}}{2}\left[1-\left(\frac{z_{\mathrm{F}}}{L_{\mathrm{c}}}-\frac{wL_{\mathrm{c}}}{2A_{\mathrm{c}}E}\right)^{2}\right]\left(\frac{z_{\mathrm{F}}}{L_{\mathrm{c}}}-\frac{wL_{\mathrm{c}}}{2A_{\mathrm{c}}E}\right)^{-1}. \tag{3.26}$$

The function $f_1(\cdot)$, which outputs the horizontal fairlead force H_{F} while zone 2 is active, is obtained by solving the following system of equations for H_{F} and V_{F} using a Newton-Raphson algorithm:

$$x_{\mathrm{F}}-L_{\mathrm{s}} = \frac{H_{\mathrm{F}}}{w}\left(\frac{V_{\mathrm{F}}}{A_{\mathrm{c}}E}+\sinh^{-1}\frac{V_{\mathrm{F}}}{H_{\mathrm{F}}}\right), \tag{3.27}$$

$$z_{\mathrm{F}} = \frac{1}{w}\left\{\frac{V_{\mathrm{F}}^{2}}{2A_{\mathrm{c}}E}-H_{\mathrm{F}}\left[1-\sqrt{1+\left(\frac{V_{\mathrm{F}}}{H_{\mathrm{F}}}\right)^{2}}\right]\right\}, \tag{3.28}$$

where L_{s} is the stretched length of the cable portion that is resting on the seabed (*i.e.* the length of the orange line in Figure 3.6a) and is calculated as follows:

$$L_{\mathrm{s}} = L_{\mathrm{c}}-\frac{V_{\mathrm{F}}}{w}+\frac{\left(1+\frac{H_{\mathrm{F}}}{A_{\mathrm{c}}E}\right)^{3}-\left[\left(1+\frac{H_{\mathrm{F}}}{A_{\mathrm{c}}E}\right)^{2}-\frac{2\mu_{\mathrm{s}}w}{A_{\mathrm{c}}E}x_{\mathrm{s}}\right]^{\frac{3}{2}}}{\frac{3\mu_{\mathrm{s}}w}{A_{\mathrm{c}}E}}-x_{\mathrm{s}}. \tag{3.29}$$

The variable x_{s} is the distance from the start of the catenary portion of the cable (*i.e.* where the orange and blue lines meet in Figure 3.6a) to the location along the flat cable portion where the horizontal tension is minimal. This minimal tension point either corresponds to the position of the anchor, or some intermediate location along the flat cable portion where the tension reaches zero due to the opposition of F_{s} against H_{F}. The distance x_{s} is therefore calculated as follows:

$$x_{\mathrm{s}} = \min\left[L_{\mathrm{c}}-\frac{V_{\mathrm{F}}}{w},\frac{H_{\mathrm{F}}}{w\mu_{\mathrm{s}}}\left(1+\frac{H_{\mathrm{F}}}{2A_{\mathrm{c}}E}\right)\right] \tag{3.30}$$

The function $f_2(\cdot)$, which outputs the horizontal fairlead force H_{F} while zone 3 is active, is obtained by solving the following system of equations for H_{F} and V_{F}

using a Newton-Raphson algorithm:

$$x_{\mathrm{F}} = \frac{H_{\mathrm{F}}}{w}\left(\frac{wL_{\mathrm{c}}}{A_{\mathrm{c}}E} + \sinh^{-1}\frac{V_{\mathrm{F}}}{H_{\mathrm{F}}} - \sinh^{-1}\frac{V_{\mathrm{F}} - wL_{\mathrm{c}}}{H_{\mathrm{F}}}\right), \tag{3.31}$$

$$z_{\mathrm{F}} = \frac{L_{\mathrm{c}}}{A_{\mathrm{c}}E}\left(V_{\mathrm{F}} - \frac{wL_{\mathrm{c}}}{2}\right) + \frac{H_{\mathrm{F}}}{w}\left[\sqrt{1+\left(\frac{V_{\mathrm{F}}}{H_{\mathrm{F}}}\right)^2} - \sqrt{1+\left(\frac{V_{\mathrm{F}} - wL_{\mathrm{c}}}{H_{\mathrm{F}}}\right)^2}\right]. \tag{3.32}$$

3.3 Optimization problem

The goal of optimization in the context of the current study is to determine the sets of axial induction factors **a** and yaw angles γ that maximize wind farm efficiency η_{farm}. Obtaining η_{farm} for given **a** and γ vectors requires solving Eq. (3.11) to determine the position of each floating turbine. This process involves $2N$ implementations of a Newton-Raphson algorithm to find the x and y positions of N platforms for each calculation of η_{farm}. To improve computational efficiency, the Newton-Raphson algorithm is removed from the function η_{farm} by instead defining a nonlinearly constrained optimization problem as follows:

$$\begin{aligned} &\min_{\mathbf{a},\gamma,\mathbf{x},\mathbf{y}} 1 - \eta_{\mathrm{farm}}(\mathbf{a},\gamma,\mathbf{x},\mathbf{y}), \\ &\text{subject to} \begin{cases} 0.2 \leq a_i \leq 0.4, \\ -20\,\mathrm{deg} \leq \gamma_i \leq 20\,\mathrm{deg}, \\ x_{i,\min} \leq x_i \leq x_{i,\max}, \\ y_{i,\min} \leq y_i \leq y_{i,\max}, \\ \mathbf{F}_{\mathrm{thrust},i} + \mathbf{F}_{\mathrm{moor},i} = 0, \end{cases} \end{aligned} \tag{3.33}$$

where the cost function $1 - \eta_{\mathrm{farm}}$ represents the wind farm efficiency deficit, and $x_{i,\min}$, $y_{i,\min}$, $x_{i,\max}$, and $y_{i,\max}$ are the bounds of the optimization variables x_i and y_i. These bounds are determined uniquely for each turbine assuming ±200 m of movable range surrounding their neutral positions. This range is simply selected to be large enough such that it exceeds the actual movable range of the floating wind turbines. The bounds on γ_i are selected to conservatively maintain nacelle yaw below ±25 deg. Yaw offset beyond this range has been attributed to excessive fatigue loads [59]. The upper bound on a_i is set to uphold the validity of actuator disc theory, while the lower bound prevents excessive power de-rating.

The minimization problem defined in Eq. (3.33) is solved using the MATLAB optimization toolbox based on Sequential Quadratic Programming (SQP). This optimization framework forms a quadratic approximation of the objective function at each iteration, while at the same time linearizing constraint equations and incorporating them into the objective function using Lagrange multipliers [142]. This quadratic problem is then solved using an efficient matrix decomposition technique [142]. Along with interior-point methods, SQP is considered the most effective technique for solving nonlinearly constrained optimization problems [142]. All axial induction factors are initialized according to greedy operation (*i.e.* $\mathbf{a}_{\text{init}} = \mathbf{a}_{\text{greedy}} = \mathbf{1/3}$). To initialize the yaw angle vector γ_{init}, a simple algorithm that identifies which pairs of turbines are strongly coupled aerodynamically, and then ensures that their nacelles are initially yawed in opposite directions, was developed. Initial turbine positions $\mathbf{x}_{\text{init}}$ and $\mathbf{y}_{\text{init}}$ are obtained by solving FOWFSim using $\mathbf{a}_{\text{init}}$ and γ_{init}.

3.4 Results and discussion

The current section first assesses the impacts of various floating wind farm design parameters on the potential of YITuR. These parameters include anchor location and mooring line cable length (Section 3.4.1), mooring system orientation (Section 3.4.2), and wind farm size and inter-turbine spacing (Section 3.4.3). For these analyses, a wind-aligned row of floating wind farms is simulated and optimized. Then, the increase in wind farm efficiency obtained using YITuR is computed for various wind directions and wind farm sizes (Section 3.4.4). It is emphasized that the results presented in this section are obtained via optimal YITuR operation, which is implemented by solving Eq. (3.33) for each given set of wind farm properties and wind conditions in order to acquire optimal $\mathbf{a}$ and γ values.

The wind turbines and floating platforms that are simulated are based upon designs outlined by Jonkman *et al.* [96] and Robertson *et al.* [152]; parametric details of these designs are listed in Table 3.1. One important characteristic of

Table 3.1: Properties of the floating wind turbine detailed by Jonkman *et al.* [96] and Robertson *et al.* [152].

Baseline wind turbine properties	
Rotor radius, R	63 m
Power conversion efficiency, η_P	0.768
Power coefficient data fitting parameter, p_P	1.88
Baseline floating platform properties	
Distance from platform center to fairleads, $L_{\mathrm{G}\to\mathrm{F}}$	40.87 m
Mooring line properties	
Anchor distance from neutral platform, $L_{\mathrm{A}\to\mathrm{G,neutral}}$	837.6 m
Vertical distance between anchors and fairleads, z_{F}	186 m
Cable length of baseline mooring system, L_{c}	835 m
Cable diameter, D_{c}	7.66 cm
Cable weight per unit length in water, w	1065.7 N/m
Cable tension per meter of elongation, $A_{\mathrm{c}}E$	753.6×10^6 N
Static friction coefficient between cable and seabed, μ_{s}	1.0

the simulated floating platform is its baseline orientation relative to the incoming wind direction; this orientation is shown in Figure 3.7. Mooring line 1 is oriented 60 deg counterclockwise from the positive x axis, and subsequent cables are oriented 120 deg counterclockwise from their preceding line. This figure will be alluded to in Section 3.4.2 when mooring system orientation changes are discussed.

3.4.1 Effects of anchor location and cable length

It is first necessary to determine what length of mooring line cables and what anchor locations will permit adequate efficiency gains from YITuR operation. Since the distance between an anchor and its respective floating platform influences the length of the connecting mooring line, the cable length factor α_{cable} is defined to parametrize mooring line length L_{c} as follows:

$$L_{\mathrm{c}} = L_{\mathrm{c,min}} + \alpha_{\mathrm{cable}} \left(L_{\mathrm{c,max}} - L_{\mathrm{c,min}} \right), \tag{3.34}$$

where the smallest permissible cable length $L_{\mathrm{c,min}}$ for any anchor location is defined as follows:

$$L_{\mathrm{c,min}} = \sqrt{x_{\mathrm{F,neutral}}^2 + z_{\mathrm{F}}^2}, \tag{3.35}$$

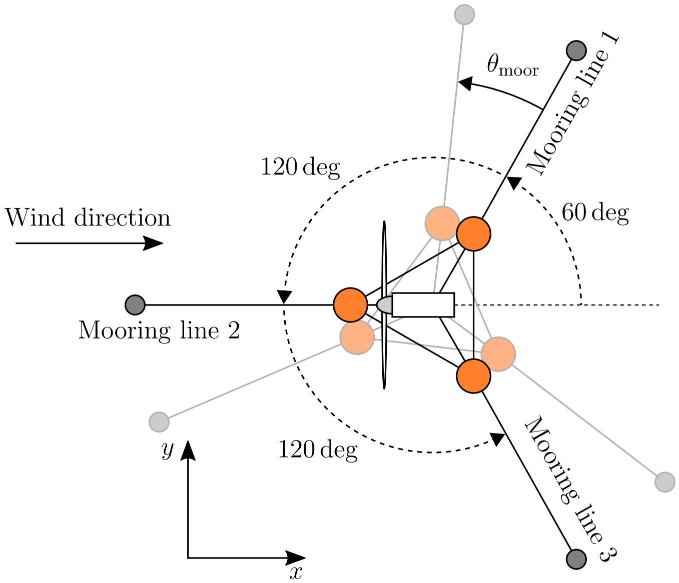


Figure 3.7: Schematic demonstrating the baseline orientation of a floating platform and mooring system relative to the incoming wind direction. The orientation of a mooring system that is rotated by a value θ_{moor} is also presented in the faded diagram.

and the largest permissible cable length $L_{\text{c,max}}$ for any anchor location is defined as follows:

$$L_{\text{c,max}} = x_{\text{F,neutral}} + z_{\text{F}}. \tag{3.36}$$

The constant $x_{\text{F,neutral}}$ is the distance from anchor to fairlead when the respective floating platform is in its neutral position (*i.e.* no aerodynamic loads are present), and is calculated as follows:

$$x_{\text{F,neutral}} = L_{\text{A}\rightarrow\text{G,neutral}} - L_{\text{G}\rightarrow\text{F}}, \tag{3.37}$$

where $L_{\text{A}\rightarrow\text{G,neutral}}$ is the distance between any anchor to the center of gravity of its respective floating platform under neutral conditions, and $L_{\text{G}\rightarrow\text{F}}$ is the distance between the center of gravity of a floating platform to any of its fairleads. The length $L_{\text{c,min}}$ is small enough such that the unstretched cable follows a straight line from anchor to fairlead, while $L_{\text{c,max}}$ is a large enough length such that the cable is suspended vertically from the fairlead until it contacts the seabed. The cable length factor therefore falls within the range $0 \leq \alpha_{\text{cable}} \leq 1$ and linearly specifies the length of a cable such that $L_{\text{c,min}} \leq L_{\text{c}} \leq L_{\text{c,max}}$.

Figure 3.8 plots the relative gain in the efficiency $\Delta\eta_{\text{farm,rel}}$ of an optimized two-turbine floating wind farm as a function of different anchor placement distances $L_{\text{A}\to\text{G,neutral}}$ and cable length factors α_{cable}. The general trends in this figure lead to the following important point:

Observation 1. *Lengthening mooring line cables while keeping anchor locations fixed increases the effectiveness of YITuR operation.*

For example, with $L_{\text{A}\to\text{G,neutral}} = 800\,\text{m}$, increasing α_{cable} from 0.55 to 0.8 raises $\Delta\eta_{\text{farm,rel}}$ from 10.0 to 31.8 %. This finding is explained by the simple fact that if mooring lines are not sufficiently long, then floating platforms cannot be relocated far enough in the crosswind direction for YITuR to be fully effective in reducing wake overlap. Figure 3.9 shows the positions of the two floating wind turbines for two scenarios; $\alpha_{\text{cable}} = 0.55$ and $\alpha_{\text{cable}} = 0.95$, with $L_{\text{A}\to\text{G,neutral}} = 900\,\text{m}$ in both cases. When $\alpha_{\text{cable}} = 0.55$, the upstream turbines is shifted by 24.0 m in the negative y direction and the downstream machine is relocated by 39.4 m in the positive y direction. The total crosswind offset between their axes of rotation is therefore 63.4 m, which is approximately equal to half of one rotor diameter of 126 m, and a relative wind farm efficiency gain of 10.0 % is achieved. Raising α_{cable} to 0.95 increases this crosswind offset to 145.3 m, which exceeds a single rotor diameter and generates a larger relative efficiency gain of 34.6 %.

The following point may also be drawn from the trends depicted in Figure 3.8:

Observation 2. *Placing mooring system anchors farther from floating turbine neutral positions increases the effectiveness of YITuR operation.*

Fixing $\alpha_{\text{cable}} = 0.8$, increasing $L_{\text{A}\to\text{G,neutral}}$ from 200 to 900 m raises $\Delta\eta_{\text{farm,rel}}$ from 16.3 to 32.4 %. This effect is attributed to the nonlinearity of mooring line tensions. The horizontal and vertical components of mooring line tension for a sample cable are plotted in Figure 3.10 as a function of the horizontal distance x_{F} between its anchor and fairlead. Notice that there exists a value of x_{F} at which the cable tensions begin to rise dramatically; this point corresponds to $x_{\text{F},2\to3}$, which is the fairlead

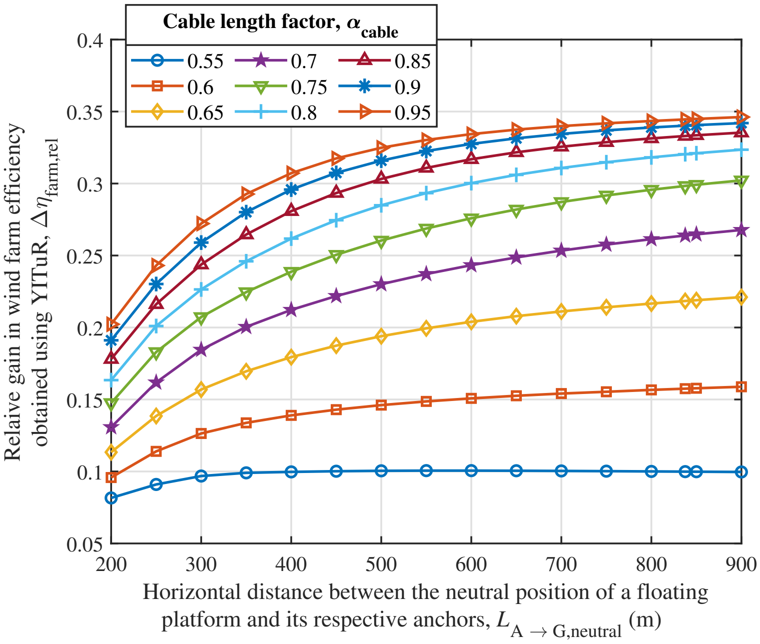


Figure 3.8: Relative gain in wind farm efficiency using YITuR operation with different mooring line length factors α_{cable} and anchor distances from turbine neutral positions $L_{\text{A}\rightarrow\text{G,neutral}}$. Simulation properties: 1×2 wind farm size, free stream wind speed $U_\infty = 8\,\text{m/s}$, inter-turbine spacing of $7D$, baseline mooring system orientation $\theta_{\text{moor}} = 0\,\text{deg}$.

location that begins to fully lift the cable off of the seabed. For the cable plotted in Figure 3.10, $x_{\text{F},2\rightarrow3}$ occurs at 902.9 m. This dramatic rise in cable tension is the limiting factor for YITuR, since it leads to a large restoring mooring line force that counters aerodynamic loads and prevents further platform displacement.

Placing an anchor farther away from its respective platform while maintaining α_{cable} constant allows for a larger range of platform motion before x_{F} reaches $x_{\text{F},2\rightarrow3}$. Figure 3.11 demonstrates this concept by showing the profiles of two mooring lines before and after platform displacement. Let the anchor in system a) be placed closer to the neutral position of its respective platform in comparison to that of system b). Let both of these turbines then be moved to the right by an equal distance such that the mooring line from system a) reaches its $x_{\text{F},2\rightarrow3}$ value (which corresponds to the orange line in Figure 3.11a). The turbine from system a) has

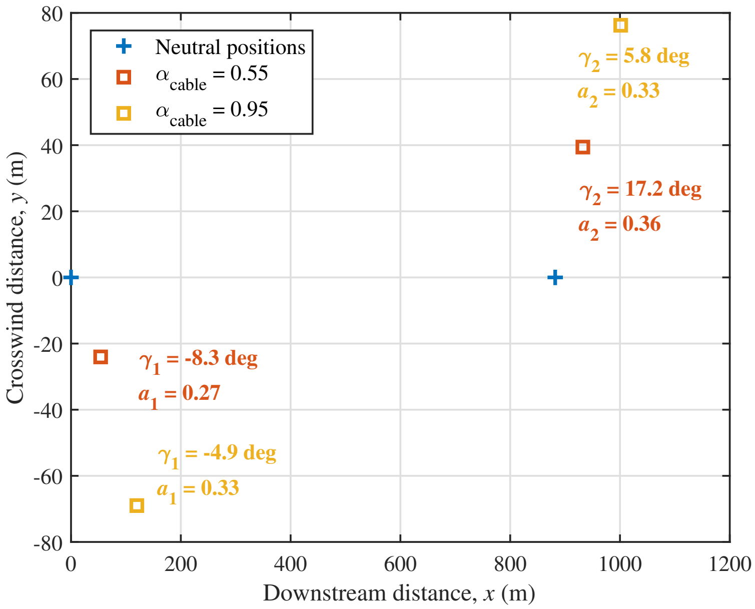


Figure 3.9: Floating wind turbine positions corresponding to optimization results presented in Figure 3.8. Simulation properties: 1×2 wind farm size, free stream wind speed $U_\infty = 8\,\text{m/s}$, inter-turbine spacing of $7D$, baseline mooring system orientation $\theta_{\text{moor}} = 0\,\text{deg}$, distance from anchors to turbine neutral positions $L_{\text{A}\rightarrow\text{G,neutral}} = 900\,\text{m}$.

now reached the point where its cable is being fully lifted off of the seabed, and thus the restoring force that counteracts further displacement begins to rise dramatically. The cable from system b), however, still has a large portion of cable resting on the seabed (refer to the flat portion of the orange line in Figure 3.11b), and is therefore far from its $x_{\text{F},2\rightarrow3}$ value. As a result, the turbine from system b) has additional range of motion toward the right before its cable tension begins to increase sharply. This turbine will therefore relocate farther from its neutral position before mooring line forces equalize with aerodynamic loads.

It may be concluded from the above findings that, in order for YITuR to be an effective means of wind farm control, anchors must be placed sufficiently far from the neutral positions of their respective platforms and cables must be adequately long. These characteristics will ensure a range of platform motion that permits

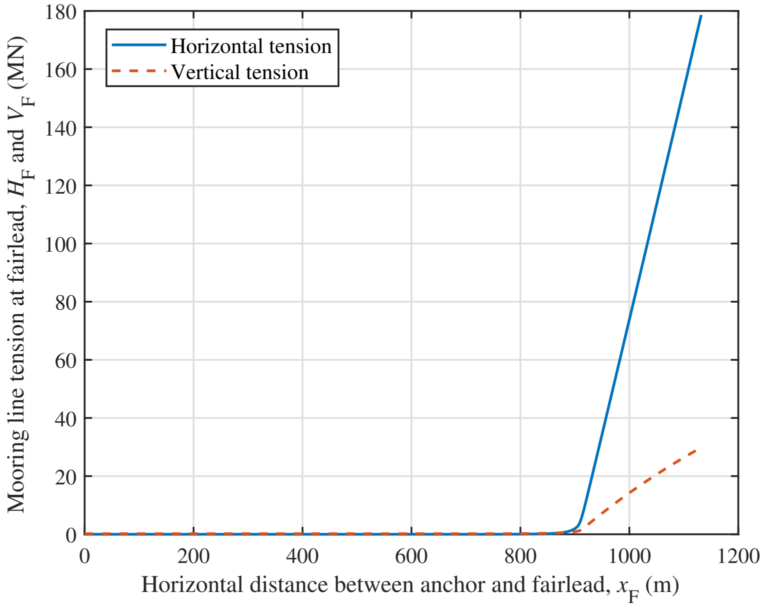


Figure 3.10: Horizontal and vertical components of cable tension corresponding to a mooring line of length $L_c = 925\,\text{m}$ with distance from anchor to platform neutral position $L_{A \to G,\text{neutral}} = 837.6\,\text{m}$.

significant reduction of wake overlap. However, this range of motion must not be excessive since there may be safety and practicality concerns associated with massive floating structures freely moving within large radii.

3.4.2 Effects of mooring system orientation

Another question of interest is whether the nonlinearity of mooring line forces alters the potential of YITuR if mooring systems are oriented differently. The angle θ_{moor} is used to denote the counterclockwise rotation of a mooring system relative to the baseline design described in Figure 3.7. The faded platform, cables, and anchors in this diagram represent the rotated mooring system. Since a rotation of 120 deg results in mooring system orientation that is identical to that of the baseline, only the range $0\,\text{deg} \leq \theta_{\text{moor}} \leq 120\,\text{deg}$ is considered.

Figure 3.12 plots the relative gain in wind farm efficiency $\Delta\eta_{\text{farm,rel}}$ resulting from YITUR operation with different values of θ_{moor} and α_{cable}. Similar to Sec-

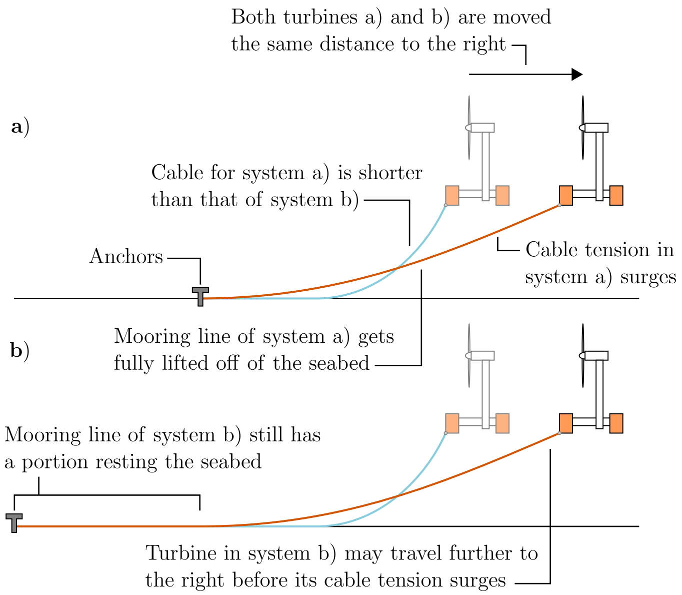

Both turbines a) and b) are moved
the same distance to the right
a)
Cable for system a) is shorter
than that of system b)
Anchors
Cable tension in
system a) surges
b)
Mooring line of system a) gets
fully lifted off of the seabed
Mooring line of system b) still has
a portion resting the seabed
Turbine in system b) may travel further to
the right before its cable tension surges

Figure 3.11: Schematic demonstrating the effect of increased anchor distance from floating platform neutral position.

tion 3.4.1, the simulated wind farm consists of two floating wind turbines aligned with the wind. It is observed that regardless of the value of α_{cable}, YITuR is only effective in significantly raising wind farm efficiency in the approximate ranges of $0\,\text{deg} \leq \theta_{\text{moor}} \leq 20\,\text{deg}$ and $100\,\text{deg} \leq \theta_{\text{moor}} \leq 120\,\text{deg}$. For instance, with $\alpha_{\text{cable}} = 0.8$, rotating the baseline mooring system counterclockwise by 20 deg reduces $\Delta\eta_{\text{farm,rel}}$ from 32.0 to 21.3 %. This result implies a third important point:

Observation 3. *The baseline mooring system orientation shown in Figure 3.7 is optimal for YITuR, and variations from this orientation exceeding* ±20 deg *will yield significant reduction in wind farm efficiency gains.*

The trends in Figure 3.12 are the direct result of mooring line tension nonlinearity. Consider the floating wind turbines and mooring systems shown in Figure 3.13. With the wind blowing from the left, the mooring system in Figure 3.13a is oriented similar to the baseline from Figure 3.7 while the mooring system in Figure 3.13b is rotated 60 deg counterclockwise relative to the baseline orientation. The platform locations shown correspond to zero yaw offset in both cases. Under this setting,

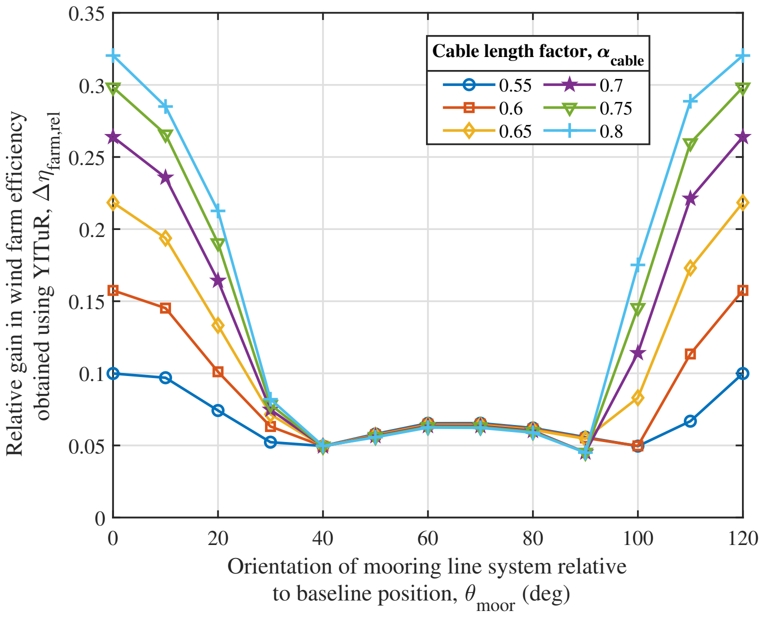


Figure 3.12: Relative gain in wind farm efficiency using YITuR operation with different cable length factors α_{cable} and mooring system orientations θ_{moor}. Simulation properties: 1×2 wind farm size, free stream wind speed $U_\infty = 8\,\text{m/s}$, inter-turbine spacing of $7D$, distance from anchors to platform neutral positions $L_{\text{A}\to\text{G,neutral}} = 837.6\,\text{m}$.

mooring lines 1 and 3 of system a) are far from zone 3 operation (refer back to Figure 3.5 for discussions on mooring line zones) since their respective fairleads and anchors are in close proximity. These cables therefore do not exert large restoring forces even if yaw offset were introduced and the platform were to shift in the crosswind direction. Additionally, although mooring line 2 of system a) is close to zone 3 operation, it exerts zero force in the crosswind direction. As a result, if yaw offset were introduced, the floating turbine in system a) must traverse a significant distance in the crosswind direction for the tension in cable 2 to equalize against aerodynamic forces.

The same characteristic does not hold true for the mooring system in Figure 3.13b. Due to its altered orientation, cables 1 and 2 of system b) operate closer to zone 3 and their horizontal tension components will rise sharply as their fairleads move farther away from their anchors. Furthermore, these two cables are

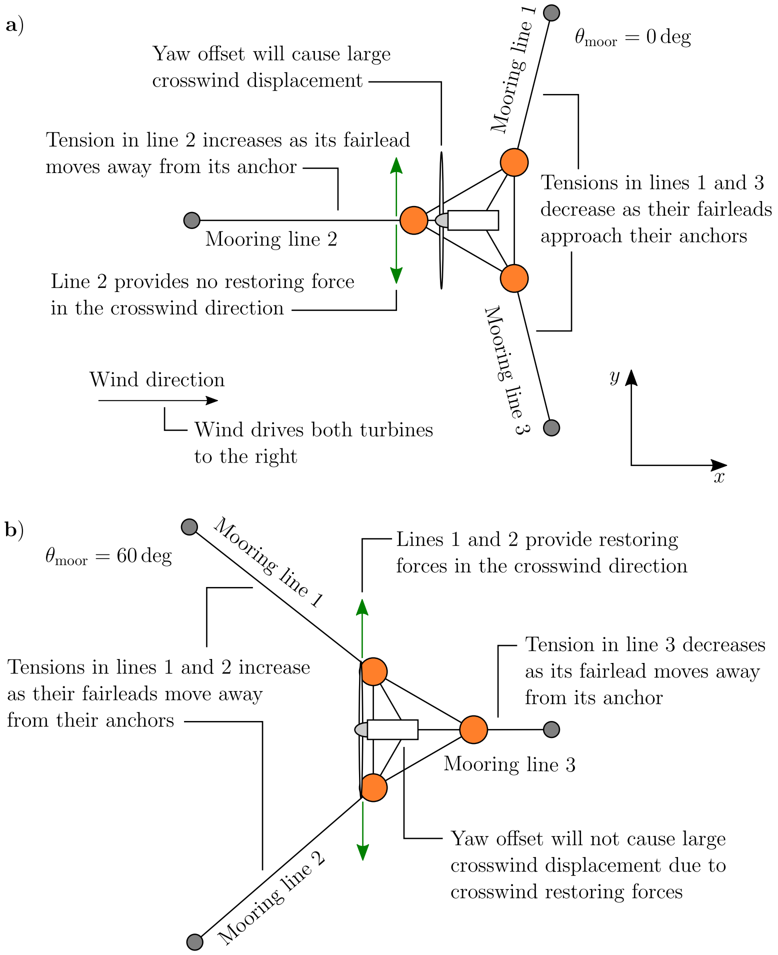

a)
$\theta_{\text{moor}} = 0\,\text{deg}$
Mooring line 1
Yaw offset will cause large crosswind displacement
Tension in line 2 increases as its fairlead moves away from its anchor
Tensions in lines 1 and 3 decrease as their fairleads approach their anchors
Mooring line 2
Line 2 provides no restoring force in the crosswind direction
Mooring line 3
Wind direction
Wind drives both turbines to the right
y
x
b)
$\theta_{\text{moor}} = 60\,\text{deg}$
Mooring line 1
Lines 1 and 2 provide restoring forces in the crosswind direction
Tensions in lines 1 and 2 increase as their fairleads move away from their anchors
Tension in line 3 decreases as its fairlead moves away from its anchor
Mooring line 3
Yaw offset will not cause large crosswind displacement due to crosswind restoring forces
Mooring line 2

Figure 3.13: Schematic demonstrating the effects of mooring system orientation on the crosswind mobility of floating offshore wind turbines.

oriented such that they exert restoring forces in the crosswind direction even without yaw offset. Consequently, if yaw misalignment were introduced, the floating turbine from system b) would have a smaller range of mobility in comparison to that of system a) before mooring line forces equalize with aerodynamic loads. Figure 3.14 plots the locations of the floating turbines simulated in Figure 3.12 at different mooring system orientations. Increasing θ_{moor} from 0 to 20 deg reduces the optimal crosswind offset between the two turbines from 139.6 to 97.7 m (or 1.1D to 0.8D). Further raising θ_{moor} to 40 deg decreases this crosswind offset to

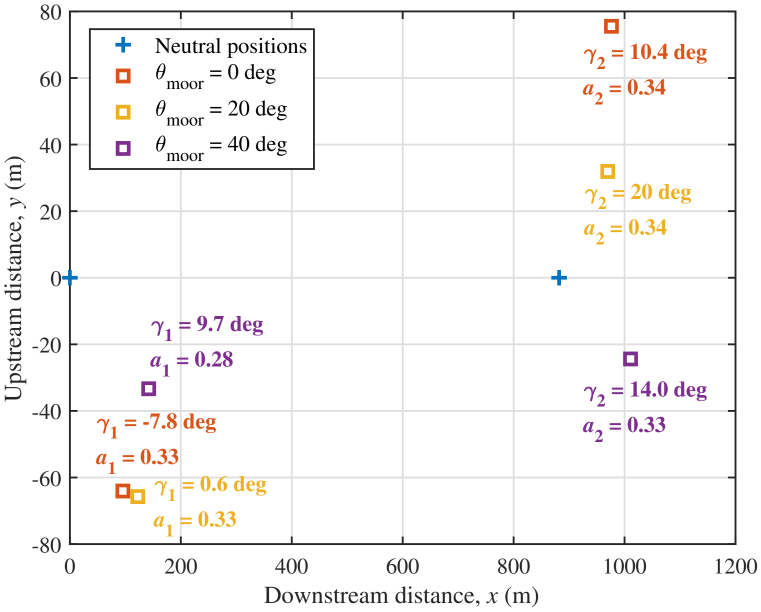


Figure 3.14: Floating wind turbine positions corresponding to optimization results presented Figure 3.12. Simulation properties: 1×2 wind farm size, free stream wind speed $U_\infty = 8\,\mathrm{m/s}$, inter-turbine spacing of $7D$, distance from anchors to platform neutral positions $L_{\mathrm{A}\to\mathrm{G,neutral}} = 837.6\,\mathrm{m}$, cable length factor $\alpha_{\mathrm{cable}} = 0.8$ (*i.e.* cable length $L_{\mathrm{c}} = 950\,\mathrm{m}$).

8.95 m (or $0.1D$).

Another observation from Figure 3.14 is that bias appears in the crosswind positioning of floating platforms as θ_{moor} deviates from zero. At $\theta_{\mathrm{moor}} = 20\,\mathrm{deg}$ for instance, turbine 1 maintains an insignificant yaw offset angle of 0.6 deg yet is shifted by 65.7 m in the $-y$ direction. Additionally, with $\theta_{\mathrm{moor}} = 40\,\mathrm{deg}$, both turbine nacelles are yawed counterclockwise yet they are displaced in the $-y$ direction. Clearly, adjusting the mooring system orientation offsets the crosswind steady-state positions of the floating platforms even with zero yaw misalignment. Introducing yaw offset then repositions turbines relative to these biased steady-state locations. This outcome is also attributed to the nonlinearity of mooring line forces. Figure 3.15 shows the mooring system orientation corresponding to $\theta_{\mathrm{moor}} = 20\,\mathrm{deg}$. As wind drives the floating platform in the downstream direction, the tensions in both mooring lines 1 and 3 decrease since their respective anchors move closer to

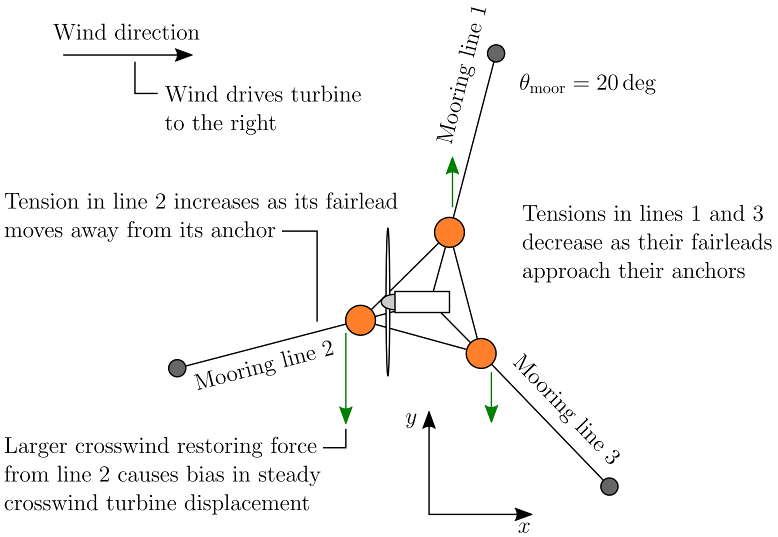

Wind direction
Wind drives turbine to the right
Mooring line 1
$\theta_{\text{moor}} = 20\,\text{deg}$
Tension in line 2 increases as its fairlead moves away from its anchor
Tensions in lines 1 and 3 decrease as their fairleads approach their anchors
Mooring line 2
Mooring line 3
Larger crosswind restoring force from line 2 causes bias in steady crosswind turbine displacement
y
x

Figure 3.15: Schematic demonstrating the origin of bias in the steady-state displacements of a floating offshore wind turbines when mooring system orientations deviate from the baseline shown in Figure 3.7.

their fairleads; thus, these mooring lines diverge from zone 3 operation. Mooring line 2, however, approaches zone 3 operation and experiences a significant increase in tension which, due to its orientation, exerts a force on the floating platform in the $-y$ direction. This force introduces the bias which is evident in Figure 3.14.

The above observations indicate that care must be given to the orientation of floating turbine mooring systems if YITuR (or any turbine repositioning concept) is to be employed for wind farm control. The orientation of a mooring system places limits on the mobility of floating platforms, which diminishes the potential of any turbine repositioning concept to maximize wind farm efficiency. This point is addressed quantitatively in Section 3.4.4, where the effect of wind direction on the potential of YITuR is examined.

3.4.3 Effects of turbine spacing and wind farm size

The effect of altering the spacing between wind turbines on the potential of YITuR is also examined. Varying inter-turbine spacing alone is uninteresting however. Increased spacing allows for more time and space over which wakes re-energize and reduce their velocity deficits [115], hence raising wind farm efficiency and rendering any wind farm control concept redundant. Instead, the simultaneous impacts of spacing and wind farm size (*i.e.* the number of installed wind turbines) are studied to answer the following question; how beneficial is YITuR in terms of increasing the turbine-density of a wind farm without relinquishing efficiency?

Figure 3.16 plots the efficiency η_{farm} of a wind-aligned row of N wind turbines with various inter-turbine spacings under both greedy[1] and YITuR operation. The wind direction and mooring system orientation are defined according to the baseline system defined in Figure 3.7, and the mooring line length in these simulations is 925 m. The following significant conclusion may be derived from Figure 3.16:

Observation 4. *By introducing YITuR operation, a wind farm developer may increase the quantity of wind turbines within the same farm area by* $N-1$ *while approximately maintaining wind farm efficiency.*

For instance, a wind farm consisting of a row of $N = 5$ turbines placed $8D$ apart and operating with greedy settings has an efficiency $\eta_{\text{farm}} = 60.7\,\%$. If a single wind turbine were installed between each existing pair, which would increase the wind farm size to $N = 9$ and decrease inter-turbine spacing to $4D$ without requiring additional wind farm area, the efficiency of the new wind farm with YITuR operation would remain at $60.0\,\%$.

To explain this outcome, wind farm layouts corresponding to $N = 5$ with $8D$ spacing and $N = 9$ with $4D$ spacing are plotted in Figure 3.17. Since YITuR minimizes wake overlap by yawing and relocating adjacent floating wind turbines in opposite directions, it effectively doubles the downstream spacing between any two machines with significant wake interaction. Referring to Figure 3.17, the floating platforms corresponding to $N = 9$ are spaced $4D$ apart under neutral conditions. YITuR then shifts adjacent pairs in this set of turbines in opposite crosswind directions. The result is two new rows at approximately $y = \pm 60\,\text{m}$ along which adjacent turbines that are directly aligned with the wind are actually spaced $8D$ apart. As a result, this wind farm will have the same effective inter-turbine spacing and efficiency as one with $N = 5$ turbines under greedy operation, since these machines will also be spaced $8D$ apart.

[1] Refer to the discussion surrounding $C_{P,\max}$ in Section 3.2.2 for a description of greedy operation.

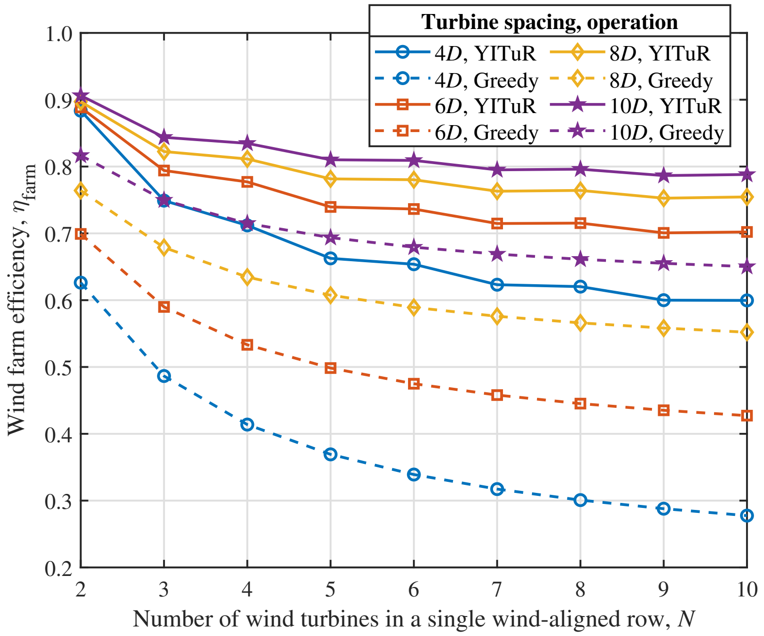


Figure 3.16: Efficiency η_{farm} of a wind-aligned row of N turbines obtained from greedy and YITuR operation for different inter-turbine spacings and wind turbine quantities N. Simulation properties: free stream wind speed $U_\infty = 8\,\mathrm{m/s}$, mooring line cable length $L_\mathrm{c} = 925\,\mathrm{m}$, distance from anchors to platform neutral positions $L_{\mathrm{A}\to\mathrm{G,neutral}} = 837.6\,\mathrm{m}$, baseline mooring system orientation $\theta_{\mathrm{moor}} = 0\,\mathrm{deg}$.

3.4.4 Wind farm efficiency studies

As a final assessment, the potential of YITuR for maximizing the efficiencies of several wind farms of different sizes subject to different wind directions is examined. All wind farms possess a grid layout with turbines spaced $7D$ apart in both x and y directions. All mooring systems are oriented according to the baseline design shown in Figure 3.7 with anchors located 837.6 m away from the neutral positions of their respective platforms and cables that are 925 m long. The wind speed in all simulations is 8 m/s.

Figure 3.18 shows polar plots of wind farm efficiencies under greedy and YITuR operation as functions of wind direction; each plot corresponds to a different farm size. For all wind farm sizes, greedy operation yields significantly low

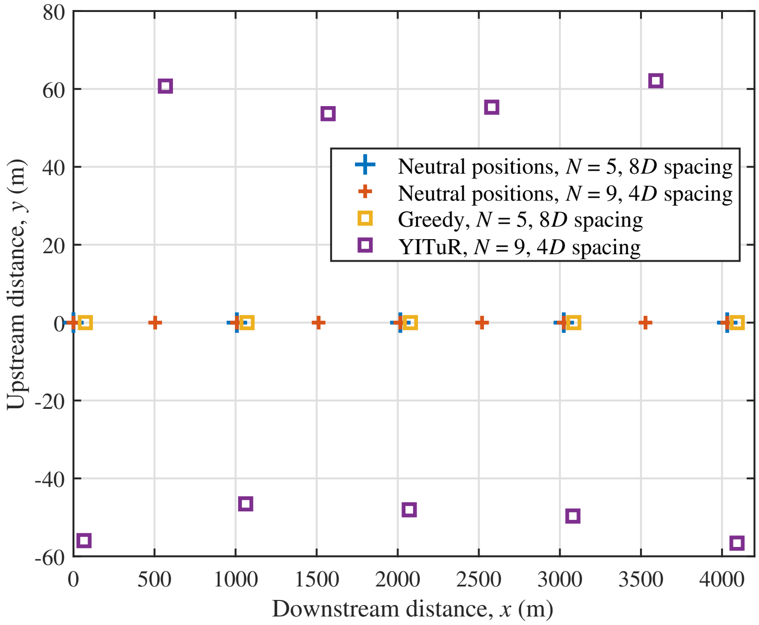


Figure 3.17: Floating wind turbine positions corresponding to optimization results presented Figure 3.16. Simulation properties: free stream wind speed $U_\infty = 8\,\mathrm{m/s}$, mooring line cable length $L_\mathrm{c} = 925\,\mathrm{m}$, distance from anchors to platform neutral positions $L_{\mathrm{A}\rightarrow\mathrm{G,neutral}} = 837.6\,\mathrm{m}$, baseline mooring system orientation $\theta_\mathrm{moor} = 0\,\mathrm{deg}$.

wind farm efficiencies near wind directions that are aligned with or diagonal to the farm's grid layout (*i.e.* 0, 45, 90 deg, etc.). At these angles, the distances between turbines that are directly aligned with the wind are reduced; hence, wakes do not sufficiently re-energize via turbulent diffusion prior to interacting with downstream rotors. When assessing the potential of YITuR operation for mitigating this problem, the following major point is established:

Observation 5. *YITuR operation is not equally effective at raising wind farm efficiency across all wind directions.*

At a wind direction of 0 deg for example, YITuR operation is significantly beneficial as it increases η_farm for a 2×2 wind farm from 73.3 to 89.3 %, which is a 21.8 % relative rise. At wind directions of 90, 180, and 270 deg however, relative efficiency gains obtained via YITuR operation are 7.1, 6.4, and 4.5 %.

This limitation is directly attributed to discussions from Section 3.4.2 regarding the effects of mooring system orientation on floating turbine mobility. Moreover, it is an extension of Observation 3 since changing the direction of the wind is tantamount to altering mooring system orientation relative to the incoming wind. Consequently, YITuR is only effective when the relative orientation between mooring systems and the wind is close to the baseline scenario shown in Figure 3.7. Wind farm developers hoping to benefit from YITuR must therefore identify the wind direction at which efficiency is minimal, and then orient mooring systems appropriately in order to maximize the potential of this wind farm control technique. Alternatively, mooring line orientation and YITuR operation could be considered in optimization processes that are used to design wind farm layouts.

As wind farm size is increased from 2×2 to 7×7, efficiencies under greedy operation and at a wind direction of 0 deg drop from 73.3 to 52.0 %. This outcome is expected; a larger quantity of wind turbines means that a higher portion of machines experience aerodynamic coupling. Losses resulting from the wake effect are therefore amplified. By the same logic, YITuR operation, or any wind farm control concept, must be more effective in raising efficiency for larger wind farm sizes; that is, a higher portion of machines experience the benefits of wind farm control. This outcome is observed in Figure 3.18. As wind farm size is increased from 2×2 to 7×7, the absolute gain in wind farm efficiency obtained via YITuR operation rises from 16.0 to 22.2 % at a wind direction of 0 deg. Relative gains in wind farm efficiency increase from 21.8 to 42.7 %; although these values are inflated due to the substantial drop in efficiency of the 7×7 wind farm under greedy operation.

3.5 Conclusions and recommendations for future research

The goal of this chapter has been to evaluate the steady-state potential of Yaw and Induction-based Turbine Repositioning (YITuR) as a control strategy for floating offshore wind farms. Optimization studies have shown that this potential is highly dependent upon wind farm design parameters. Mooring lines must be sufficiently long to permit significant platform displacement such that wake overlap along turbine rotors may be minimized. Concurrently, due to the nonlinearity of mooring

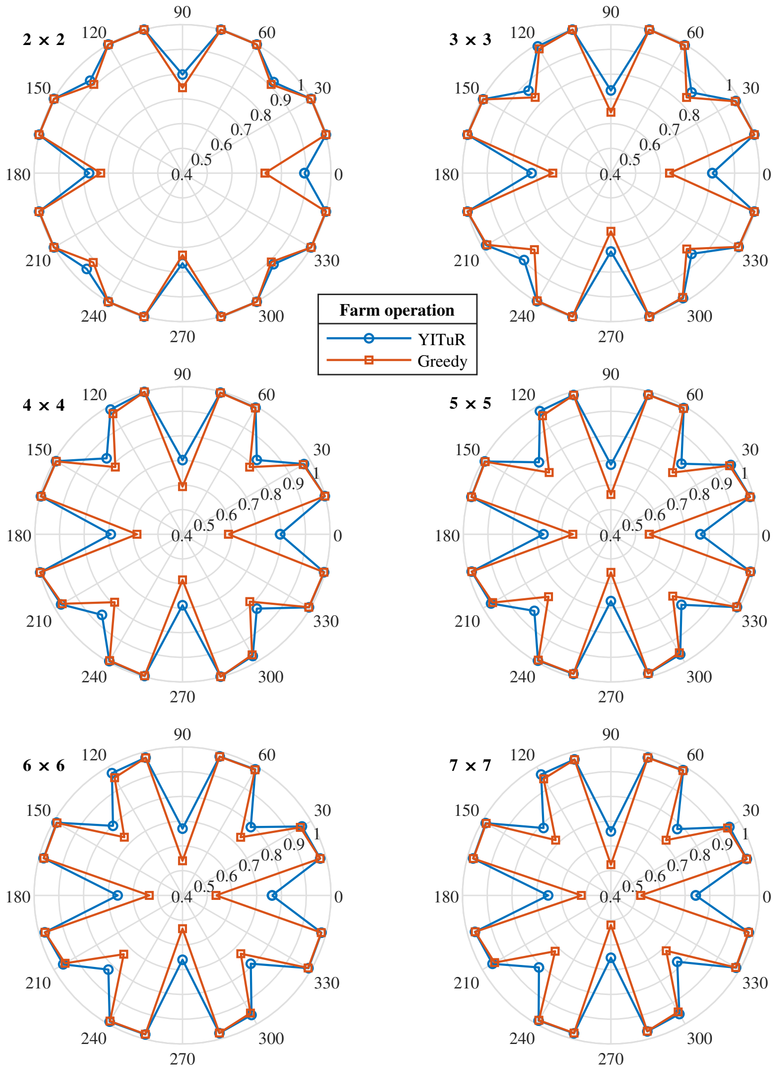


Figure 3.18: Efficiency η_{farm} of wind farms with grid layouts and different sizes under greedy and YITuR operation subject to different wind directions. The radial axes represent wind farm efficiency η_{farm} and the circumferential axes denote the wind direction with 0 deg indicating wind flow to the right. Simulation properties: free stream wind speed $U_\infty = 8\,\text{m/s}$, inter-turbine spacing of $7D$ in x and y directions, mooring line cable length $L_\text{c} = 925\,\text{m}$, distance from anchors to turbine neutral positions $L_{\text{A}\to\text{G,neutral}} = 837.6\,\text{m}$, baseline mooring system orientation $\theta_{\text{moor}} = 0\,\text{deg}$.

line tensions, anchors must be placed far enough from the neutral positions of floating turbines such that platforms may be adequately relocated before mooring line restoring forces increase exorbitantly. Platform mobility was also heavily influenced by mooring system orientation. The nonlinearity of mooring line tensions engendered unequal restoring forces at specific mooring system orientations which, in addition to limiting platform displacement, generated biases in the steady-state locations of floating turbines.

Studies of wind farm efficiency showed that, with specific mooring system orientations and anchor locations, and long enough mooring lines, YITuR operation could increase the efficiency of a 7×7 wind farm by $42.7\,\%$ relative to greedy operation. It was also shown that, due to mooring line tension nonlinearity, the effectiveness of YITuR operation is highly dependent upon the relative orientation between mooring systems and the direction of incoming wind. Wind farm developers considering any form of turbine repositioning as a wind farm control technique must therefore select appropriate mooring system orientations based on either comprehensive optimization studies or a predetermined wind direction at which YITuR operation is necessary.

Future research on YITuR should attempt to solve such optimization problems wherein not only optimal turbine operating parameters are of interest, but also optimal wind farm design parameters. It is also necessary to move beyond steady-state analyses to investigate the efficacy of dynamic optimization algorithms and control systems in implementing YITuR in real-time under the presence of turbulence and time-varying wind and wave conditions. Evaluating such controllers will require dynamic floating offshore wind farm models that simulate turbine motion along the ocean surface in addition to dynamic wake propagation.

Further, aspects of safety and practicality should be investigated. Specifically, the effects of increasing mooring line length on floating wind turbine stability should be examined quantitatively. Additionally, the effects of large platform displacement on maintenance procedures and crew safety should be investigated. Finally, the efficacy of YITuR operation in raising wind farm efficiency should be compared against the use of thrusters for relocating floating platforms. This analysis should consider the additional capital and maintenance costs of incorporating thrusters into floating platforms.

Chapter 4

A Low-fidelity Dynamic Wind Farm Model for Simulating Time-varying Wind Conditions and Floating Platform Motion

This chapter introduces a dynamic parametric wind farm model that is capable of simulating floating wind turbine platform motion coupled with wake transport under time-varying wind conditions. The simulator is named FOWFSim-Dyn as it is a dynamic extension of the previously developed steady-state *Floating Offshore Wind Farm Simulator* (FOWFSim). One-dimensional momentum conservation is used to model dynamic propagation of wake centerline locations and average velocities, while momentum recovery is approximated with the assumption of a constant temporal wake expansion rate. Platform dynamics are captured by treating a floating offshore wind farm as a distribution of particles that are subject to aerodynamic, hydrodynamic, and mooring line forces. The finite difference method is used to discretize the momentum conservation equations to yield a nonlinear state-space model. FOWFSim-Dyn is validated against steady-state experimental wind tunnel data obtained from the literature along with simulation results acquired from FAST. The model is shown to possess a satisfactory level of fidelity for engineering applications. Finally, dynamic simulations are conducted to ensure that time-varying predictions match physical expectations and intuition.

4.1 Introduction

Since the introduction of parametric wake models by Jensen [91] and Katić *et al.* [98], such wind farm simulators have served as essential tools for enhancing wind farm performance. This enhancement has been achieved via two distinct fields of study. The older of the two is *layout optimization*, wherein the optimal installation locations of wind turbines are computed with the objective of maximizing annual revenue [161]. Since such optimization problems are solved offline prior to wind farm construction, steady-state wake models have sufficed for estimating annual energy production. The field of study that has more recently experienced a surge in interest is *wind farm control*, which involves real-time wind turbine actuation for the purpose of manipulating the wind field to achieve some wind farm-level objective [102]. This ultimate goal may be efficiency maximization, or power output tracking with turbine load alleviation [107]. In either case, since actuators are adjusted in real-time, dynamic wake phenomena such as turbulence, transport delay, time-varying mean wind speed and direction, and floating platform motion (for deep-water offshore wind farms) are pertinent when evaluating controller performance.

Steady parametric wake models have been used successfully to raise wind farm efficiency in large eddy simulations (LES) [72] and field tests [59]. Further, in one instance, Gebraad *et al.* [69] reported no significant performance gains when using a dynamic wake model for wind farm control in contrast to using a steady wake model. Nonetheless, there are benefits associated with using dynamic wake models. First, traditional state and parameter estimation techniques may be used to adapt such models to time-varying wind conditions [69]. Second, low-fidelity dynamic wake models may be used to test controller robustness against time-varying wind conditions prior to dedicating time and resources to conducting high-fidelity simulations and field tests (as performed by Johnson and Fritsch [94] and Gebraad and van Wingerden [70]). Further research comparing wind farm control based on dynamic versus steady parametric models may reveal additional benefits. Such progress will only be possible, however, provided the availability of various dynamic wake models.

Reviews of wake modeling may be found in the works of Boersma *et al.* [24], Göçmen *et al.* [75], Vermeer *et al.* [179], and in our previous review article [102]. We will focus our current discussion on parametric dynamic wake models. The earliest of such models found application in power de-rating wind farm control research conducted by Gebraad and van Windgerden [70], Johnson and Fritsch [94], and Ahmad *et al.* [3]. Power de-rating involves reducing the thrust force exerted onto the wind by upstream wind turbines as a means of increasing the fluid momentum available to downstream machines [95]. Since this application involves neither wake deflection nor wind turbine relocation, wake dynamics in these studies were modeled using time-delays in computed steady-state incident wind speeds. These time-delays represented the duration required for changes in the wind field at some upstream turbine to propagate to downstream machines. This approach is valid as long as the wind direction remains constant, and wake centerline deflection and turbine relocation are not pertinent.

In order to account for transport delay of steered wakes, Gebraad and van Wingerden [67] developed the Flow Redirection and Induction Dynamics (FLORIDyn) model. Their approach involved representing flow within a wake using translating points that were initialized at each turbine and then transported downstream. Each point contained information regarding its corresponding turbine's operating parameters at the instant in time at which the point was initialized. Using this information, wake properties at the downstream location of the translating point were obtained using the Flow Redirection and Induction in Steady-state (FLORIS) wake model [72], which utilizes integral forms of mass and momentum conservation to compute downstream wake properties. More simply put, FLORIDyn transports steady-state wake characteristics computed with FLORIS in the free stream wind direction. Time varying wind direction and floating platform motion are not considered in FLORIDyn however.

In an alternative approach, Shapiro *et al.* [162] used the differential forms of mass and momentum conservation to simulate dynamic wake behavior. Local and convective wake accelerations in the free stream wind direction were described by material derivatives, and these accelerations were equated to force terms representing turbulent mixing and rotor thrust. The advantage of this modeling approach was that wake transport would be inherently captured by convective acceleration terms, thus eliminating the need for the translating points employed by the FLORIDyn model. Instead, all wake characteristics were functions of a fixed grid in the downstream direction. Shapiro *et al.* [163] later extended their model to capture wake redirection resulting from rotor yaw misalignment. Prandtl's lifting line theory was used to compute transverse wake velocities, shed circulation, and vortex properties immediately downstream of yawed rotors. Wake centerline deflection in the free stream wind direction was then computed by equating the material derivative of the centerline position to the transverse component of the wake velocity. These works do not capture floating platform motion or time-varying wind speed and direction.

Finally, Boersma *et al.* [25] developed the *Wind Farm Simulator* (WFSim), which is a control-oriented dynamic wake model based on the two-dimensional form of the unsteady turbulent Navier-Stokes equations. The major benefit of WFSim is that individual wake expansion and the interaction of multiple wakes are inherently captured by the mixing length turbulence model employed. In the previously discussed models, the rate of linear wake expansion was either estimated or assumed. Further, the previous models simulated flow behavior in regions with overlapping wakes by assuming that the effective kinetic energy deficit in the wind field is equal to the sum of deficits corresponding to all pertinent wakes. Despite its higher-fidelity, WFSim requires approximately 1000 sec of computation for 1000 sec of simulation in comparison to previously discussed models (several seconds of computation for 1000 sec of simulation). Moreover, WFSim does not model floating platform motion or time-varying wind speed and direction.

In the current chapter, we loosely follow the approach of Shapiro *et al.* [162], whereby partial differential equations are used to capture wake transport, and we develop a dynamic parametric wake model capable of simulating time-varying wind speed and direction, along with platform motion for floating offshore wind farms. The novelty of this chapter therefore includes the following: (i) additional terms in the wake momentum conservation equations to capture time-varying free stream wind velocity effects; and (ii) a coupled dynamic model that captures planar floating wind turbine motion in the presence of aerodynamic interaction. Our approach is physics-based with the rate of wake expansion as the only parametric assumption. This model serves as a dynamic extension of our previously developed steady-state tool [103], which was named the *Floating Offshore Wind Farm Simulator* (FOWFSim), and will henceforth be referred to as FOWFSim-Dyn. Fixed-foundation wind farms may also be modeled by simply deactivating turbine platform motion.

The remainder of this chapter is organized as follows: Section 4.2 provides a detailed mathematical description of FOWFSim-Dyn along with a discussion of its limitations. In Section 4.3, we perform a mesh convergence study and validate FOWFSim-Dyn's ability to predict floating platform motion and steady and dynamic wake transport. We finally conclude the chapter in Section 4.4 by listing potential research directions for enhancing FOWFSim-Dyn.

4.2 Mathematical model - FOWFSim-Dyn

This section details the mathematical formulation behind FOWFSim-Dyn. First, the problem setup, solver block diagram, and resulting equations of motion are presented in Section 4.2.1–Section 4.2.7. Finally, important assumptions and limitations pertaining to FOWFSim-Dyn are discussed in Section 4.2.8.

4.2.1 Wind farm description

Figure 4.1 shows a top view schematic of the general floating offshore wind farm that we model in the current work. Floating wind turbines are treated as a system of particles that are distributed along the two-dimensional ocean surface. Throughout this chapter, we consider only three-cylinder semi-submersible floating platforms as per the baseline design presented by Robertson *et al.* [152]. Each floating structure is therefore connected to three anchors via mooring lines for the purpose of station-keeping.

We define the set $\mathscr{F} = \{1, 2, \cdots, N\}$ to denote the indices of the N floating wind turbines within the wind farm, and we refer to each individual turbine using the identifier i. We then number the wind turbines in ascending order based on their downstream location. That is to say, the most upstream turbine is numbered by $i = 1$, while the most downstream machine is identified by $i = N$.

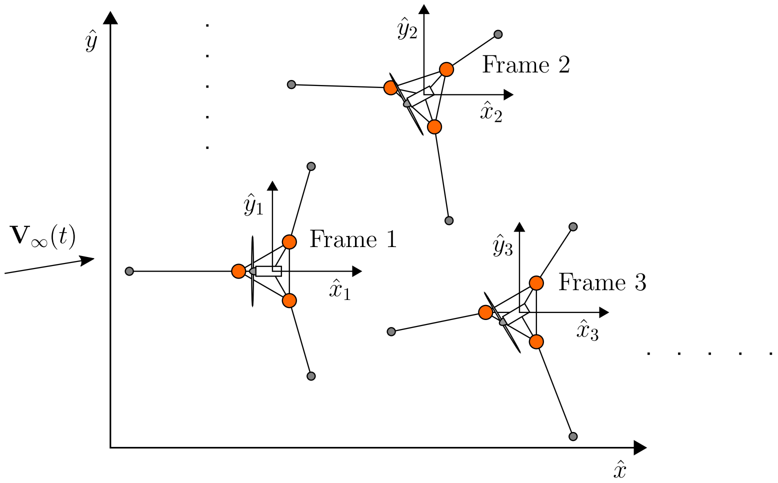


Figure 4.1: Schematic of a general floating offshore wind farm with semi-submersible platforms used as a basis for FOWFSim-Dyn's mathematical model.

The fixed global frame of reference is identified by the $\hat{x}$ and $\hat{y}$ axes. Each wind turbine also possesses a local non-inertial translating (though not rotating) reference frame that is attached to its center of gravity. We identify the reference frame that is fixed to turbine i as frame i. Further, the axes of frame i are referred to as $\hat{x}_i$ and $\hat{y}_i$.

We assume that a predominant wind direction exists, and that it is aligned with the positive $\hat{x}$ axis. The free stream wind velocity is then denoted by the vector $\mathbf{V}_\infty(t)$, which contains $\hat{x}$ and $\hat{y}$ components $U_\infty(t)$ and $V_\infty(t)$ as follows:

$$\mathbf{V}_\infty(t) \coloneqq \begin{bmatrix} U_\infty(t) & V_\infty(t) \end{bmatrix}^{\mathrm{T}}. \tag{4.1}$$

$U_\infty(t)$ therefore represents the free stream wind speed in the predominant wind direction, while $V_\infty(t)$ accounts for fluctuations in the transverse free stream wind speed.

4.2.2 States and inputs

Ultimately, FOWFSim-Dyn takes the following nonlinear state-space form:

$$\dot{\mathbf{x}}_{\text{farm}}(t) := f(\mathbf{x}_{\text{farm}}(t), \mathbf{u}_{\text{farm}}(t), \mathbf{V}_{\infty}(t)), \tag{4.2}$$

where the wind farm state vector $\mathbf{x}_{\text{farm}}(t)$ combines the floating wind turbine state vector $\mathbf{x}(t)$ with the wake state vector $\mathbf{x}_{\text{w}}(t)$ as follows:

$$\mathbf{x}_{\text{farm}}(t) := \begin{bmatrix} \mathbf{x}^{\text{T}}(t) & \mathbf{x}_{\text{w}}^{\text{T}}(t) \end{bmatrix}^{\text{T}}. \tag{4.3}$$

The wind turbine state vector $\mathbf{x}(t)$ comprises the position and velocity vectors of all floating wind turbines within the wind farm as follows:

$$\mathbf{x}(t) := \begin{bmatrix} \mathbf{r}_1^{\text{T}}(t) & \mathbf{r}_2^{\text{T}}(t) & \cdots & \mathbf{r}_N^{\text{T}}(t) & \mathbf{v}_1^{\text{T}}(t) & \mathbf{v}_2^{\text{T}}(t) & \cdots & \mathbf{v}_N^{\text{T}}(t) \end{bmatrix}^{\text{T}}, \tag{4.4}$$

where $\mathbf{r}_i(t)$ and $\mathbf{v}_i(t)$ are vectors containing $\hat{x}$ and $\hat{y}$ components of the position and velocity of turbine i as follows:

$$\mathbf{r}_i(t) \;:=\; \begin{bmatrix} x_i(t) & y_i(t) \end{bmatrix}^{\text{T}}, \tag{4.5}$$

$$\mathbf{v}_i(t) \;:=\; \begin{bmatrix} v_{x,i}(t) & v_{y,i}(t) \end{bmatrix}^{\text{T}}. \tag{4.6}$$

The wake state vector $\mathbf{x}_{\text{w}}(t)$ contains the states of the wakes generated by the N floating wind turbines as follows:

$$\mathbf{x}_{\text{w}}(t) := \begin{bmatrix} \mathbf{x}_{\text{w},1}^{\text{T}}(t) & \mathbf{x}_{\text{w},2}^{\text{T}}(t) & \cdots & \mathbf{x}_{\text{w},N}^{\text{T}}(t) \end{bmatrix}^{\text{T}}. \tag{4.7}$$

Assuming that the states of wake i are defined at $N_{\text{p},i}$ discrete points along the downstream direction, $\mathbf{x}_{\text{w},i}(t)$ comprises the states of wake i at each of these discrete points as follows:

$$\mathbf{x}_{\text{w},i}(t) := \begin{bmatrix} \mathbf{x}_{\text{w},i,1}^{\text{T}}(t) & \mathbf{x}_{\text{w},i,2}^{\text{T}}(t) & \cdots & \mathbf{x}_{\text{w},i,N_{\text{p},i}}^{\text{T}}(t) \end{bmatrix}^{\text{T}}. \tag{4.8}$$

The state vector $\mathbf{x}_{\text{w},i,p}(t)$ at each point p along wake i then consists of the wake centerline location $y_{\text{w},i,p}(t)$, wake velocity components $u_{\text{w},i,p}(t)$ and $v_{\text{w},i,p}(t)$, which correspond to the $\hat{x}_i$ and $\hat{y}_i$ directions, and the wake diameter $D_{\mathrm{w},i,p}(t)$ as follows:

$$\mathbf{x}_{\mathrm{w},i,p}(t) \coloneqq \begin{bmatrix} y_{\mathrm{w},i,p}(t) & u_{\mathrm{w},i,p}(t) & v_{\mathrm{w},i,p}(t) & D_{\mathrm{w},i,p}(t) \end{bmatrix}^{\mathrm{T}}. \tag{4.9}$$

These wake characteristics are portrayed in Figure 4.6 and discussed in Section 4.2.6.

The wind farm input vector $\mathbf{u}_{\mathrm{farm}}(t)$ contains the input vectors for the N floating wind turbines as follows:

$$\mathbf{u}_{\mathrm{farm}}(t) \coloneqq \begin{bmatrix} \mathbf{u}_1^{\mathrm{T}}(t) & \mathbf{u}_2^{\mathrm{T}}(t) & \cdots & \mathbf{u}_N^{\mathrm{T}}(t) \end{bmatrix}^{\mathrm{T}}, \tag{4.10}$$

where $\mathbf{u}_i(t)$ consists of the axial induction factor $a_i(t)$ and yaw angle $\gamma_i(t)$ of turbine i as follows:

$$\mathbf{u}_i(t) \coloneqq \begin{bmatrix} a_i(t) & \gamma_i(t) \end{bmatrix}^{\mathrm{T}}, \tag{4.11}$$

with all yaw angles defined as positive counter-clockwise from the $\hat{x}$ axis.

4.2.3 Solver block diagram

The block diagram for FOWFSim-Dyn is shown in Figure 4.2. The simulator consists of two main modules. The *aerodynamics module* requires the states $\mathbf{x}(t)$ and inputs $\mathbf{u}(t)$ of all turbines, along with the free stream wind velocity and acceleration vectors $\mathbf{V}_\infty(t)$ and $\dot{\mathbf{V}}_\infty(t)$ at time t. Its function is to compute the effective wind velocity vector $\mathbf{V}_i(t)$ that is incident on the rotor of turbine i for all $i \in \mathscr{F}$.

The *Floating turbine dynamics module* uses these incident wind velocity vectors, along with turbine states and inputs, to compute the rates of change of turbine states $\dot{\mathbf{x}}_i(t)$ at time t. Using a standard ordinary differential equation solver, state derivatives are integrated to compute state trajectories over time. This module also computes the power outputs of individual wind turbines as well as that of the entire wind farm $P_{\mathrm{farm}}(t)$.

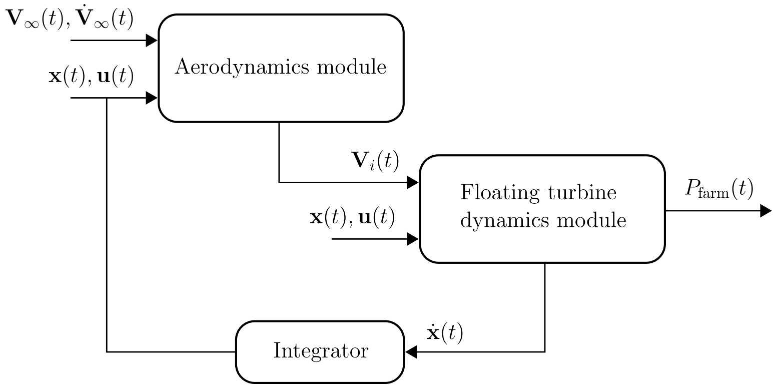

$\mathbf{V}_\infty(t), \dot{\mathbf{V}}_\infty(t)$
$\mathbf{x}(t), \mathbf{u}(t)$
Aerodynamics module
$\mathbf{V}_i(t)$
$\mathbf{x}(t), \mathbf{u}(t)$
Floating turbine dynamics module
$P_{\text{farm}}(t)$
$\dot{\mathbf{x}}(t)$
Integrator

Figure 4.2: Block diagram showing the computation modules of FOWFSim-Dyn along with information transfer routes.

4.2.4 Wind farm power output

The total power output of the wind farm $P_{\text{farm}}(t)$ is computed as the sum of electricity production from all wind turbines as follows:

$$P_{\text{farm}}(t) = \sum_{i \in \mathscr{F}} P_i(t), \tag{4.12}$$

where $P_i(t)$ is the power output of turbine i, and is estimated assuming steady-state performance as follows [121]:

$$P_i(t) = \frac{1}{8} C_{\text{p},i}(t) \rho_{\text{a}} \pi D_i^2 \left\| \mathbf{V}_{\text{rel},i}(t) \right\|^3 . \tag{4.13}$$

D_i is rotor diameter of turbine i, ρ_{a} is the density of air, and $\mathbf{V}_{\text{rel},i}(t)$ is the wind velocity that is incident upon the rotor of turbine i from the perspective of an observer who is fixed to turbine i. Referring to Figure 4.3, $\mathbf{V}_{\text{rel},i}(t)$ is defined as follows:

$$\mathbf{V}_{\text{rel},i}(t) = \mathbf{V}_i(t) - \mathbf{v}_i(t), \tag{4.14}$$

where $\mathbf{v}_i(t)$ is the velocity vector of turbine i, and $\mathbf{V}_i(t)$ is the wind velocity vector (in the global frame) that is incident upon the rotor of turbine i with the following $\hat{x}$ and $\hat{y}$ components:

$$\mathbf{V}_i(t) := \begin{bmatrix} U_i(t) & V_i(t) \end{bmatrix}^{\text{T}} . \tag{4.15}$$

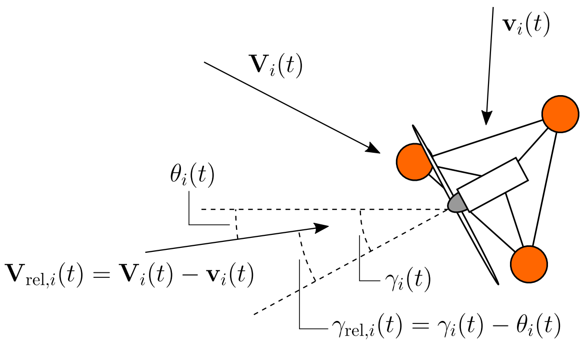

$\mathbf{v}_i(t)$
$\mathbf{V}_i(t)$
$\theta_i(t)$
$\mathbf{V}_{\mathrm{rel},i}(t) = \mathbf{V}_i(t) - \mathbf{v}_i(t)$
$\gamma_i(t)$
$\gamma_{\mathrm{rel},i}(t) = \gamma_i(t) - \theta_i(t)$

Figure 4.3: Schematic of floating platform velocity vector $\mathbf{v}_i(t)$, incident wind velocity vector $\mathbf{V}_i(t)$, and the relative incident velocity vector $\mathbf{V}_{\mathrm{rel},i}(t)$ at the location of turbine i.

$\mathbf{V}_i(t)$ is calculated using the wake interaction model discussed in Section 4.2.7.

The power coefficient $C_{\mathrm{p},i}(t)$ of turbine i is computed based on the vortex cylinder model of a yawed actuator disc as follows [29]:

$$C_{\mathrm{p},i}(t) = 4a_i(t)\left(\cos\gamma_{\mathrm{rel},i}(t) - a_i(t)\right)\Bigg(\cos\gamma_{\mathrm{rel},i}(t) + \tan\frac{\chi_i(t)}{2}\sin\gamma_{\mathrm{rel},i}(t) - a_i(t)\sec^2\frac{\chi_i(t)}{2}\Bigg), \quad (4.16)$$

where $a_i(t)$ is the axial induction factor of turbine i and, as per Figure 4.3, $\gamma_{\mathrm{rel},i}(t)$ is the yaw misalignment of turbine i relative to $\mathbf{V}_{\mathrm{rel},i}(t)$ as follows:

$$\gamma_{\mathrm{rel},i}(t) = \gamma_i(t) - \theta_i(t). \quad (4.17)$$

In the above expression, $\gamma_i(t)$ is the yaw angle of turbine i and $\theta_i(t)$ is the angle of $\mathbf{V}_{\mathrm{rel},i}(t)$ relative to the positive $\hat{x}$ axis as follows:

$$\theta_i(t) = \tan^{-1}\frac{V_i(t) - v_{y,i}(t)}{U_i(t) - v_{x,i}(t)}. \quad (4.18)$$

Finally, $\chi_i(t)$ is the wake skew angle immediately past the rotor of turbine i and is approximated as follows [29]:

$$\chi_i(t) = (0.6a_i(t) + 1)\,\gamma_{\mathrm{rel},i}(t). \tag{4.19}$$

4.2.5 Floating wind turbine motion

The rates of change of the position and velocity of turbine i are expressed as follows:

$$\begin{aligned} \dot{\mathbf{r}}_i(t) &= \mathbf{v}_i(t), & (4.20)\\ \dot{\mathbf{v}}_i(t) &= \frac{\mathbf{F}_i(t)}{m_i + m_{\mathrm{a},i}}, & (4.21) \end{aligned}$$

where m_i is the mass of floating wind turbine i. The added mass[1] $m_{\mathrm{a},i}$ associated with turbine i will be discussed along with the hydrodynamic drag force.

As shown in Figure 4.4, the total force $\mathbf{F}_i(t)$ acting on turbine i is the sum of its respective aerodynamic, hydrodynamic, and mooring line forces as follows:

$$\mathbf{F}_i(t) = \mathbf{F}_{\mathrm{a},i}(t) + \mathbf{F}_{\mathrm{h},i}(t) + \mathbf{F}_{\mathrm{m},i}(t). \tag{4.22}$$

The aerodynamic thrust force $\mathbf{F}_{\mathrm{a},i}(t)$ acting on the rotor of turbine i is expressed as follows:

$$\mathbf{F}_{\mathrm{a},i}(t) = \frac{1}{8}C_{\mathrm{t},i}(t)\rho_{\mathrm{a}}\pi D_i^2 \left\|\mathbf{V}_{\mathrm{rel},i}(t)\right\|^2 \mathbf{n}_i(t), \tag{4.23}$$

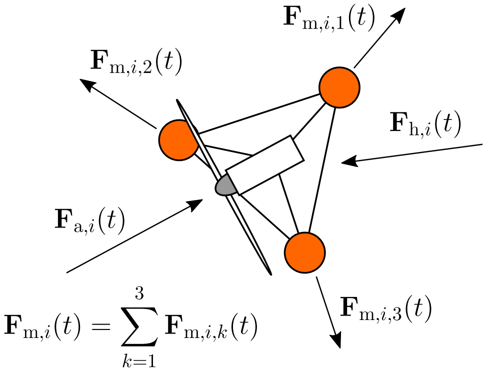
where the thrust coefficient $C_{\mathrm{t},i}(t)$ is computed based on the vortex cylinder model of a yawed actuator disc as follows [29]:

$$C_{\mathrm{t},i}(t) = 4a_i(t)\left(\cos\gamma_{\mathrm{rel},i}(t) + \tan\frac{\chi_i(t)}{2}\sin\gamma_{\mathrm{rel},i}(t) - a_i(t)\sec^2\frac{\chi_i(t)}{2}\right), \tag{4.24}$$

and $\mathbf{n}_i(t)$ is a unit vector normal to the rotor of turbine i as follows:

$$\mathbf{n}_i(t) = \begin{bmatrix}\cos\gamma_i(t) & \sin\gamma_i(t)\end{bmatrix}^{\mathrm{T}}. \tag{4.25}$$

[1] Added mass accounts for hydrodynamic loads that act upon an object that is accelerating with respect to the surrounding fluid. It compounds with hydrodynamic drag forces, which are typically modeled as functions of instantaneous velocity only.



$\mathbf{F}_{\mathrm{m},i,1}(t)$
$\mathbf{F}_{\mathrm{m},i,2}(t)$
$\mathbf{F}_{\mathrm{h},i}(t)$
$\mathbf{F}_{\mathrm{a},i}(t)$
$\mathbf{F}_{\mathrm{m},i}(t) = \sum_{k=1}^{3} \mathbf{F}_{\mathrm{m},i,k}(t)$
$\mathbf{F}_{\mathrm{m},i,3}(t)$

Figure 4.4: Schematic of aerodynamic thrust force $\mathbf{F}_{\mathrm{a},i}(t)$, hydrodynamic drag force $\mathbf{F}_{\mathrm{h},i}(t)$, and mooring line forces $\mathbf{F}_{\mathrm{m},i,k}(t)$ acting on wind turbine i with a semi-submersible floating platform.

Based on elementary fluid mechanics principles concerning immersed bodies, $\mathbf{F}_{\mathrm{h},i}(t)$ is approximated by summing the drag force contributions of all submerged components of turbine i as follows:

$$\mathbf{F}_{\mathrm{h},i}(t) = \frac{1}{2}\left(\sum_{j\in\mathscr{D}_i} C_{\mathrm{d},i,j}A_{\mathrm{d},i,j}\right)\rho_{\mathrm{w}}\left\|\mathbf{w}(t)-\mathbf{v}_i(t)\right\|\left(\mathbf{w}(t)-\mathbf{v}_i(t)\right), \tag{4.26}$$

where ρ_{w} is the density of ocean water, and $\mathbf{w}(t)$ is the ocean current velocity vector (which we assume to be $\mathbf{w}(t) = 0\,\mathrm{m/s}$ in this work). Let the set $\mathscr{D}_i = \{1,2,\cdots,N_{\mathrm{h},i}\}$ denote the indices of all submerged components that contribute to the hydrodynamic drag force acting on turbine i, with $N_{\mathrm{h},i}$ being equal to the total number of submerged components of turbine i. $C_{\mathrm{d},i,j}$ and $A_{\mathrm{d},i,j}$ are thereby the drag coefficient and reference area of the j^{th} submerged component of turbine i.

In a similar manner, the total added mass $m_{\mathrm{a},i}$ associated with turbine i is estimated by summing the added mass contributions of all submerged components of turbine i as follows:

$$m_{\mathrm{a},i} = \rho_{\mathrm{w}} \sum_{j\in\mathscr{D}_i} C_{\mathrm{a},i,j}A_{\mathrm{a},i,j}, \tag{4.27}$$

where $C_{\mathrm{a},i,j}$ is the added mass coefficient of the j^{th} submerged component of turbine i, and $A_{\mathrm{a},i,j}$ is the added mass reference area of the same component.

Let the set $\mathscr{M}_i = \{1, 2, \cdots, N_{\mathrm{m},i}\}$ denote the indices of all mooring lines connected to turbine i, with $N_{\mathrm{m},i}$ being equal to the total number of mooring lines attached to turbine i. $\mathbf{F}_{\mathrm{m},i}(t)$ may then be expressed as the sum of all mooring force contributions acting on turbine i as follows:

$$\mathbf{F}_{\mathrm{m},i}(t) = \sum_{k \in \mathscr{M}_i} \mathbf{F}_{\mathrm{m},i,k}(t), \tag{4.28}$$

where $\mathbf{F}_{\mathrm{m},i,k}(t)$ is the restoring force exerted on turbine i by its k^{th} mooring line. This force is calculated by first finding the magnitude of the horizontal component of tension within mooring line k of turbine i, and then projecting this tension in the appropriate direction as follows:

$$\mathbf{F}_{\mathrm{m},i,k}(t) = -H_{\mathrm{F},i,k}(t) \frac{\mathbf{r}_{\mathrm{F/A},i,k}(t)}{\left\| \mathbf{r}_{\mathrm{F/A},i,k}(t) \right\|}. \tag{4.29}$$

The function $H_{\mathrm{F},i,k}(t)$ outputs the horizontal component of tension along the k^{th} mooring line of turbine i. This function is generated by solving the static differential equations describing a suspended cable which is either partially contacting or fully lifted above the seabed. The relevant solution was described in Section 3.2.5.

As shown in Figure 4.5, the term $\mathbf{r}_{\mathrm{F/A},i,k}(t)$ describes the position vector from the anchor of the k^{th} mooring line of turbine i to the corresponding fairlead, and is expressed as follows:

$$\mathbf{r}_{\mathrm{F/A},i,k}(t) = \mathbf{r}_i(t) + \mathbf{r}_{\mathrm{F/G},i,k} - \mathbf{r}_{\mathrm{A},i,k}, \tag{4.30}$$

where $\mathbf{r}_{\mathrm{F/G},i,k}$ is a constant position vector from the center-of-gravity of turbine i to the fairlead that connects to the k^{th} mooring line of the same turbine, and $\mathbf{r}_{\mathrm{A},i,k}$ is a constant position vector representing the location of the anchor of the same mooring line. In Eq. (4.29), dividing $\mathbf{r}_{\mathrm{F/A},i,k}(t)$ by its Euclidean norm therefore produces a unit vector that points from the anchor of the k^{th} mooring line of turbine i to the corresponding fairlead. The restoring force associated with this mooring line pulls the turbine in the opposite direction.

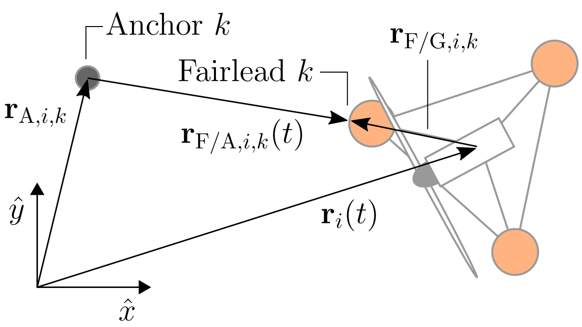

Anchor k
$\mathbf{r}_{\mathrm{F/G},i,k}$
Fairlead k
$\mathbf{r}_{\mathrm{A},i,k}$
$\mathbf{r}_{\mathrm{F/A},i,k}(t)$
$\hat{y}$
$\mathbf{r}_i(t)$
$\hat{x}$

Figure 4.5: Schematic of position vectors that are relevant for calculating the force in mooring line *k* of turbine *i*.

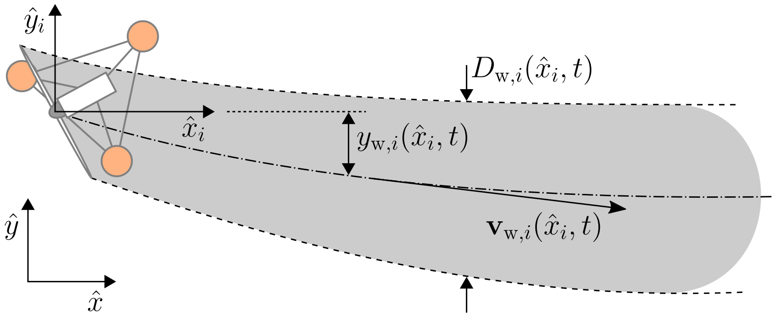

$\hat{y}_i$
$D_{\mathrm{w},i}(\hat{x}_i,t)$
$\hat{x}_i$
$y_{\mathrm{w},i}(\hat{x}_i,t)$
$\hat{y}$
$\mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)$
$\hat{x}$

Figure 4.6: Schematic of characteristics necessary for modeling the wake generated by turbine *i*. The wake centerline position $y_{\mathrm{w},i}(\hat{x}_i,t)$, average wake velocity $\mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)$, and wake diameter $D_{\mathrm{w},i}(\hat{x}_i,t)$ are defined within the reference frame that is fixed to turbine *i*.

4.2.6 Single wake model

Figure 4.6 shows the characteristics of interest when modeling wake *i*, which is the wake generated by the rotor of turbine *i*. These characteristics include the wake's centerline position $y_{\mathrm{w},i}(\hat{x}_i,t)$ relative to the $\hat{x}_i$ axis, its average velocity vector $\mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)$ measured in frame *i*, and its diameter $D_{\mathrm{w},i}(\hat{x}_i,t)$.

Two key assumptions are necessary for justifying the mathematical formulation presented in this section. First, if fluctuations in the wind direction relative to the $\hat{x}_i$ axes are presumed to be small, then all wake characteristics may be defined as smooth functions of only $\hat{x}_i$ and t. Furthermore, wake cross-sections may be assumed to always remain normal to the predominant flow direction, which corresponds to the positive $\hat{x}$ and $\hat{x}_i$ axes in our work.

Second, if the free stream wind speed is presumed to be significantly larger than the velocities of floating platforms, then the equations of motion describing any wake may be defined relative to a reference frame that is fixed to the wake-generating turbine. The frame of reference shown in Figure 4.6 is therefore non-inertial and translates with turbine i, while $y_{\mathrm{w},i}(\hat{x}_i,t)$, $\mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)$, and $D_{\mathrm{w},i}(\hat{x}_i,t)$ are defined in this translating frame. This approach eliminates the need to model wake behavior upstream of turbine i, while removing time-dependency from the wake centerline boundary condition (*i.e.* $y_{\mathrm{w},i}(\hat{x}_i,t)$ is always equal to zero at $\hat{x}_i = 0\,\mathrm{m}$).

Granting these preliminaries, the equations of motion describing wake i may now be derived. Specifically, we shall present partial differential equations that model wake average velocities, wake centerline locations, and wake diameters over space and time. Let the vector $\mathbf{L}_i(\hat{x}_i,t)$ describe the linear momentum deficit of wake i per unit length along the $\hat{x}_i$ axis as follows:

$$\mathbf{L}_i(\hat{x}_i,t) = \rho_a \frac{\pi}{4} D^2_{\mathrm{w},i}(\hat{x}_i,t)\left[\mathbf{V}_\infty(t) - (\mathbf{v}_i(t) + \mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t))\right]. \tag{4.31}$$

As $\mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)$ is measured in frame i, the term $\mathbf{v}_i(t) + \mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)$ redefines the velocity of wake i in the global frame. Since no external forces impact wake i, the time-derivative of $\mathbf{L}_i(\hat{x}_i,t)$ must equate to zero, which results in the following momentum conservation equation:

$$\begin{aligned}\frac{\partial \mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)}{\partial t} &+ (U_\infty(t) - v_{x,i}(t))\,\frac{\partial \mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)}{\partial \hat{x}_i} = \\ &\dot{\mathbf{V}}_\infty(t) - \dot{\mathbf{v}}_i(t) + \frac{2}{D_{\mathrm{w},i}(\hat{x}_i,t)}\frac{dD_{\mathrm{w},i}(\hat{x}_i,t)}{dt}\left(\mathbf{V}_\infty(t) - \mathbf{v}_i(t) - \mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)\right).\end{aligned} \tag{4.32}$$

The time-derivative of $y_{\mathrm{w},i}(\hat{x}_i,t)$ must equate to the $\hat{y}_i$ component of $\mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)$, which results in the following expression describing the wake centerline location:

$$\frac{\partial y_{\mathrm{w},i}(\hat{x}_i,t)}{\partial t} + (U_\infty(t) - v_{x,i}(t))\,\frac{\partial y_{\mathrm{w},i}(\hat{x}_i,t)}{\partial \hat{x}_i} = v_{\mathrm{w},i}(\hat{x}_i,t). \tag{4.33}$$

In Eqs. (4.32) and (4.33), $u_{\mathrm{w},i}(\hat{x}_i,t)$ and $v_{\mathrm{w},i}(\hat{x}_i,t)$ are the $\hat{x}_i$ and $\hat{y}_i$ components of $\mathbf{v}_{\mathrm{w},i}(\hat{x}_i,t)$, $v_{x,i}(t)$ is the velocity of turbine i in the $\hat{x}$ direction, $U_\infty(t)$ is the free stream wind speed in the $\hat{x}$ direction, and the term $U_\infty(t) - v_{x,i}(t)$ serves as the transport speed in the $\hat{x}_i$ direction. When modeling fluids using the three-dimensional Navier-Stokes equations, the transport and fluid velocities at any given point are equal. Following this logic, the transport speed in Eqs. (4.32) and (4.33) should simply be $u_{\mathrm{w},i}(\hat{x}_i,t)$. However, when neglecting three-dimensional effects, it is debatable exactly how the transport velocity should be defined. Our simulations indicate that defining the transport speed as the free stream wind speed (defined in frame *i*) yields predictions closer to experimental observations than does setting the transport speed to $u_{\mathrm{w},i}(\hat{x}_i,t)$.

In steady-state parametric wake models, the wake diameter is typically assumed to grow at a constant spatial expansion rate k_x along the downstream direction. When modeling wakes dynamically, however, we assume that wake diameters grow at a constant temporal expansion rate k_t. In other words, the time-derivative of $D_{\mathrm{w},i}(\hat{x}_i,t)$ must equate to k_t as follows:

$$\frac{\partial D_{\mathrm{w},i}(\hat{x}_i,t)}{\partial t} + \left(U_\infty(t) - v_{x,i}(t)\right)\frac{\partial D_{\mathrm{w},i}(\hat{x}_i,t)}{\partial \hat{x}_i} = k_t. \tag{4.34}$$

If the spatial expansion rate k_x under steady-state conditions is known for some reference free stream wind speed $U_{\infty,\mathrm{ref}}$, the temporal expansion rate at $U_{\infty,\mathrm{ref}}$ must be $k_t = k_x U_{\infty,\mathrm{ref}}$. Assuming that the free stream wind speed $\|\mathbf{V}_\infty(t)\|$ does not vary significantly from $U_{\infty,\mathrm{ref}}$, then k_t may be assumed to remain constant.

In order to obtain the wake states employed in Eq. (4.9), the spatial gradients in Eqs. (4.32), (4.33), and (4.34) must be discretized over some fixed downstream distance using the finite difference method, which would yield a system of nonlinear ordinary differential equations that would be rearranged to state-space form. We will not present the discretized forms of these equations as the finite difference method is an elementary numerical technique.

When implementing the above solution, we recommend the following initial conditions:

$$
\begin{aligned}
y_{\mathrm{w},i}(\hat{x}_i,0) &= \frac{V_\infty(0)}{U_\infty(0)}\hat{x}_i, &(4.35)\\
\mathbf{v}_{\mathrm{w},i}(\hat{x}_i,0) &= \mathbf{V}_\infty(0)-\mathbf{v}_i(0), &(4.36)\\
D_{\mathrm{w},i}(\hat{x}_i,0) &= D_i+k_x\hat{x}_i, &(4.37)
\end{aligned}
$$

which ensure, respectively, that all wake centerlines are initially aligned with the free stream wind, wake velocities are initially equal to the free stream wind velocity, and that wake diameters initially grow at a predefined spatial rate k_x. Note that D_i is the diameter of turbine i. With regards to boundary conditions, the following are necessary based on assumptions inherent to FOWFSim-Dyn:

$$
\begin{aligned}
y_{\mathrm{w},i}(0,t) &= 0, &(4.38)\\
\mathbf{v}_{\mathrm{w},i}(0,t) &= \mathbf{v}_{\mathrm{w,init},i}(t), &(4.39)\\
D_{\mathrm{w},i}(0,t) &= D_i, &(4.40)
\end{aligned}
$$

Equation (4.38) states that the centerline of wake i at $\hat{x}_i = 0\,\mathrm{m}$ must always correspond to the location of turbine i, which is in fact the origin of frame i. Equation (4.39) states that the velocity of wake i at $\hat{x}_i = 0\,\mathrm{m}$ must always be equal to the wake velocity $\mathbf{v}_{\mathrm{w,init},i}(t)$ immediately downstream of the rotor of turbine i. Finally, Eq. (4.40) requires that the diameter of wake i at the location of turbine i is always equal to the rotor diameter of this turbine.

We calculate the velocity vector $\mathbf{v}_{\mathrm{w,init},i}(t)$ based on simplifications made to Glauert's momentum theory [29] by Bastankhah and Porté Agel [18] as follows:

$$
\mathbf{v}_{\mathrm{w,init},i}(t) = \|\mathbf{V}_{\mathrm{rel},i}(t)\|\sqrt{1-C_{\mathrm{t},i}(t)}\begin{bmatrix}\cos\left(\xi_{\mathrm{w,init},i}(t)+\theta_i(t)\right)\\ \sin\left(\xi_{\mathrm{w,init},i}(t)+\theta_i(t)\right)\end{bmatrix}, \qquad (4.41)
$$

where $\xi_{\mathrm{w,init},i}(t)$ is the initial wake skew angle, which is expressed as follows based on a momentum conservation derivation reported by Jiménez *et al.* [93]:

$$
\xi_{\mathrm{w,init},i}(t) = -\frac{C_{\mathrm{t},i}(t)}{2}\cos^2\gamma_{\mathrm{rel},i}(t)\sin\gamma_{\mathrm{rel},i}(t). \qquad (4.42)
$$

The derivation by Bastankhah and Porté Agel [18] assumes that the free stream wind velocity is aligned with the $\hat{x}$ axis. As a result, the addition of $\theta_i(t)$ to $\xi_{\mathrm{w,init},i}(t)$ in Eq. (4.41) accounts for the misalignment of $\mathbf{V}_\infty(t)$ relative to the $\hat{x}$ axis.

4.2.7 Wake interaction model

When a wind turbine rotor is influenced by wakes that are generated from multiple upstream turbines, a wake interaction model is necessary for approximating the resultant effective wind speed that is incident on the downstream rotor. The most commonly used wake interaction technique is based on the assumption that the effective kinetic energy deficit at the location of the downstream rotor must be equal to the sum of kinetic energy deficits of all pertinent wakes [98]. As a result, the effective wind speed at the downstream rotor is a function of the root-sum-square of relevant wake velocity deficits. Further enhancement may be obtained by approximating wake velocity profiles using Gaussian distributions [18]. We continue to make use of this wake interaction methodology.

Let the set $\mathscr{U}_i = \{1, 2, \cdots, i-1\}$ denote the indices of all turbines that are located upstream of turbine i. The effective wind velocity vector that is incident on the rotor of turbine i may therefore be expressed as follows:

$$\mathbf{V}_i(t) = \left\{ \|\mathbf{V}_\infty(t)\| - \sqrt{\sum_{q \in \mathscr{U}_i} \left(\|\mathbf{V}_\infty(t)\| - \overline{\mathbf{v}}_{\mathrm{w},q\to i}(t) \cdot \mathbf{n}_\infty(t)\right)^2} \right\} \mathbf{n}_\infty(t). \qquad (4.43)$$

where $\mathbf{n}_\infty(t)$ is a unit vector aligned with $\mathbf{V}_\infty(t)$ as follows:

$$\mathbf{n}_\infty(t) = \frac{\mathbf{V}_\infty(t)}{\|\mathbf{V}_\infty(t)\|}, \qquad (4.44)$$

and $\overline{\mathbf{v}}_{\mathrm{w},q\to i}(t)$ is the *effective* velocity of wake q that is incident upon the rotor of wake i. Equation (4.43) projects $\overline{\mathbf{v}}_{\mathrm{w},q\to i}(t)$ along the free stream wind direction (hence the dot product operation with $\mathbf{n}_\infty(t)$), and then computes the velocity deficit in this direction. Average wake velocities perpendicular to the free stream wind direction are assumed to be negligibly small far enough downstream; their effects are therefore neglected.

We now describe our procedure for computing $\overline{\mathbf{v}}_{\mathrm{w},q\to i}(t)$. Let $\mathbf{v}_{\mathrm{w},q\to i}(t)$ denote the average velocity of wake q at the location of wake i as follows:

$$\mathbf{v}_{\mathrm{w},q\to i}(t) = \mathbf{v}_q(t) + \mathbf{v}_{\mathrm{w},q}(x_i(t) - x_q(t), t). \tag{4.45}$$

Since the average velocity vector of wake q is defined in frame q, the substitution $\hat{x}_q = x_i(t) - x_q(t)$ into $\mathbf{v}_{\mathrm{w},q}(\hat{x}_q, t)$ is necessary for identifying the location of turbine i in frame q. The addition of the turbine velocity vector $\mathbf{v}_q(t)$ then transforms $\mathbf{v}_{\mathrm{w},q}(x_i(t) - x_q(t), t)$ to the global frame.

The next step is to generate a Gaussian profile $\check{\mathbf{v}}_{\mathrm{w},q\to i}(r,t)$, where r is the radial distance from the centerline of wake q, to approximate the continuous velocity distribution of wake q at the location of wake i. Imposing a requirement that the total momentum deficit of $\mathbf{V}_\infty(t) - \check{\mathbf{v}}_{\mathrm{w},q\to i}(r,t)$ per unit length must equate that of a top-hat distribution with amplitude $\mathbf{V}_\infty(t) - \mathbf{v}_{\mathrm{w},q\to i}(t)$ as follows:

$$\int_0^\infty \rho_a 2\pi r \left(\mathbf{V}_\infty(t) - \check{\mathbf{v}}_{\mathrm{w},q\to i}(r,t)\right) dr = \rho_a \frac{\pi}{4} D_{\mathrm{w},q\to i}(t) \left(\mathbf{V}_\infty(t) - \mathbf{v}_{\mathrm{w},q\to i}(t)\right), \tag{4.46}$$

the following Gaussian profile is then obtained:

$$\mathbf{V}_\infty(t) - \check{\mathbf{v}}_{\mathrm{w},q\to i}(r,t) = \frac{1}{8}\left(\frac{D_{\mathrm{w},q\to i}(t)}{\sigma}\right)^2 \left(\mathbf{V}_\infty(t) - \mathbf{v}_{\mathrm{w},q\to i}(t)\right) \exp\frac{-r^2}{2\sigma^2}, \tag{4.47}$$

where $D_{\mathrm{w},q\to i}(t)$ is the diameter of wake q at the location of turbine i as follows:

$$D_{\mathrm{w},q\to i} = D_{\mathrm{w},q}(x_i(t) - x_q(t), t). \tag{4.48}$$

The standard deviation σ in Eq. (4.47) may be estimated based on experimental or high-fidelity numerical data.

Finally, the effective velocity $\overline{\mathbf{v}}_{\mathrm{w},q\to i}(t)$ is obtained by averaging $\check{\mathbf{v}}_{\mathrm{w},q\to i}(r,t)$ along the rotor area A_i of turbine i. This task is achieved by numerically computing the following integral at each time-step:

$$\overline{\mathbf{v}}_{\mathrm{w},q\to i}(t) = \frac{4}{\pi D_i^2} \int_{A_i} \check{\mathbf{v}}_{\mathrm{w},q\to i}(r,t) dA. \tag{4.49}$$

4.2.8 Model limitations

Several assumptions have been made when developing FOWFSim-Dyn which impose limitations on its fidelity and applicability. The current subsection summarizes these limitations.

Two-dimensional floating wind turbine dynamics

The first and most crucial of these assumptions is that floating platform motion may be adequately captured using a two-dimensional planar model. That is to say, we neglect floating platform heave, yaw, pitch, and roll. In consequence, FOWFSim-Dyn fails to capture dynamic effects induced by ocean waves and oscillatory wind conditions on platform rotation. FOWFSim-Dyn remains appropriate for wind farm controller design and testing, since this application is primarily concerned with average rotor positions over extended periods of time. However, any attempt to control or evaluate individual wind turbine dynamics requires the use of three-dimensional multi-body nonlinear modeling tools.

Steady-state mooring line model

Although we present a dynamic model, mooring line tensions are found based on the solution to a static suspended cable problem. It has been reported by Hall *et al.* [79] that such static models accurately predict mooring line loads and floating wind turbine motion; thus rendering them appropriate for wind farm control. However, Hall *et al.* [79] also mentioned that use of such models may lead to large inaccuracies in turbine load predictions. Therefore, analysis and control of individual turbine motion must consider higher-fidelity modeling techniques such as a lumped-mass dynamic mooring line model [78].

Steady-state turbine aerodynamics

Turbine power outputs and thrust forces (Eqs. (4.13) and (4.23)), along with their respective coefficients (Eqs. (4.16) and (4.24)), are calculated based on steady-state actuator disc theory. This approach assumes ideal rotors and fails to capture unsteady aerodynamic effects and asymmetric rotor loadings. These phenomena significantly influence blade loads when yaw misalignment occurs; however, for the purpose of wind farm control, our focus lies on the overall influence of rotor operation on fully-developed wake regions. Nonetheless, any turbine-level analysis requires more detailed fluid-structure interaction modeling.

The computation of $\mathbf{v}_{\mathrm{w,init},i}(t)$ in Eq. (4.41), which is the average wake velocity immediately downstream of turbine i, relies on a steady-state momentum balance on a control volume spanning across the rotor of turbine i. As a result, momentum fluxes into and out of this control volume are considered, while the rate-of-change of momentum within the control volume is neglected. Given the low density of air, these inertial effects may be neglected, although their significance should be investigated.

Sources of wake deflection

FOWFSim-Dyn does not capture wake centerline deflection caused by rotor rotation. This phenomenon was first observed in high-fidelity simulations conducted by Gebraad *et al.* [72]; however, more recent work by Fleming *et al.* [60] showed that the scale of this phenomenon is insignificant. Instead, Fleming *et al.* [60] observed that vortices generated by turbine rotors induce wake deflection past downstream machines, even if their rotors are not operated with yaw offset. Additional terms may be added to Eqs. (4.32) and (4.33) to account for such phenomena.

Spatial-uniformity and consistency of the free stream wind

In the current chapter, we have assumed that the free stream wind velocity is uniform throughout the wind farm, which is why the variable $\mathbf{V}_\infty(t)$ is solely a function of time. This variable may readily be expressed as $\mathbf{V}_\infty(\hat{x},t)$ if spatial variations of the free stream wind velocity are known. Furthermore, in order to represent wake characteristics purely as a function of the downstream distance along the $\hat{x}_i$ axes, while ignoring changes in the cross-sectional areas of wakes, we assumed that variations in the free stream wind direction are small relative to the $\hat{x}$ axis. The $\hat{y}$ component of the free stream wind velocity must therefore remain small in comparison to its $\hat{x}$ component.

4.3 Model validation

In this section, we first perform a mesh sensitivity analysis to ascertain the dependency of model predictions upon the size of finite difference elements in Section 4.3.1. We then validate FOWFSim-Dyn's capacity for predicting floating platform motion in Section 4.3.2, and wake transport in Section 4.3.3 and Section 4.3.4.

4.3.1 Mesh sensitivity analysis

For a mesh sensitivity study, we simulate the experimental setup employed by Bastankhah and Porté Agel [18]. Namely, the wake of a single fixed-foundation turbine with diameter $D = 15\,\mathrm{cm}$ is simulated with a steady free stream wind speed of $U_\infty = 4.88\,\mathrm{m/s}$. The turbine's axial induction factor is set to the optimal value of $a = 1/3$ and a yaw angle of $\gamma = 20\,\mathrm{deg}$ is implemented to observe mesh effects on wake deflection. To approximate steady-state results, all simulations are run for a duration of 5 sec and data is extracted from the final time-step. The Gaussian profile standard deviation is set to $\sigma = 0.025\hat{x} + 0.396\,\mathrm{m}$ based on experimental data[2] reported by Bastankhah and Porté Agel [18]. The (diametrical) spatial wake expansion constant is set to $k_x = 0.08$ as per the recommendation by Shakoor *et al.* [161].

Simulated wake centerlines and normalized velocity deficit profiles at a downstream distance of $7D$ are plotted in Figure 4.7 for different finite difference element sizes. Qualitatively, it is apparent that the evolution of the wake centerline is insignificantly influenced by the mesh size. At $\hat{x}/D = 16$, the centerline deflection obtained using an element size of $8D$ only differs by 5 % relative to the value corresponding to an element size of $0.25D$. As a result, we solely utilize the maximum normalized velocity deficit as a convergence criterion.

Table 4.1 lists the computation times corresponding to different element sizes from Figure 4.7 as well as predicted maximum normalized velocity deficits. Dynamic simulations were performed using the MATLAB fourth-order Runge-Kutta solver implemented on a laptop computer with a 2.80 GHz Intel Core i7-7700HQ processor. Table 4.1 also lists the convergence of the maximum normalized ve-

[2] Velocity profiles corresponding to a yaw angle of $\gamma = 0\,\mathrm{deg}$ from Fig. 21 in the paper by Bastankhah and Porté Agel [18] were digitized and Gaussian function curve fitting was used to compute the standard deviation.

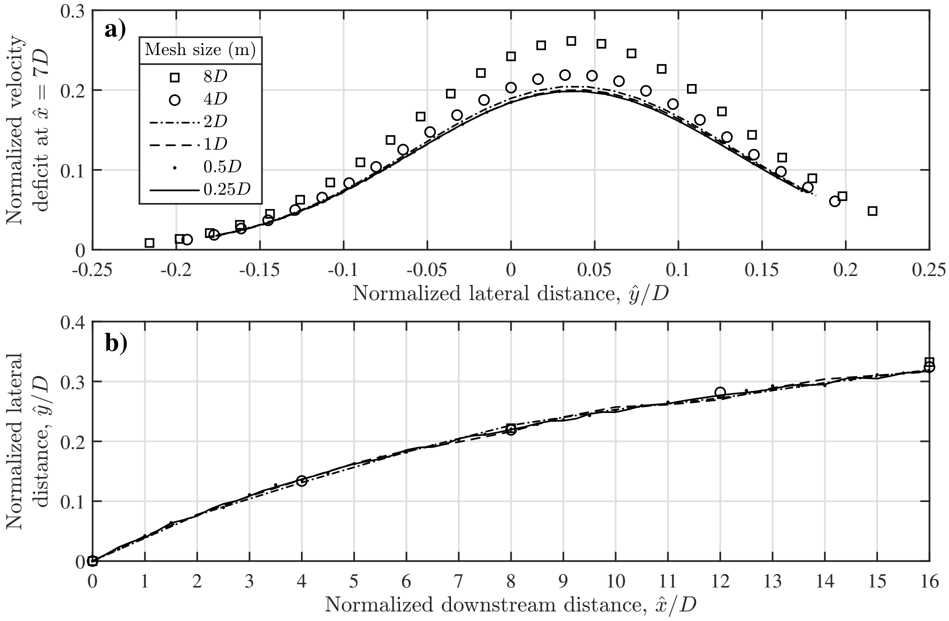


Figure 4.7: Effects of various finite difference mesh element sizes on **a)** the steady-state wake velocity profile at a downstream distance of $7D$, and **b)** the steady-state wake centerline evolution. Simulation parameters: $D = 15\,\mathrm{cm}$, $U_\infty = 8\,\mathrm{m/s}$, $a = 1/3$, $\gamma = 20\,\mathrm{deg}$, $k_x = 0.08$, $\sigma = 0.025\hat{x} + 0.396\,\mathrm{m}$.

locity deficit as the element size is decreased. We observe that mesh sensitivity is sufficiently reduced at an element size of $1D$ since further reduction to $0.5D$ only results in a 0.69 % change in the predicted maximum normalized velocity deficit. An element size of $1D$ is also appropriate from the standpoint of time-efficiency as it requires 3.2 sec of computation time to run a 5 sec simulate.

4.3.2 Validation of platform dynamics

To validate the platform dynamics module of FOWFSim-Dyn, we compare its predictions for a single floating wind turbine against those of the National Renewable Energy Laboratory's (NREL's) wind turbine modeling software *Fatigue, Aerodynamics, Structures, and Turbulence* (FAST) in Figure 4.8. FAST is a widely-accepted tool for modeling three-dimensional floating offshore wind turbine dy-

Table 4.1: Computation times and maximum normalized velocity deficits corresponding to different simulated mesh element sizes from Figure 4.7. The final column lists the convergence of the maximum normalized velocity deficit. In other words, it contains the relative difference in the maximum normalized velocity deficit that would be obtained if each element size was halved. For instance, if the element size were to be reduced from $8D$ to $4D$, the predicted velocity deficit would change by $19.46\,\%$. The computation times correspond to 5 sec long simulations.

Elm. size (D)	Comp. time (sec)	Max. velocity deficit (-)	Rel. diff. (%)
8	0.397	0.261	19.46
4	0.703	0.219	7.19
2	1.496	0.204	2.00
1	3.237	0.200	0.69
0.5	8.948	0.199	0.17
0.25	24.035	0.198	-

namics and has been experimentally validated in the context of semi-submersible floating platforms by Coulling *et al.* [44]. Results presented in Figure 4.8 were obtained using a steady free stream wind speed of $U_\infty = 10\,\text{m/s}$, optimal axial induction, a yaw angle of $\gamma = 15\text{deg}$, and mooring line cable lengths of $L = 850\,\text{m}$. In FAST simulations, all floating platform degrees-of-freedom were enabled. The simulated wind turbine is based on NREL's 5 MW baseline design presented by Jonkman *et al.* [96], and the floating platform and mooring subsystem is modeled after the design described by Robertson *et al.* [152]. Details corresponding to these designs are listed in **??**. Figure 4.8 also includes FOWFSim-Dyn predictions with the static mooring line model replaced by the dynamic lumped-parameter cable model presented by Hall *et al.* [79]; this result is identified by *FOWFSim-Dyn** in the figure legend.

From a qualitative standpoint, Figure 4.8 shows that all models predict the expected platform motion in the downwind and crosswind directions. Since the turbine nacelle is yawed counterclockwise by $\gamma = 15\,\text{deg}$, aerodynamic forces displace the floating platform in the positive crosswind direction. There is little discrepancy between FAST and FOWFSim-Dyn predictions with regards to platform displacement in the downwind direction. The root-mean-square error (RMSE) between these datasets is 1.37 m, which is 8.5 % of the steady-state downwind platform displacement of 16.1 m predicted by FAST.

Table 4.2: List of floating wind farm properties used during dynamic simulations. All wind turbines are based on the NREL 5 MW baseline design presented by Jonkman *et al.* [96], and all floating platforms and mooring subsystems are modeled after the design described by Robertson *et al.* [152].

External properties		
ρ_a $\left(\mathrm{kg/m^3}\right)$	1.225	Air density
ρ_w $\left(\mathrm{kg/m^3}\right)$	1028	Water density
Floating turbine properties		
m_i (kg)	1.4×10^7	Mass
D_i (m)	126	Rotor diameter
A_i $\left(\mathrm{m^2}\right)$	$\frac{\pi}{4} D_i^2$	Rotor area
η_p	0.786	Electrical power conversion efficiency [72]
p_p	1.88	Power coefficient tuning parameter [72]
Floating platform hydrodynamic properties		
$C_{\mathrm{d},i,1\to 3}$	0.61	Drag coefficients of three top cylinder portions
$C_{\mathrm{d},i,4\to 6}$	0.68	Drag coefficients of three bottom cylinder portions
$C_{\mathrm{d},i,7}$	0.56	Drag coefficient of middle cylinder
$D_{\mathrm{d},i,1\to 3}$ (m)	12	Diameters of three top cylinder portions
$D_{\mathrm{d},i,4\to 6}$ (m)	24	Diameters of three bottom cylinder portions
$D_{\mathrm{d},i,7}$ (m)	6.5	Diameter of middle cylinder
$L_{\mathrm{d},i,1\to 3}$ (m)	14	Submerged lengths of three top cylinder portions
$L_{\mathrm{d},i,4\to 6}$ (m)	6	Submerged lengths of three bottom cylinder portions
$L_{\mathrm{d},i,7}$ (m)	20	Submerged length of middle cylinder
$A_{\mathrm{d},i,j}$ $\left(\mathrm{m^2}\right)$	$L_{\mathrm{d},i,j} D_{\mathrm{d},i,j}$	Drag reference area of any cylinder
$C_{\mathrm{a},i,j}$	0.63	Added mass coefficients of any cylinder
$A_{\mathrm{a},i,j}$ $\left(\mathrm{m^2}\right)$	$\frac{\pi}{4} L_{\mathrm{d},i,j} D_{\mathrm{d},i,j}^2$	Added mass reference area of any cylinder
Mooring system properties		
$\mathbf{r}_{\mathrm{F/G},i,1}^\mathrm{T}$ (m)	$\begin{bmatrix} 20.4 & 35.4 \end{bmatrix}$	Position vector from turbine center to first fairlead
$\mathbf{r}_{\mathrm{F/G},i,2}^\mathrm{T}$ (m)	$\begin{bmatrix} -40.9 & 0 \end{bmatrix}$	Position vector from turbine center to second fairlead
$\mathbf{r}_{\mathrm{F/G},i,3}^\mathrm{T}$ (m)	$\begin{bmatrix} 20.4 & -35.4 \end{bmatrix}$	Position vector from turbine center to third fairlead
$\mathbf{r}_{\mathrm{A},i,1}^\mathrm{T}$ (m)	$\mathbf{r}_{\mathrm{neutral},i}^\mathrm{T} + \begin{bmatrix} 418.80 & 725.4 \end{bmatrix}$	Location of first anchor of any turbine
$\mathbf{r}_{\mathrm{A},i,1}^\mathrm{T}$ (m)	$\mathbf{r}_{\mathrm{neutral},i}^\mathrm{T} + \begin{bmatrix} -837.6 & 0 \end{bmatrix}$	Location of second anchor of any turbine
$\mathbf{r}_{\mathrm{A},i,1}^\mathrm{T}$ (m)	$\mathbf{r}_{\mathrm{neutral},i}^\mathrm{T} + \begin{bmatrix} 418.80 & -725.4 \end{bmatrix}$	Location of third anchor of any turbine
z_F (m)	186	Fairlead distance above seabed
L (m)	900	Cable length
w (N/m)	1065.7	Cable weight per unit length in water
$A_\mathrm{m} E$ (N)	753.6×10^6	Cable tension per unit strain
μ_s	1	Coefficient of static friction between cable and seabed

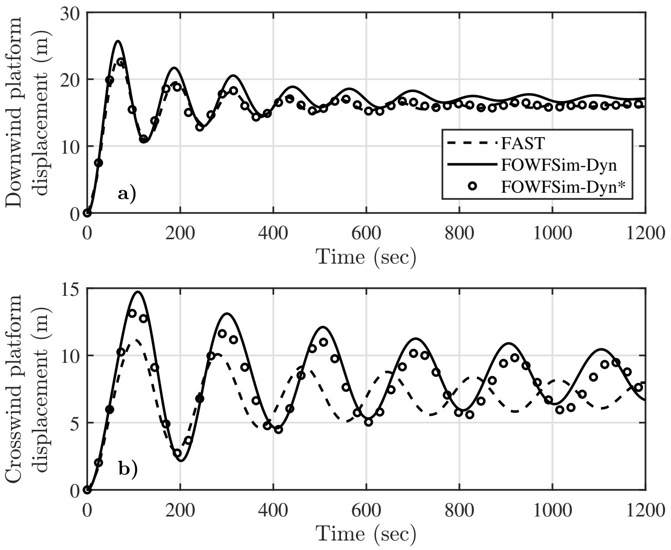


Figure 4.8: Comparison between FOWFSim-Dyn and FAST predictions of platform motion for a single floating wind turbine based on the baseline designs described in **??**. The curve identified by FOWFSim-Dyn* utilizes the dynamics lumped-parameter mooring line model proposed by Hall *et al.* [79] in place of the static catenary model described in Chapter 3. Simulation parameters: $U_\infty = 10\,\text{m/s}$, $a = 1/3$, $\gamma = 15\,\text{deg}$, $L = 850\,\text{m}$.

When examining crosswind platform displacement, the RMSE value between FAST and FOWFSim-Dyn predictions rises to 2.74 m. One method for assessing the significance of this error is to compute the normalized overlap area (NOA) between turbines rotors from the two simulations. The NOA is the overlap area between the rotors of turbines normalized against the rotor area. Therefore, if the crosswind displacements of two turbines are identical, then the NOA between their rotors must be unity. If, on the other hand, the crosswind displacements are so substantial that rotor overlap is completely eliminated, then the NOA will be zero. Since the ultimate purpose of FOWFSim-Dyn is to predict wind farm power output, which, due to the wake effect, is primarily depended upon the projected overlap areas between the rotors of adjacent turbines, the NOA is a meaningful metric for evaluating the effectiveness of FOWFSim-Dyn in predicting crosswind platform motion. The RMSE value of 2.74 m corresponds to an NOA of 97.2 %.

That is, when wind farm-level performance is of concern, FOWFSim-Dyn predicts projected rotor overlap areas between adjacent turbines with a discrepancy of 2.8 %.

One major source of error between FAST and FOWFSim-Dyn predictions is that FAST uses actuator line theory to predict aerodynamic loads, while FOWFSim-Dyn utilizes actuator disc theory. This source of error may be mitigated by computing aerodynamic loads in FOWFSim-Dyn using thrust coefficient values obtained from FAST. Further, the presence of all six platform degrees-of-freedom in FAST contributes to the discrepancy in oscillation periods observed in Figure 4.8. Finally, the mooring line modeling approach is a non-trivial contributing factor. FAST utilizes the lumped-parameter dynamic mooring line model proposed by Hall *et al.* [79], while FOWFSim-Dyn employs a static catenary model. The inclusion of FOWFSim-Dyn* curves in Figure 4.8 isolates the error introduced by switching between static and dynamic mooring line models. Comparing FOWFSim-Dyn to FOWFSim-Dyn* plots, a noticeable discrepancy in the oscillation period is introduced, while RMSE values between the two curves are 1.26 and 0.87 m in the downwind and crosswind directions.

4.3.3 Steady-state validation of wake transport

FOWFSim-Dyn predictions of steady-state[3] wake centerlines and normalized velocity profiles are compared against experimental results reported by Bastankhah and Porté Agel [18] in Figure 4.9. Wake centerline evolutions are well-predicted for all simulated yaw angles and downstream locations. For yaw angles of $\gamma = 0$, 10, and 20 deg, maximum discrepancies between predicted wake centerlines and experimental measurements are 6.87, 7.60, and 8.19 % of the rotor diameter, respectively.

Simulated normalized velocity profiles deviate significantly from experimental measurements at downstream locations closer than $7D$. For instance, at a yaw angle of $\gamma = 0$ deg, the RMSE between experimental and predicted velocity profiles ranges from 12.4 % of the free stream wind speed at $\hat{x} = 4D$ to 4.7 % at $\hat{x} = 7D$.

[3] Validating predictions of dynamic wake behaviour is not possible at this time due to the absence of high-fidelity simulation tools capable of modeling floating offshore wind farms; we thus defer this process to future work.

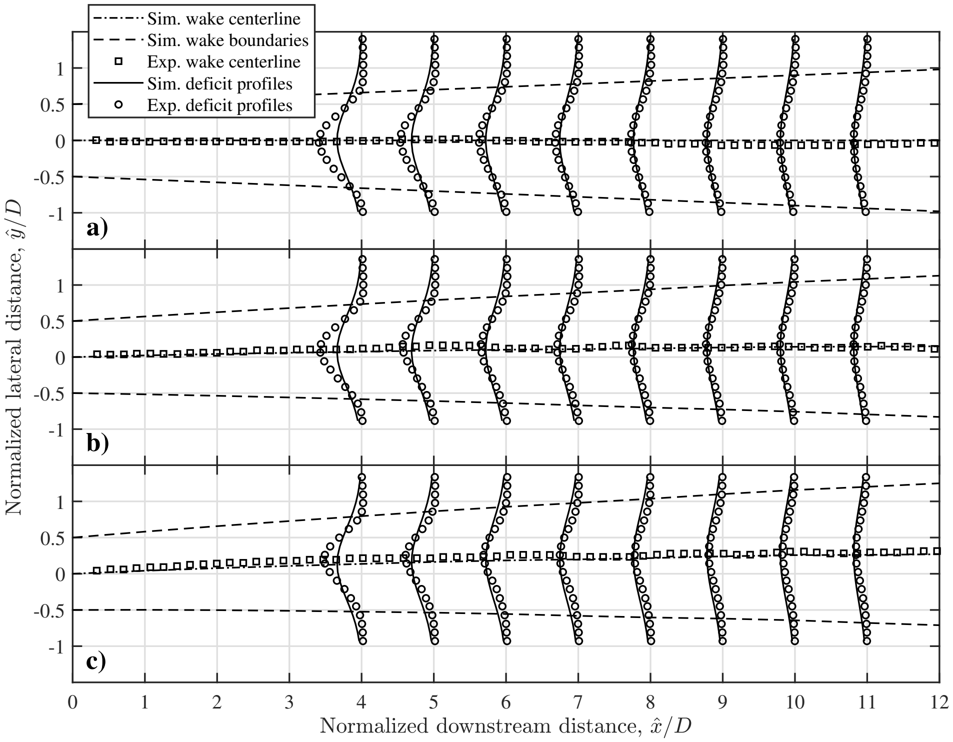


Figure 4.9: Comparison between FOWFSim-Dyn predictions and experimental results reported by Bastankhah and Porté Agel [18]. Each figure shows steady-state wake centerlines and normalized velocity profiles corresponding to yaw angles of **a)** $\gamma = 0\,\mathrm{deg}$, **b)** $\gamma = 10\,\mathrm{deg}$, and **c)** $\gamma = 20\,\mathrm{deg}$. Normalized velocity profiles range from zero to one using the same scaling as the $\hat{x}/D$ axis, but have been shifted to the downstream location where they are measured. Simulation parameters: $D = 15\,\mathrm{cm}$, $U_\infty = 8\,\mathrm{m/s}$, $a = 1/3$, $k_x = 0.08$, $\sigma = 0.025\hat{x} + 0.396\,\mathrm{m}$.

Such inaccuracies at close downstream distances are expected since FOWFSim-Dyn does not consider the inviscid nature of flow within the near-wake region. Beyond $\hat{x} = 7D$, velocity profiles are well-predicted with RMSE values that remain below 3.87 % of the free stream wind speed.

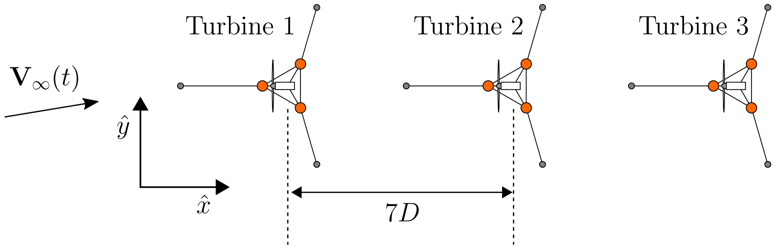


Figure 4.10: Schematic of the 1×3 wind farm with inter-turbine spacings of $7D$ used for dynamic simulations. All wind turbines are based on the NREL 5 MW baseline design presented by Jonkman *et al.* [96], and all floating platforms and mooring subsystems are modeled after the design described by Robertson *et al.* [152].

4.3.4 Dynamic assessment of wake transport

Our final tasks are to demonstrate the capability of FOWFSim-Dyn to capture the intended dynamic phenomena and to ensure that predicted turbine and wake behaviors respect physical intuition. The wind farm configuration that is used for dynamic simulations is shown in Figure 4.10. This plant contains a single row of three floating offshore wind turbines that are aligned with the predominant free stream wind direction. The neutral positions of the floating turbines are spaced $7D$ apart. All wind turbines are based on NREL's 5 MW baseline design presented by Jonkman *et al.* [96], and all floating platforms and mooring subsystems are modeled after the design described by Robertson *et al.* [152]. Details corresponding to these designs are listed in Table 4.2. In all simulations, we increase the lengths of mooring lines from their baseline values (*i.e.* $L = 835\,\text{m}$) to $L = 900\,\text{m}$ to render floating platform motion more notable. In all the following cases, less than 10 sec of computation time was required to complete simulations on a laptop computer with a 2.80 GHz Intel Core i7-7700HQ processor.

Simulation scenario 1

The first of three simulated scenarios maintains constant wind speed and direction with $U_\infty(t) = 8\,\text{m/s}$ and $V_\infty(t) = 0\,\text{m/s}$, while rotor yaw angles are fixed at $\gamma_1(t) = \gamma_3(t) = -20\,\text{deg}$ and $\gamma_2(t) = +20\,\text{deg}$. All axial induction factors are maintained at $a_1(t) = a_2(t) = a_3(t) = 1/3$. All floating platforms are locked at their neutral positions for the first 1000 sec of simulation, after which they are permitted to relocate. The aim of this scenario is to assess floating platform motion. Snapshots of velocity contours for simulation scenario 1 are shown in Figure 4.11. As expected, the alternating assignment of yaw angles causes adjacent floating platforms to shift in opposite directions over time. Further, the leading turbine displays the greatest amount of relocation from its neutral position (*i.e.* the left-most white + symbol) since its incident wind speed is the largest (*i.e.* its incident wind speed is the free stream wind speed uninhibited by upstream rotors). The trailing turbine undergoes the smallest amount of relocation over time since its incident wind speed is diminished by the velocity deficits of wakes 1 and 2.

Simulation scenario 2

The second simulation sinusoidally varies the yaw angles of the three turbines between ±20 deg with a period of 400 sec. Specifically, the following yaw angle expressions are used for $t \geq 1000\,\mathrm{sec}$:

$$\gamma_1(t) = \gamma_3(t) = (-20\,\mathrm{deg})\sin\left[\frac{2\pi}{400}(t - 1000\,\mathrm{sec})\right], \tag{4.50}$$

$$\gamma_2(t) = (+20\,\mathrm{deg})\sin\left[\frac{2\pi}{400}(t - 1000\,\mathrm{sec})\right]. \tag{4.51}$$

Velocity contours for this case are plotted in Figure 4.12. The sinusoidal yaw angle fluctuations cause oscillations of floating platforms in the $\hat{y}$ direction with the expected 400 sec excitation period. In terms of wake behaviour, the transport effect is clearly observed. As floating turbines shift in the $\hat{y}$ direction, the corresponding effects on their respective wakes are transported downstream at approximately 8 m/s. For instance, at $t = 1400\,\mathrm{sec}$, the leading turbine is located at a peak value past its neutral position in the $+\hat{y}$ direction (see the first vertical white dashed line in Figure 4.12). Given that $U_\infty = 8\,\mathrm{m/s}$, then 200 sec later, the centerline of the leading turbine's wake must peak in the $+\hat{y}$ direction at $\hat{x} = 8\,\mathrm{m/s} \times 200\,\mathrm{sec} = 1600\,\mathrm{m} = 12.7D$. Observing the velocity contours 200 sec later at $t = 1600\,\mathrm{sec}$, such a peak is observed at just under $\hat{x} = 12D$ (see the second vertical white dashed line in Figure 4.12).

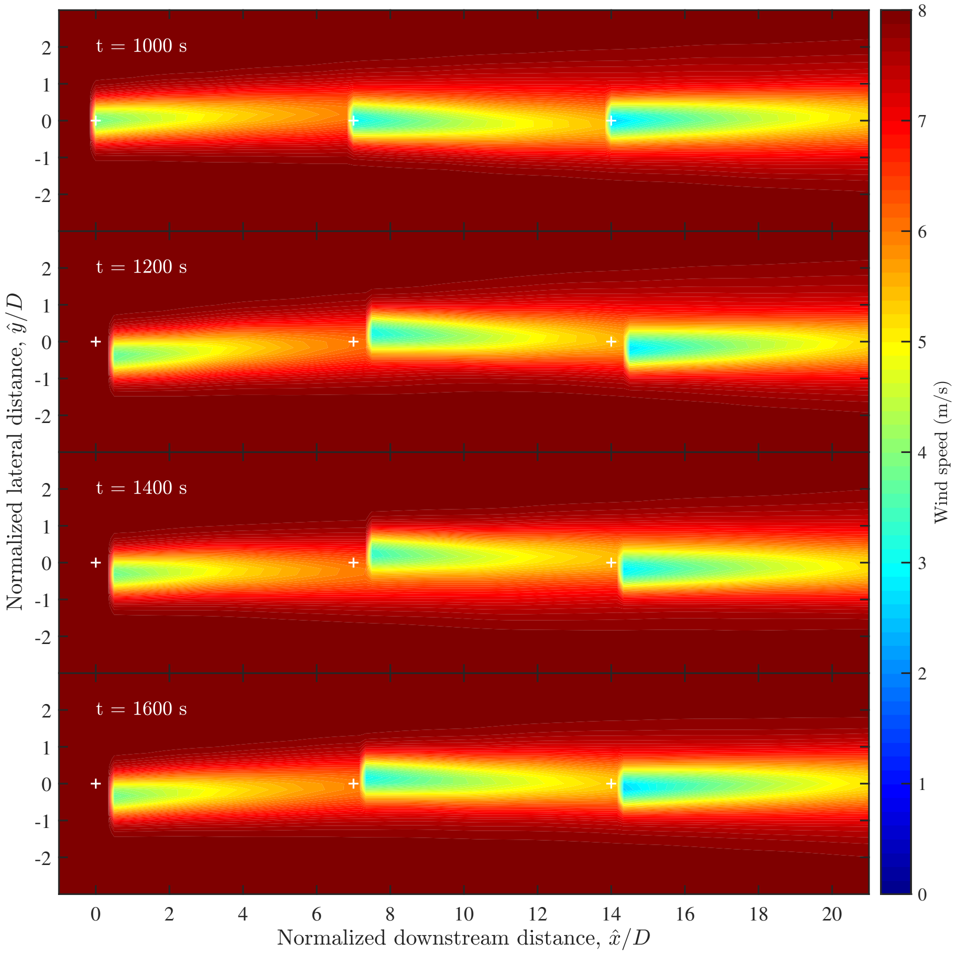


Figure 4.11: Velocity contours at various time-steps of simulation scenario 1 (*i.e.* fixed wind condition and turbine operating parameters, while platform motion is permitted). The white + symbols represent the neutral positions of the floating platforms. All floating platforms are held fixed at their respective neutral positions for the first 1000 sec of simulation. Simulation parameters: $U_\infty(t) = 8\,\mathrm{m/s}$, $V_\infty(t) = 0\,\mathrm{m/s}$, $a_1(t) = a_2(t) = a_3(t) = 1/3$, $\gamma_1(t) = \gamma_3(t) = -20\,\mathrm{deg}$ and $\gamma_2(t) = +20\,\mathrm{deg}$, $k_x = 0.08$, $\sigma = 0.025\hat{x} + 0.396\,\mathrm{m}$.

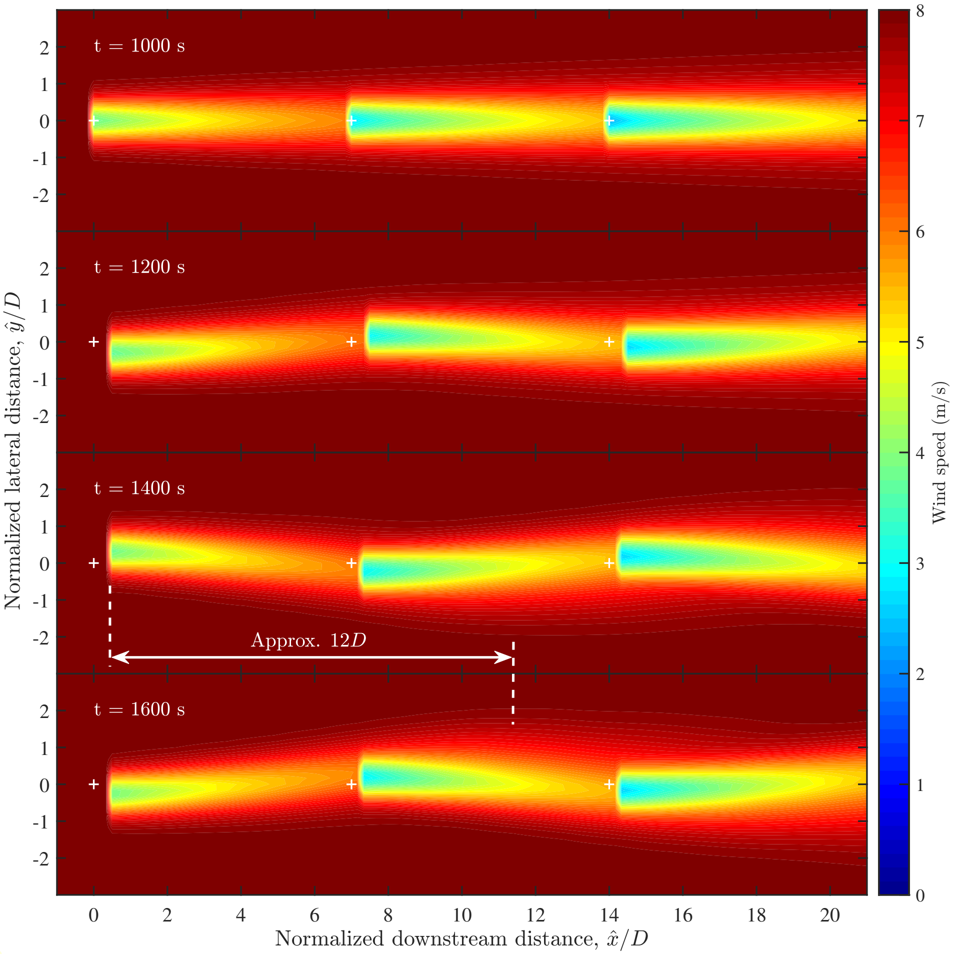


Figure 4.12: Velocity contours at various time-steps of simulation scenario 2 (*i.e.* fixed wind condition and sinusoidally varying yaw angles, while platform motion is permitted). The white + symbols represent the neutral positions of the floating platforms. All floating platforms are held fixed at their respective neutral positions for the first 1000 sec of simulation. Simulation parameters: $U_\infty(t) = 8\,\mathrm{m/s}$, $V_\infty(t) = 0\,\mathrm{m/s}$, $a_1(t) = a_2(t) = a_3(t) = 1/3$, $\gamma_1(t)$ and $\gamma_3(t)$ defined in Eq. (4.50) and $\gamma_2(t)$ defined in Eq. (4.51), $k_x = 0.08$, $\sigma = 0.025\hat{x} + 0.396\,\mathrm{m}$.

Simulation scenario 3

The third scenario assesses the impacts of time-varying wind direction, which is modeled by maintaining $U_\infty(t) = 8\,\mathrm{m/s}$ and fluctuating $V_\infty(t)$ sinusoidally between $\pm 2\,\mathrm{m/s}$ with a period of 200 sec. Specifically, $V_\infty(t)$ is expressed as follows for $t \geq 1000\,\mathrm{sec}$:

$$V_\infty(t) = (2\,\mathrm{m/s}) \sin\left[\frac{2\pi}{200}(t - 1000\,\mathrm{sec})\right]. \tag{4.52}$$

All yaw angles in this scenario are maintained at $\gamma_1(t) = \gamma_2(t) = \gamma_3(t) = 0\,\mathrm{deg}$. Velocity contours for simulation case 3 are shown in Figure 4.13. The notable expectation here is that, as the wind direction changes, wake centerlines are transported in tandem with the free stream wind in both $\hat{x}$ and $\hat{y}$ directions. For instance, at $t = 1000\,\mathrm{sec}$, the centerline of wake 1 is aligned with the $\hat{x}$ axis since $\gamma_1(t) = 0\,\mathrm{deg}$ and $V_\infty(t)$ had been equal to zero at all previous times. By $t = 1050\,\mathrm{sec}$, the effects of turbine 1 on the wind field should only be transported downstream by a distance of $8\,\mathrm{m/s} \times 50\,\mathrm{sec} = 400\,\mathrm{m} = 3.2D$. Therefore, for $\hat{x} < 3.2D$, we expect variations in the curvature of the centerline of wake 1 due to the presence of turbine 1, while for $\hat{x} > 3.2D$, this curvature should remain unchanged. Instead, for $\hat{x} > 3.2D$, the centerline of wake 1 should be shifted in the $+\hat{y}$ direction as a result of $V_\infty(t)$ having held positive values for the past 50 sec. Observing velocity contours at $t = 1050\,\mathrm{sec}$, it is evident that the centerline curvature of wake 1 remains flat at all downstream distances past approximately $\hat{x} = 3D$, while having been shifted in the $+\hat{y}$ direction.

4.4 Conclusions and recommendations for future research

This paper extended FOWFSim [103], which is a steady-state modeling tool that may be used for simulating and optimizing floating offshore wind farms, by adding capabilities that captured time-varying free stream wind velocities and floating platform motion. In addition to presenting a mathematical formulation, we performed a mesh convergence study and validated FOWFSim-Dyn predictions related to platform motion and steady wake transport. It was demonstrated that the limited number of tunable parameters produced wake centerline deflection and ve-

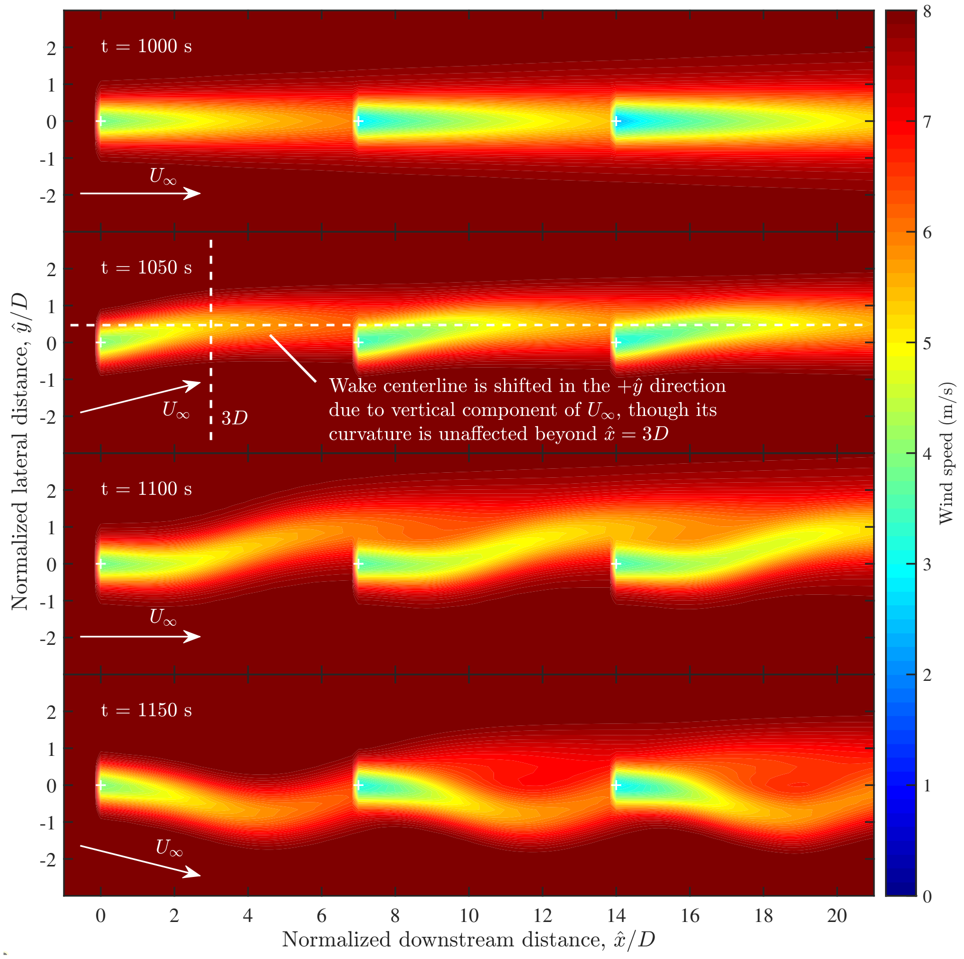


Figure 4.13: Velocity contours at various time-steps of simulation scenario 3 (*i.e.* fixed turbine operating conditions and fluctuating wind speed in the $\hat{y}$ direction, while platform motion is prohibited). The white $+$ symbols represent the neutral positions of the floating platforms. The white arrows denote the free stream wind direction. All floating platforms are held fixed throughout the simulation. Simulation parameters: $U_\infty(t) = 8\,\mathrm{m/s}$, $V_\infty(t)$ defined in Eq. (4.52), $a_1(t) = a_2(t) = a_3(t) = 1/3$, $\gamma_1(t) = \gamma_2(t) = \gamma_3(t) = 0\,\mathrm{deg}$, $k_x = 0.08$, $\sigma = 0.025\hat{x} + 0.396\,\mathrm{m}$.

locity deficit results that matched experimental observations with reasonable similarity for engineering analysis. We then conducted simulations under various wind and turbine operating conditions to assess the dynamic behavior of FOWFSim-Dyn. It was observed that FOWFSim-Dyn captures dynamic floating wind farm phenomena such as wake transport, time-varying wind speed and direction effects, and floating platform motion in line with physical reasoning and intuition.

For the purposes of further developing and enhancing the current framework, several recommendations on potential research directions are made. First, to this date, no LES-based wind farm simulators are capable of capturing floating platform motion. Developing wind farm CFD tools that consider such dynamics would therefore permit more comprehensive validation of FOWFSim-Dyn predictions pertaining to both platform motion and wake behaviour. Complementing this point, scaled wind tunnel experiments of floating wind turbines would also enable validation of dynamic FOWFSim-Dyn predictions.

Second, we did not model turbulence in the current framework. This feature may be incorporated by adding measurement noise to model outputs, or by including temporally and spatially distributed turbulence acceleration terms in the equations of motion. Finally, additional force gradients may be included in the equations of motion to capture complex wake phenomena such as secondary steering [60] and wake deflection due to rotor rotation [72].

Chapter 5

Distributed Economic Model Predictive Control – Addressing Non-convexity Using Social Hierarchies

This chapter introduces a novel concept for addressing non-convexity in the cost functions of distributed economic model predictive control (DEMPC) systems. Specifically, the proposed algorithm enables agents to self-organize into a hierarchy which determines the order in which control decisions are made. This concept is based on the formation of social hierarchies in nature. An additional feature of the algorithm is that it does not require stationary set-points that are known *a priori*. Rather, agents negotiate these targets in a truly distributed and scalable manner. Upon providing a detailed description of the algorithm, guarantees of convergence, recursive feasibility, and bounded closed-loop stability are also provided. Finally, the proposed algorithm is compared against a basic parallel distributed economic model predictive controller using an academic numerical example.

5.1 Introduction

5.1.1 Background

Model predictive control (MPC) entails recursively solving an optimization problem over a finite prediction horizon to identify optimal future control input trajectories. The popularity of MPC in academic and industrial environments is primarily attributed to its capacity for handling constraints while computing control actions that minimize nonlinear performance criteria. The reader may refer to articles by Mayne *et al.* [125] and Mayne [126] for reviews on MPC.

In large-scale processes or multi-agent systems, implementation of MPC in a centralized manner may be impractical due to the computational complexity of global optimization and the network infrastructure required for plant-wide communication. Distributed model predictive control (DMPC) surpasses these limitations by dispersing the burden of decision-making across a multitude of independent subsystems or agents. A trade-off that arises however, is that effective algorithms governing agent coordination are required to guarantee desirable closed-loop performance. The reader may refer to review articles by Al-Gherwi *et al.* [5], Christofides *et al.* [38], and Negenborn and Maestre [140] for further details on the subject of DMPC.

MPC has traditionally been utilized as a lower-level regulator and stabilizer that tracks set-points determined by upper-level stationary optimizers. Economic model predictive control (EMPC) combines these upper- and lower-level roles by employing cost functions that capture plant economics (*e.g.* power production or operating cost over a finite time horizon). The effect is improved economic performance; however, additional measures for ensuring stability are required since the primary control objective no longer involves regulation. The reader may refer to articles by Ellis *et al.* [55] and Müller and Allgöwer [134] for reviews on EMPC.

5.1.2 Distributed economic model predictive control

This chapter addresses distributed economic model predictive control (DEMPC) of systems with non-convex objective functions and unknown stationary set-points. Applications with such characteristics include autonomous vehicle trajectory planning [54] and floating offshore wind farm control (see Chapter 6). DEMPC algorithms intended for such systems have been scarce in the literature as a result of challenges pertaining to stability and convergence. This subsection reviews relevant DEMPC and nonlinear DMPC algorithms as justification for the contributions of the current work.

Stabilizing DEMPC algorithms

Achieving stability in DEMPC requires first computing optimal stationary set-points for all agents, and then constraining state trajectories to approach these optima within the prediction horizon. If there exist feedback control laws that are then capable of maintaining subsystems within specified bounds of their respective steady-states, stability may be guaranteed. To achieve such an outcome, theoretical studies focused on DEMPC have either treated these stationary set-points as predefined references [184], or computed their values using centralized optimization [6, 36, 50, 111, 112, 185]. The latter group of algorithms are therefore not truly distributed.

To overcome this gap, Köhler *et al.* [108] were the first to develop a DEMPC scheme without the requirement for centralized processing. They presumed that optimal stationary set-points were unattainable via centralized optimization, and instead had to be negotiated online between agents in a distributed manner. Consequently, in tandem with solving their local EMPC problems and obtaining optimal input trajectories, agents also performed one iterate of a distributed coordination algorithm at each sampling time to update their respective optimal steady-states. Nonetheless, this work focused on linear systems with convex cost functions and used a sequential coordination algorithm [109, 151]; thus suffering from lack of scalability.

Convergent DEMPC algorithms

If DEMPC cost functions are non-convex, agents making decisions in parallel cannot guarantee convergence of their optimal input trajectories [116]. Several alternative classes of coordination algorithms within the nonlinear DMPC literature address this convergence issue. Sequential methods first proposed by Kuwata *et* *al.* [109] and Richards and How [151] represent the simplest solution. Agents solve their local optimization problems and exchange information with their neighbors in some predetermined order. The resulting advantage is that each subsequent agent computes its input trajectory based on updated and fixed information from its predecessors; guaranteeing convergence, stability, and feasibility is thus facilitated. The major drawback is lack of scalability to large interconnected systems, since agents at the tail-end of sequence must await decisions from all other subsystems. A secondary concern involves predetermining the sequence order, particularly in systems with time-varying interaction topologies.

Coordination algorithms based on negotiation between agents were developed by Müller *et al.* [135], Maestre *et al.* [120], and Stewart *et al.* [170]. An agent receives optimal decisions from its neighbors in the form of a proposal. Then, upon computing the corresponding effects of these decisions on its local objective function, the agent may reject or approve proposals. These algorithms are capable of resolving conflict; however they face two limitations. The first is that, in order to identify the impact of a specific agent's control trajectory on neighboring cost functions, this agent must not operate in parallel with others; thus limiting scalability. The second is that agents whose control actions are discarded at particular time-steps remain idle. Finally, these algorithms possess no learning mechanism to ensure that, after a sufficient number of negotiations, proposals are guaranteed or more likely to be approved.

Finally, group-based DMPC methods employ the connectivity information of a plant to identify the order in which agents should solve their local MPC problems to resolve conflict. Pannek [144] proposed a covering algorithm that permitted non-interacting agent pairs to operate in parallel, while those that were coupled made decisions sequentially according to some predetermined priority rule. This algorithm eliminated the scalability issue of pure sequential DMPC; however it required a predetermined set of priority rules. Liu *et al.* [117] developed a clustering algorithm that assigned agents to dominant or connecting groups. Agents in dominant clusters solved their local optimization problems first, thus eliminating conflict with agents in connecting groups. The downside in this method was that a sequential algorithm was required to determine clustering. Asadi and Richards [12] employed a slot allocation algorithm wherein each agent communicated with other subsystems to randomly select an available space in the global sequential order. This method addressed the secondary drawback of sequential DMPC, which concerned determining an effective sequence order in systems with time-varying interaction topologies. Nonetheless, the fully serial nature of the algorithm still suffered from lack of scalability.

5.1.3 Contributions

Based on the preceding literature review, we state, to the best of our knowledge, that a DEMPC algorithm that handles non-convex cost functions and unknown stationary set-points in a scalable and truly distributed manner, with no predetermined rules, has yet to be proposed. The existing method that meets all of these criteria except for scalability and non-convexity is the algorithm of Köhler *et al.* [108].

The main contribution in this chapter is thus a DEMPC coordination algorithm that is scalable, fully distributed, and that guarantees stability and convergence in the presence of non-convex cost functions and unknown stationary set-points. In brief, our approach borrows from the method of conflict resolution observed in nature. Namely, when it becomes apparent that agents operating in parallel generate conflicting decisions, a social hierarchy is established to yield resolution. Additionally, proofs of convergence, recursive feasibility, and bounded closed-loop stability are provided along with validation using a numerical example.

5.1.4 Chapter organization

The remainder of this chapter is organized as follows: Section 5.2 provides a description of the nonlinear systems and cost functions that the proposed algorithm addresses, along with an explanation of conflict and convergence issues arising from non-convex objectives; Section 5.3 highlights the proposed DEMPC algorithm along with proofs of convergence, feasibility, and stability; Section 5.4 implements the proposed method on a numerical example with non-convex cost functions; and finally, Section 5.5 concludes the chapter with a summary of major findings, along with recommendations for future research directions.

5.2 Problem description

5.2.1 Notation

This brief subsection introduces the reader to the set theory and other notation used in this work. The term $\mathbb{I}_{a:b}$ indicates a set of real integers ranging from a to b. The symbols $\mathbf{x} \in \mathbb{R}^n$ state that $\mathbf{x}$ is a real-valued vector of dimensions $n \times 1$. The expression $\mathscr{A} \setminus \mathscr{B}$ denotes the difference between the sets $\mathscr{A}$ and $\mathscr{B}$ (*i.e.* the set $\mathscr{A}$ with all elements of set $\mathscr{B}$ removed). The operation $\mathscr{A} \times \mathscr{B}$ yields the Cartesian product of the sets $\mathscr{A}$ and $\mathscr{B}$.

5.2.2 Dynamic model

We consider N agents that are dynamically decoupled and uninfluenced by disturbances. The dynamics of each agent $i \in \mathscr{I} = \{1, 2, \cdots, N\}$ are represented by the following discrete-time nonlinear state-space model:

$$\mathbf{x}_i^+ = \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i), \tag{5.1}$$

where $\mathbf{x}_i \in \mathbb{R}^{n_i}$ and $\mathbf{u}_i \in \mathbb{R}^{m_i}$ denote vectors containing the n_i states and m_i inputs of agent i, and $\mathbf{x}_i^+$ represents $\mathbf{x}_i$ at the subsequent sampling time-step. We consider the case where $\mathbf{x}_i$ and $\mathbf{u}_i$ must be bounded within the convex sets $\mathscr{X}_i$ and $\mathscr{U}_i$ at all times, which results in the following state and input constraints:

$$\begin{aligned} \mathbf{u}_i &\in \mathscr{U}_i, & (5.2)\\ \mathbf{x}_i &\in \mathscr{X}_i. & (5.3) \end{aligned}$$

With these operational bounds defined, we make the following assumptions concerning controllability and continuity.

Assumption 1. *(Weak controllability) Let the set $\mathscr{Z}_i^s$ comprise all feasible stationary points of agent i as follows:*

$$\mathscr{Z}_i^s := \{(\mathbf{x}_i, \mathbf{u}_i) \in \mathscr{X}_i \times \mathscr{U}_i \mid \mathbf{x}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i)\}. \tag{5.4}$$

All feasible stationary state vectors of agent i may then be collected within the set

$\mathscr{X}_i^s$, which is defined as follows:

$$\mathscr{X}_i^s := \{\mathbf{x}_i \in \mathscr{X}_i \mid \exists \mathbf{u}_i \in \mathscr{U}_i : (\mathbf{x}_i, \mathbf{u}_i) \in \mathscr{Z}_i^s\}. \tag{5.5}$$

Let the set $\mathscr{Z}_i^{0\to s}$ contain all pairings of initial state vectors $\mathbf{x}_i^0$ and input trajectories $\overline{\mathbf{u}}_i = \left(\mathbf{u}_i^0, \mathbf{u}_i^1, \cdots, \mathbf{u}_i^{H-1}\right)$ that steer agent i to each feasible stationary point $\mathbf{x}_i^s$ in H time-steps, while satisfying constraints. $\mathscr{Z}_i^{0\to s}$ may therefore be defined as follows:

$$\begin{aligned}\mathscr{Z}_i^{0\to s} := \big\{ &(\mathbf{x}_i^0, \overline{\mathbf{u}}_i, \mathbf{x}_i^s) \in \mathscr{X}_i \times \overline{\mathscr{U}}_i \times \mathscr{X}_i^s \mid \\ &\exists \mathbf{x}_i^1, \mathbf{x}_i^2, \cdots, \mathbf{x}_i^H : \mathbf{x}_i^k = \mathbf{f}_i(\mathbf{x}_i^{k-1}, \mathbf{u}_i^{k-1}), \\ &\mathbf{x}_i^k \in \mathscr{X}_i, \forall k \in \mathbb{I}_{1:H}, \mathbf{x}_i^H = \mathbf{x}_i^s \Big\},\end{aligned} \tag{5.6}$$

where $\overline{\mathscr{U}}_i = \mathscr{U}_i \times \cdots \times \mathscr{U}_i = \mathscr{U}_i^H$. All possible initial state vectors $\mathbf{x}_i^0$ that may be steered to a feasible stationary point $\mathbf{x}_i^s$, with constraint satisfaction, are then contained within the set $\mathscr{X}_i^{0\to s}$ defined as follows:

$$\mathscr{X}_i^{0\to s} := \left\{\mathbf{x}_i^0 \in \mathscr{X}_i \mid \exists \overline{\mathbf{u}}_i \in \overline{\mathscr{U}}_i, \mathbf{x}_i^s \in \mathscr{X}_i^s : \left(\mathbf{x}_i^0, \overline{\mathbf{u}}_i, \mathbf{x}_i^s\right) \in \mathscr{Z}_i^{0\to s}\right\}. \tag{5.7}$$

For any agent $i \in \mathscr{I}$, any initial state vector $\mathbf{x}_i^0 \in \mathscr{X}_i^{0\to s}$, input vector trajectory $\overline{\mathbf{u}}_i \in \overline{\mathscr{U}}_i$, and stationary state vector $\mathbf{x}_i^s \in \mathscr{X}_i^s$ such that $\left(\mathbf{x}_i^0, \overline{\mathbf{u}}_i, \mathbf{x}_i^s\right) \in \mathscr{Z}_i^{0\to s}$, and any stationary input vector $\mathbf{u}_i^s \in \mathscr{U}_i$ such that $(\mathbf{x}_i^s, \mathbf{u}_i^s) \in \mathscr{Z}_i^s$, there exists a $\mathscr{K}_\infty$ function $\gamma(\cdot)$ that satisfies the following condition:

$$\sum_{k=0}^{H-1} \|\mathbf{u}_i^k - \mathbf{u}_i^s\| \le \gamma(\|\mathbf{x}_i^0 - \mathbf{x}_i^s\|). \tag{5.8}$$

Remark. *The weak controllability assumption simply states that, for any feasible stationary point, there exists some surrounding set from which an initial state vector may be steered to the stationary point. This assumption is necessary for guaranteeing feasibility of the optimization problem of the DEMPC algorithm since reaching a stationary target is one of its constraints. Therefore, if an input trajectory exists that can steer an initial state vector to a stationary point, then a solution to the optimization problem exists that satisfies its constraints.*

Assumption 2. *(Lipschitz continuous dynamics) For any agent $i \in \mathscr{I}$, $\mathbf{f}_i(\cdot)$ satisfies the following condition for Lipschitz continuity for all $(\mathbf{x}_i^a, \mathbf{u}_i^a), (\mathbf{x}_i^b, \mathbf{u}_i^b) \in \mathscr{X}_i \times \mathscr{U}_i$:*

$$\left\| \mathbf{f}_i(\mathbf{x}_i^b, \mathbf{u}_i^b) - \mathbf{f}_i(\mathbf{x}_i^a, \mathbf{u}_i^a) \right\| \leq \Lambda_i^f \left\| (\mathbf{x}_i^b, \mathbf{u}_i^b) - (\mathbf{x}_i^a, \mathbf{u}_i^a) \right\|, \tag{5.9}$$

where the scalar $\Lambda_i^f \geq 0$ is the Lipschitz constant of $\mathbf{f}_i(\cdot)$ on the set $\mathscr{X}_i \times \mathscr{U}_i$.

Remark. *Lipschitz continuity simply states that the function $\mathbf{f}_i(\cdot)$ must be continuous. In other words, there must exist no discontinuities along $\mathbf{f}_i(\cdot)$ that lead to an undefined gradient. This assumption is necessary for guaranteeing optimality in the solution of an optimization problem. If gradients are defined, a local minimum of a cost function will be reached after a sufficient number of iterations.*

5.2.3 Control objective

At each time-step, the control objective of agent i is to minimize a cooperative economic stage cost function $J_i(\cdot)$ over a finite prediction horizon H as follows:

$$\min \sum_{k=0}^{H-1} J_i(\mathbf{x}_i^k, \mathbf{u}_i^k, \mathbf{x}_{-i|J}^k, \mathbf{u}_{-i|J}^k), \tag{5.10}$$

where the superscript k identifies the time-step number along the prediction horizon H, $\mathbf{x}_i^k$ and $\mathbf{u}_i^k$ denote the state and input vectors of agent i at time-step k along the prediction horizon, and $\mathbf{x}_{-i|J}^k$ and $\mathbf{u}_{-i|J}^k$ contain the state and input vectors at time-step k along the prediction horizon of all agents $j \in \mathscr{I} \setminus i$ that influence the cooperative cost function $J_i(\cdot)$ of agent i. We collect the indices of these agents into the set $\mathscr{N}_{-i|J}$. Likewise, the indices of all agents $j \in \mathscr{I} \setminus i$ whose cooperative stage cost functions $J_j(\cdot)$ are influenced by $\mathbf{x}_i$ and $\mathbf{u}_i$ are collected into the set $\mathscr{N}_{+i|J}$.

The objective function $J_i(\cdot)$ may be non-convex; however, it must adhere to the following assumptions concerning cooperation, boundedness, and continuity.

Assumption 3. *(Neighborhood-cooperative objectives) Let each agent $i \in \mathscr{I}$ possess a stage cost function $\ell_i(\cdot)$ that represents its local economic interests. Then, let the set $\mathscr{N}_{-i}$ contain the indices of all agents $j \in \mathscr{I} \setminus i$ whose state and input vectors $\mathbf{x}_j$ and $\mathbf{u}_j$ influence the local stage cost function $\ell_i(\cdot)$. Likewise, let the set*

$\mathscr{N}_{+i}$ contain the indices of all agents $j \in \mathscr{I} \setminus i$ whose local stage cost functions $\ell_j(\cdot)$ are influenced by $\mathbf{x}_i$ and $\mathbf{u}_i$.

The stage cost function $J_i(\cdot)$ for any agent $i \in \mathscr{I}$ is neighborhood-cooperative in that it comprises the local interests of agent i and those of each downstream neighbor $j \in \mathscr{N}_{+i}$ as follows:

$$J_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{x}_{-i|J}, \mathbf{u}_{-i|J}) := \ell_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{x}_{-i}, \mathbf{u}_{-i}) + \sum_{j \in \mathscr{N}_{+i}} \ell_j(\mathbf{x}_j, \mathbf{u}_j, \mathbf{x}_{-j}, \mathbf{u}_{-j}). \tag{5.11}$$

The vectors $\mathbf{x}_{-i}$ and $\mathbf{u}_{-i}$ contain the states and inputs of all agents $j \in \mathscr{N}_{-i}$. Note that $\mathscr{N}_{-i|J} = \mathscr{N}_{-i} \cup \mathscr{N}_{+i} \cup \mathscr{N}_{-j} \forall j \in \mathscr{N}_{+i}$.

Remark. *The purpose of the social hierarchy-based DEMPC algorithm presented in this work is to enable coupled agents to compute optimal decisions that are mutually beneficial. A fundamental requirement for this algorithm is therefore that coupled agents share interests; hence the assumption of neighborhood-cooperative cost functions. If this assumption is not present, then the agents are competitive, and mutually beneficial decisions cannot be guaranteed.*

Assumption 4. *(Bounded cost function minima) Let the sets $\mathscr{X}_{-i}$ and $\mathscr{U}_{-i}$ be defined as follows:*

$$\mathscr{X}_{-i} := \prod_{j \in \mathscr{N}_{-i}} \mathscr{X}_j, \tag{5.12}$$

$$\mathscr{U}_{-i} := \prod_{j \in \mathscr{N}_{-i}} \mathscr{U}_j, \tag{5.13}$$

For any agent $i \in \mathscr{I}$, there exist state and input vectors $\left(\mathbf{x}_i^*, \mathbf{u}_i^*, \mathbf{x}_{-i}^*, \mathbf{u}_{-i}^*\right) \in \mathscr{X}_i \times \mathscr{U}_i \times \mathscr{X}_{-i} \times \mathscr{U}_{-i}$ such that the following condition holds for all $(\mathbf{x}_i, \mathbf{u}_i, \mathbf{x}_{-i}, \mathbf{u}_{-i}) \in \mathscr{X}_i \times \mathscr{U}_i \times \mathscr{X}_{-i} \times \mathscr{U}_{-i}$:

$$\ell_i(\mathbf{x}_i^*, \mathbf{u}_i^*, \mathbf{x}_{-i}^*, \mathbf{u}_{-i}^*) \leq \ell_i(\mathbf{x}_i, \mathbf{u}_i, \mathbf{x}_{-i}, \mathbf{u}_{-i}). \tag{5.14}$$

Remark. *The bounded cost function minima assumption is necessary for guaranteeing convergence of optimization problems. If the global minimum of a cost function is finite, then an optimization algorithm that descends along the gradients* *of the cost function is guaranteed to reach a point that satisfies optimality conditions after a sufficient number of iterations.*

Assumption 5. *(Lipschitz continuous objectives) For any agent $i \in \mathscr{I}$, the local cost function $\ell_i(\cdot)$ satisfies the following condition for Lipschitz continuity for all $(\mathbf{x}_i^a, \mathbf{u}_i^a, \mathbf{x}_{-i}^a, \mathbf{u}_{-i}^a), (\mathbf{x}_i^b, \mathbf{u}_i^b, \mathbf{x}_{-i}^b, \mathbf{u}_{-i}^b) \in \mathscr{X}_i \times \mathscr{U}_i \times \mathscr{X}_{-i} \times \mathscr{U}_{-i}$:*

$$\begin{aligned} &\left\| \ell_i(\mathbf{x}_i^b, \mathbf{u}_i^b, \mathbf{x}_{-i}^b, \mathbf{u}_{-i}^b) - \ell_i(\mathbf{x}_i^a, \mathbf{u}_i^a, \mathbf{x}_{-i}^a, \mathbf{u}_{-i}^a) \right\| \leq \\ &\qquad \Lambda_i^\ell \left\| (\mathbf{x}_i^b, \mathbf{u}_i^b, \mathbf{x}_{-i}^b, \mathbf{u}_{-i}^b) - (\mathbf{x}_i^a, \mathbf{u}_i^a, \mathbf{x}_{-i}^a, \mathbf{u}_{-i}^a) \right\|, \end{aligned} \tag{5.15}$$

where the scalar $\Lambda_i^\ell \geq 0$ is the Lipschitz constant of $\ell_i(\cdot)$ on the set $\mathscr{X}_i \times \mathscr{U}_i \times \mathscr{X}_{-i} \times \mathscr{U}_{-i}$.

Remark. *Refer to the remark of Assumption 2 for a simplified explanation of Lipschitz continuity.*

5.2.4 Conflict under non-convexity

In this subsection, we elaborate further on the main challenge that is associated with non-convex cost functions in DEMPC. Consider a simple problem with only two optimization variables z_1 and z_2, which are computed by agents 1 and 2, respectively. Further, let both agents share a common non-convex global objective function with contours plotted in Figure 5.1.

Assume initial values z_1^0 and z_2^0 obtained from a previous iteration or time-step. Under parallel and fully distributed operation, each agent must assume that its neighbors optimization variable remains unchanged while locally minimizing the global objective function. As a result, agent 1 assumes that z_2 remains fixed at z_2^0 and restricts its search path to the horizontal orange line shown in Figure 5.1. Likewise, agent 2 assumes that z_1 is maintained at z_1^0, which constrains its search path to the vertical orange line.

Upon completion of its local optimization problem, agent 1 finds the local optimum located at $\left(\breve{z}_1, z_2^0\right)$. Agent 2 achieves the same at $\left(z_1^0, \breve{z}_2\right)$. When the updated optimal variables $\breve{z}_1$ and $\breve{z}_2$ are combined however, the overall system operates at neither of the local optima identified by the individual agents. We refer to such

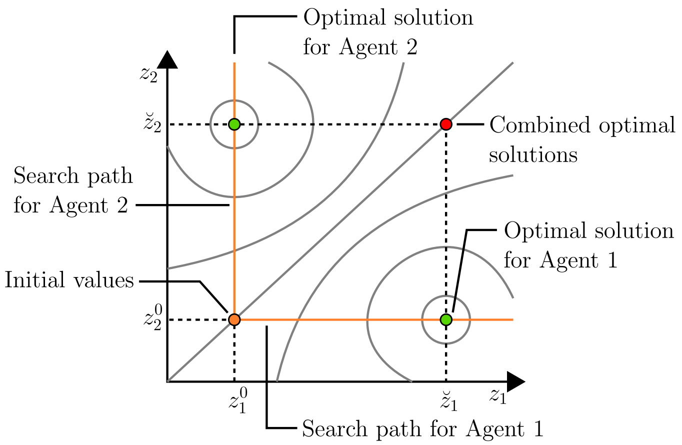


Figure 5.1: A visual explanation of conflict generated as a result of non-convexity in distributed parallel optimization.

an outcome as conflict in the current work. Specifically, we define conflict and conflict-free operation as follows.

Definition 1. *(Conflict) Agent i encounters conflict when its economic performance deteriorates upon considering the optimal control actions of its neighbors. More formally, consider $\hat{V}_i^s$ and $\breve{V}_i^s$ defined as follows:*

$$\hat{V}_i^s \quad := \quad J_i(\breve{\mathbf{x}}_i^s, \breve{\mathbf{u}}_i^s, \hat{\mathbf{x}}^s_{-i|J}, \hat{\mathbf{u}}^s_{-i|J}), \tag{5.16}$$

$$\breve{V}_i^s \quad := \quad J_i(\breve{\mathbf{x}}_i^s, \breve{\mathbf{u}}_i^s, \breve{\mathbf{x}}^s_{-i|J}, \breve{\mathbf{u}}^s_{-i|J}), \tag{5.17}$$

where $\breve{\mathbf{x}}_i^s$ and $\breve{\mathbf{u}}_i^s$ denote the optimal stationary state and input vectors computed by agent i, $\hat{\mathbf{x}}^s_{-i|J}$ and $\hat{\mathbf{u}}^s_{-i|J}$ contain stationary state and input vectors that agent i assumes for all neighbors $j \in \mathscr{N}_{-i|J}$, and $\breve{\mathbf{x}}^s_{-i|J}$ and $\breve{\mathbf{u}}^s_{-i|J}$ consist of optimal state and input vectors computed by all agents $j \in \mathscr{N}_{-i|J}$ and communicated to agent i. The terms $\hat{V}_i^s$ and $\breve{V}_i^s$ represent naive and informed values of the stage cost function of agent i at some stationary point, respectively. The term naive indicates that the cost function value is computed based on assumed values of neighboring agents' state and input vectors. Contrarily, the term informed is employed when agent i considers recently communicated updated optimal state and input vectors. Given these definitions, while attempting to negotiate an optimal stationary point $(\breve{\mathbf{x}}_i^s, \breve{\mathbf{u}}_i^s)$, *agent i operates in conflict with its neighbors if the following statement is true:*

$$\check{V}_i^s > \hat{V}_i^s. \tag{5.18}$$

Similarly, one may define naive and informed values of cost functions summed along the prediction horizon as follows:

$$\begin{aligned} \hat{V}_i &:= \sum_{k=0}^{H-1} J_i(\check{\mathbf{x}}_i^k, \check{\mathbf{u}}_i^k, \hat{\mathbf{x}}_{-i|J}^k, \hat{\mathbf{u}}_{-i|J}^k), & (5.19) \\ \check{V}_i &:= \sum_{k=0}^{H-1} J_i(\check{\mathbf{x}}_i^k, \check{\mathbf{u}}_i^k, \check{\mathbf{x}}_{-i|J}^k, \check{\mathbf{u}}_{-i|J}^k), & (5.20) \end{aligned}$$

where $\check{\mathbf{x}}_i^k$ and $\check{\mathbf{u}}_i^k$ denote optimal state and input vectors computed by agent i at time-step k along the prediction horizon, $\hat{\mathbf{x}}_{-i|J}^k$ and $\hat{\mathbf{u}}_{-i|J}^k$ contain state and input vectors that agent i assumes for all neighbors $j \in \mathscr{N}_{-i|J}$ at some time-step k along the prediction horizon, and $\check{\mathbf{x}}_{-i|J}^k$ and $\check{\mathbf{u}}_{-i|J}^k$ consist of optimal state and input vectors computed by all agents $j \in \mathscr{N}_{-i|J}$ at some time-step k along the prediction horizon and communicated to agent i. Given this information, while attempting to negotiate optimal state and input trajectories $\overline{\mathbf{x}}_i^* = \left(\check{\mathbf{x}}_i^0, \check{\mathbf{x}}_i^1, \cdots, \check{\mathbf{x}}_i^H\right)$ and $\overline{\mathbf{u}}_i^* = \left(\check{\mathbf{u}}_i^0, \check{\mathbf{u}}_i^1, \cdots, \check{\mathbf{u}}_i^{H-1}\right)$, agent i operates in conflict with its neighbors if the following statement is true:

$$\check{V}_i > \hat{V}_i. \tag{5.21}$$

Definition 2. *(Conflict-free operation) Agent i operates free of conflict when its economic performance improves or remains unchanged upon considering the optimal control actions of its neighbors. More formally, consider once again the values $\hat{V}_i^s$, $\check{V}_i^s$, $\hat{V}_i$, and $\check{V}_i$ as previously defined. While attempting to negotiate an optimal stationary point $(\check{\mathbf{x}}_i^s, \check{\mathbf{u}}_i^s)$, agent i operates free of conflict with its neighbors if the following statement is true:*

$$\check{V}_i^s \leq \hat{V}_i^s. \tag{5.22}$$

While attempting to negotiate optimal trajectories $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$, agent i operates free of conflict with its neighbors if the following statement is true:

$$\check{V}_i \leq \hat{V}_i. \tag{5.23}$$

In the introduction, several algorithms based on sequential operation, agent negotiation, and agent grouping that could resolve non-convex conflict were discussed. The main disadvantages of these algorithms were lack of scalability, idleness of certain agents, and the requirement of predefined rules. In the next section, we propose a fully distributed and scalable solution to the problem of conflict that addresses these drawbacks by establishing social hierarchies.

5.3 Social hierarchy-based DEMPC algorithm

5.3.1 Social hierarchy framework

In the current context, a social hierarchy consists of a finite number of levels that establish the sequence in which agents generate decisions in order to resolve conflict. The concept is based loosely on social hierarchies that appear naturally among living organisms as a means to resolve conflict and establish which individuals' decisions take priority over those of others. These hierarchies are often determined by the evolutionary or cultural characteristics of the individuals, which are, in essence, randomly assigned. In a similar fashion, we propose a framework which permits the formation of hierarchies with elements of randomness in order to resolve conflict resulting from non-convexity.

For generality, we assume an iterative parallel coordination algorithm, the first of which was presented by Du *et al.* [51]. Agents synchronously solve their local optimization problems and communicate repeatedly within a single sampling time-step until some termination condition is satisfied or until a maximum number of iterations have been implemented. The socially hierarchy framework is equally applicable to non-iterative parallel methods which were first investigated by Jia and Krogh [92] and Dunbar and Murray [53].

A visual representation of the social hierarchy framework is shown in Figure 5.2. Within a single iteration, there exist N_q hierarchy levels which specify the order in which agents make decisions. Each agent may solve for its optimal stationary point and control trajectory only once within an iteration; however, this computation may take place within any hierarchy level. During each iteration, agents occupying hierarchy level $q = 1$ make decisions first and transfer relevant

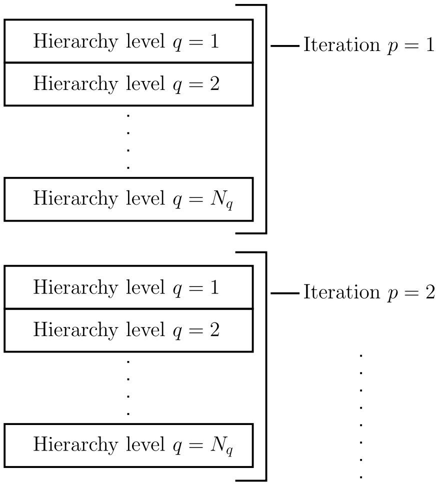


Figure 5.2: Schematic demonstrating the organization structure of the social hierarchy framework.

information to their neighbors. Following this step, agents allocated to hierarchy level $q = 2$ perform the same task. This trend continues until all hierarchy level computations have been performed, at which point the entire process is repeated during the next iteration. It is important to note that multiple agents may occupy the same hierarchy level, and that N_q may be substantially smaller than N.

Two fundamental questions now arise regarding (i) how agents should sort themselves among the N_q hierarchy levels in order to resolve conflict, and (ii) how should N_q be determined by the control system designer. The former concern is addressed in Section 5.3.2, which describes and assesses a novel DEMPC coordination algorithm that utilizes the concept of a social hierarchy. The latter question is discussed in Section 5.3.4, which burrows elements from vertex coloring theory to establish social hierarchy properties.

5.3.2 DEMPC coordination algorithm

This subsection details a novel coordination algorithm for DEMPC with non-convex objectives based on the social hierarchy framework described in Section 5.3.1. Our approach allows agents to resolve their conflicts in a truly distributed and scalable manner without the requirement of predefined rules or access to the full system interaction topology. Each agent achieves this outcome by occupying an appropriate level along a social hierarchy when conflict arises. In brief, our algorithm follows evolutionary principles. That is, if an agent occupying a particular hierarchy level during some iteration experiences conflict, then its current hierarchy level is detrimental to its performance and must be randomly mutated.

While negotiating optimal trajectories $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$, agent i solves the following optimization problem during each iteration:

$$\min_{\overline{\mathbf{u}}_i,\overline{\mathbf{x}}_i} \sum_{k=0}^{H-1} J_i(\mathbf{x}_i^k, \mathbf{u}_i^k, \hat{\mathbf{x}}_{-i|J}^k, \hat{\mathbf{u}}_{-i|J}^k), \tag{5.24}$$

subject to

$$\begin{aligned} \mathbf{x}_i^0 &= \mathbf{x}_i, && (5.25a)\\ \mathbf{x}_i^{k+1} &= \mathbf{f}_i(\mathbf{x}_i^k, \mathbf{u}_i^k), && (5.25b)\\ \mathbf{x}_i^k &\in \mathscr{X}_i, && (5.25c)\\ \mathbf{u}_i^k &\in \mathscr{U}_i, && (5.25d)\\ \mathbf{x}_i^H &= \mathbf{x}_i^s, && (5.25e) \end{aligned}$$

where $\mathbf{x}_i^k$ and $\mathbf{u}_i^k$ denote the candidate state and input vectors of agent i at some time-step k along the prediction horizon H, $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{u}}_i$ represent candidate state and input vector trajectories as follows:

$$\begin{aligned} \overline{\mathbf{x}}_i &:= \left(\mathbf{x}_i^0, \mathbf{x}_i^1, \cdots, \mathbf{x}_i^H\right), && (5.26)\\ \overline{\mathbf{u}}_i &:= \left(\mathbf{u}_i^0, \mathbf{u}_i^1, \cdots, \mathbf{u}_i^{H-1}\right), && (5.27) \end{aligned}$$

and $\hat{\mathbf{x}}_{-i|J}^k$ and $\hat{\mathbf{u}}_{-i|J}^k$ contain the state and input vectors that agent i assumes for all neighbors $j \in \mathscr{N}_{-i|J}$ at time-step k. For notational brevity in Algorithm 1, we condense the assumed states and input of agent j into the trajectories $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{u}}_j$ as follows:

$$\begin{aligned} \tilde{\mathbf{x}}_j &:= \left(\hat{\mathbf{x}}_j^0, \hat{\mathbf{x}}_j^1, \cdots, \hat{\mathbf{x}}_j^H\right), && (5.28)\\ \tilde{\mathbf{u}}_j &:= \left(\hat{\mathbf{u}}_j^0, \hat{\mathbf{u}}_j^1, \cdots, \hat{\mathbf{u}}_j^{H-1}\right). && (5.29) \end{aligned}$$

Solving Problem (5.24) yields the optimal state and input vector trajectories $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$ defined as follows:

$$\overline{\mathbf{x}}_i^* \;:=\; \left(\breve{\mathbf{x}}_i^0, \breve{\mathbf{x}}_i^1, \cdots, \breve{\mathbf{x}}_i^H\right), \tag{5.30}$$

$$\overline{\mathbf{u}}_i^* \;:=\; \left(\breve{\mathbf{u}}_i^0, \breve{\mathbf{u}}_i^1, \cdots, \breve{\mathbf{u}}_i^{H-1}\right), \tag{5.31}$$

where $\breve{\mathbf{x}}_i^k$ and $\breve{\mathbf{u}}_i^k$ denote the optimal state and input vectors at time-step k along the prediction horizon. The difference between *candidate* and *optimal* solutions is that the optimal solutions $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$ are not immediately *accepted* by the DEMPC algorithm. In other words, if it was required that DEMPC algorithm implement a decision immediately, the enacted action would follow the candidate trajectories $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{u}}_i$, not the optimal solutions $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$. As detailed in Algorithm 1, $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$ are only accepted as candidate solutions if they do not yield conflict. If $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$ do yield conflict, then the previous values of $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{u}}_i$ remain as the candidate trajectories to be implemented.

Constraint (5.25a) serves as the initial condition of the prediction horizon by setting the candidate state vector of agent i at $k = 0$ equal to the most recent state measurement $\mathbf{x}_i$. Constraint (5.25b) requires that the optimal input and state trajectories $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$ computed along the prediction horizon satisfy the plant dynamics of agent i. Constraints (5.25c) and (5.25d) state that the optimal input and state vectors $\breve{\mathbf{x}}_i^k$ and $\breve{\mathbf{u}}_i^k$ of agent i must remain within the process constraint sets $\mathscr{X}_i$ and $\mathscr{U}_i$, respectively, at any time-step k along the prediction horizon. Finally, Constraint (5.25e) ensures that, by the end of the prediction horizon H, the computed optimal input trajectory $\overline{\mathbf{u}}_i^*$ leads the local state vector to a feasible candidate steady state $\mathbf{x}_i^s$.

To negotiate the candidate stationary state vector $\mathbf{x}_i^s$, agent i solves the following optimization problem during each iteration:

$$\min_{\overline{\mathbf{u}}_i, \mathbf{u}_i^s, \overline{\mathbf{x}}_i, \mathbf{x}_i^s} J_i(\mathbf{x}_i^s, \mathbf{u}_i^s, \hat{\mathbf{x}}_{-i|J}^s, \hat{\mathbf{u}}_{-i|J}^s), \tag{5.32}$$

subject to

$$\begin{aligned}
\mathbf{x}_i^0 &= \mathbf{x}_i, && (5.33a)\\
\mathbf{x}_i^{k+1} &= \mathbf{f}_i(\mathbf{x}_i^k, \mathbf{u}_i^k), \forall k \in \mathbb{I}_{0:H-1}, && (5.33b)\\
\mathbf{x}_i^s &= \mathbf{f}_i(\mathbf{x}_i^s, \mathbf{u}_i^s), && (5.33c)\\
\mathbf{x}_i^s &= \mathbf{x}_i^H, && (5.33d)\\
\mathbf{x}_i^k &\in \mathscr{X}_i, \forall k \in \mathbb{I}_{0:H}, && (5.33e)\\
\mathbf{u}_i^k &\in \mathscr{U}_i, \forall k \in \mathbb{I}_{0:H-1}, && (5.33f)\\
(\mathbf{x}_i^s, \mathbf{u}_i^s) &\in \mathscr{X}_i \times \mathscr{U}_i, && (5.33g)
\end{aligned}$$

where $\mathbf{x}_i^s$ and $\mathbf{u}_i^s$ denote the candidate stationary state and input vectors of agent i, and $\hat{\mathbf{x}}_{-i|J}^s$ and $\hat{\mathbf{u}}_{-i|J}^s$ contain similar information that agent i assumes for all neighbors $j \in \mathscr{N}_{-i|J}$. The steady state and input vectors that agent i assumes for an individual neighbor j are denoted by $\hat{\mathbf{x}}_j^s$ and $\hat{\mathbf{u}}_j^s$, respectively. Solving Problem (5.32) yields the optimal steady state and input vectors $\check{\mathbf{x}}_i^s$ and $\check{\mathbf{u}}_i^s$. Similar to the optimal trajectories $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$, the optimal stationary solution is only accepted if it yields conflict-free operation. Otherwise, the previous values of $\mathbf{x}_i^s$ and $\mathbf{u}_i^s$ remain as the candidate stationary terminal set-point.

Constraint (5.33c) ensures that the computed optimum $(\check{\mathbf{x}}_i^s, \check{\mathbf{u}}_i^s)$ is a stationary point. Constraint (5.33d) then requires that the computed state vector trajectory ends at the optimal steady state $\check{\mathbf{x}}_i^s$. Finally, Constraint (5.33g) states that the computed optimal stationary point $(\check{\mathbf{x}}_i^s, \check{\mathbf{u}}_i^s)$ must lie within the process constraint set $\mathscr{X}_i \times \mathscr{U}_i$.

The difference between Problems (5.24) and (5.32) is that the latter only considers the stage cost function $J_i(\cdot)$ and therefore computes a feasible optimal steady state vector $\check{\mathbf{x}}_i^s$ without minimizing $J_i(\cdot)$ over the prediction horizon. The reason that dynamics are considered in Problem (5.32) is to ensure that the computed optimal steady state vector $\check{\mathbf{x}}_i^s$ is reachable from the initial state vector $\mathbf{x}_i^0$. Problem (5.24) then computes optimal trajectories that minimize $J_i(\cdot)$ over the prediction horizon and steer the system to the reachable candidate steady state vector $\mathbf{x}_i^s$. A description of the proposed social hierarchy-based DEMPC algorithm now follows.

Algorithm 1. *Social hierarchy-based DEMPC coordination scheme. Implement in parallel for all agents $i \in \mathscr{I}$.*

Communication protocol:

- Send $\overline{\mathbf{x}}_i$, $\overline{\mathbf{u}}_i$, $\mathbf{x}_i^s$, and $\mathbf{u}_i^s$ to all agents $j \in \mathscr{N}_{+i|J}$, receive $\overline{\mathbf{x}}_j$, $\overline{\mathbf{u}}_j$, $\mathbf{x}_j^s$, and $\mathbf{u}_j^s$ from all agents $j \in \mathscr{N}_{-i|J}$, and set $\tilde{\mathbf{x}}_j = \overline{\mathbf{x}}_j$, $\tilde{\mathbf{u}}_j = \overline{\mathbf{u}}_j$, $\hat{\mathbf{x}}_j^s = \mathbf{x}_j^s$, and $\hat{\mathbf{u}}_j^s = \mathbf{u}_j^s$ for all $j \in \mathscr{N}_{-i|J}$.

Initialization:

1. Specify $N_q \geq 1$ and set $q_i = 1$.
2. Initialize $\mathbf{x}_i$, $\overline{\mathbf{x}}_i \in \overline{\mathscr{X}}_i = \mathscr{X}_i \times \cdots \times \mathscr{X}_i = \mathscr{X}_i^H$, $\overline{\mathbf{u}}_i \in \overline{\mathscr{U}}_i$, $\mathbf{x}_i^s \in \mathscr{X}_i$, $\mathbf{u}_i^s \in \mathscr{U}_i$ such that $\mathbf{x}_i^0 = \mathbf{x}_i$, $\mathbf{x}_i^{k+1} = \mathbf{f}_i(\mathbf{x}_i^k, \mathbf{u}_i^k) \forall k \in \mathbb{I}_{0:H-1}$, $\mathbf{x}_i^H = \mathbf{x}_i^s$, and $\mathbf{x}_i^s = \mathbf{f}_i(\mathbf{x}_i^s, \mathbf{u}_i^s)$.
3. Implement *communication protocol*.

Perform at each new time-step:

1. Measure $\mathbf{x}_i$, and compute $\overline{\mathbf{x}}_i$ such that $\mathbf{x}_i^0 = \mathbf{x}_i$ and $\mathbf{x}_i^{k+1} = \mathbf{f}_i(\mathbf{x}_i^k, \mathbf{u}_i^k) \forall k \in \mathbb{I}_{0:H-1}$.
2. Implement *communication protocol*.
3. For iteration number $p = 1, 2, \cdots, N_p$, do:

(a) For sequence slot number $q = 1, 2, \cdots, N_q$, do:

i. If $q = q_i$, (i) solve Problem (5.32) to acquire $\breve{\mathbf{x}}_i^s$ and $\breve{\mathbf{u}}_i^s$, (ii) compute $\hat{V}_i^s$ according to Eq. (5.16), (iv) send $\breve{\mathbf{x}}_i^s$ and $\breve{\mathbf{u}}_i^s$ to all agents $j \in \mathscr{N}_{+i|J}$, (v) receive $\breve{\mathbf{x}}_j^s$ and $\breve{\mathbf{u}}_j^s$ from all agents $j \in \mathscr{N}_{-i|J}$, and update $\hat{\mathbf{x}}_j^s = \breve{\mathbf{x}}_j^s$ and $\hat{\mathbf{x}}_j^s = \breve{\mathbf{x}}_j^s$.

ii. Else, receive $\breve{\mathbf{x}}_j^s$ and $\breve{\mathbf{u}}_j^s$ from all agents $j \in \mathscr{N}_{-i|J}$, and update $\hat{\mathbf{x}}_j^s = \breve{\mathbf{x}}_j^s$ and $\hat{\mathbf{x}}_j^s = \breve{\mathbf{x}}_j^s$.

(b) Compute $\breve{V}_i^s$ according to Eq. (5.17). If $\breve{V}_i^s > \hat{V}_i^s$, randomly change q_i with uniform probability and set $\breve{\mathbf{x}}_i^s = \mathbf{x}_i^s$ and $\breve{\mathbf{u}}_i^s = \mathbf{u}_i^s$. Else, if $\breve{V}_i^s \leq \hat{V}_i^s$, update $\mathbf{x}_i^s = \breve{\mathbf{x}}_i^s$ and $\mathbf{u}_i^s = \breve{\mathbf{u}}_i^s$.

(c) Implement *communication protocol*.

4. For iteration number $p = 1, 2, \cdots, N_p$, do:

(a) For sequence slot number $q = 1, 2, \cdots, N_q$, do:

i. If $q = q_i$, (i) solve Problem (5.24) to acquire $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$, (ii) compute $\hat{V}_i$ according to Eq. (5.20), (iv) send $\overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i^*$ to all agents $j \in \mathscr{N}_{+i|J}$, (v) receive $\overline{\mathbf{x}}_j^*$ and $\overline{\mathbf{u}}_j^*$ from all agents $j \in \mathscr{N}_{-i|J}$, and update $\tilde{\mathbf{x}}_j = \overline{\mathbf{x}}_j^*$ and $\tilde{\mathbf{u}}_j = \overline{\mathbf{u}}_j^*$.

ii. Else, receive $\overline{\mathbf{x}}_j^*$ and $\overline{\mathbf{u}}_j^*$ from all agents $j \in \mathscr{N}_{-i|J}$, and update $\tilde{\mathbf{x}}_j = \overline{\mathbf{x}}_j^*$ and $\tilde{\mathbf{u}}_j = \overline{\mathbf{u}}_j^*$.

(b) Compute $\breve{V}_i$ according to Eq. (5.20). If $\breve{V}_i > \hat{V}_i$, randomly change q_i with uniform probability and set $\overline{\mathbf{x}}_i^* = \overline{\mathbf{x}}_i$ and $\overline{\mathbf{u}}_i^* = \overline{\mathbf{u}}_i$. Else, if $\breve{V}_i \leq \hat{V}_i$, update $\overline{\mathbf{x}}_i = \overline{\mathbf{x}}_i^*$ and $\overline{\mathbf{u}}_i = \overline{\mathbf{u}}_i^*$.

(c) Implement *communication protocol*.

5. Apply candidate control input $\mathbf{u}_i^0$ to the system, update $\overline{\mathbf{u}}_i$ such that $\mathbf{u}_i^k = \mathbf{u}_i^{k+1} \forall k \in \mathbb{I}_{0:H-2}$ and $\mathbf{u}_i^{H-1} = \mathbf{u}_i^s$.

During initialization, step 1 requires the control system designer to specify the quantity N_q of hierarchy levels and to allocate agent i to the first level. As a result, all agents initially solve their local optimization problems in parallel. Step 2 requires that feasible state and input trajectories $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{u}}_i$ that satisfy the constraints of Eq. (5.25), and steer the system to a reachable stationary point $(\mathbf{x}_i^s, \mathbf{u}_i^s)$, be specified for agent i given the initial state vector $\mathbf{x}_i$. Finally, step 3 involves the exchange of candidate trajectories $\overline{\mathbf{x}}_j$ and $\overline{\mathbf{u}}_j$ and stationary vectors $\mathbf{x}_j^s$ and $\mathbf{u}_j^s$ between agent i and all of its neighbors $j \in \mathscr{N}_{-i|J}$. Agent i then uses this incoming information to establish the assumed values of its neighbors' states and inputs $\tilde{\mathbf{x}}_j$, $\tilde{\mathbf{u}}_j$, $\hat{\mathbf{x}}_j^s$, and $\hat{\mathbf{u}}_j^s$ for all $j \in \mathscr{N}_{-i|J}$. These assumptions remain unchanged until future communication.

Focusing on the recursive portion of the DEMPC algorithm, agent i first measures its state vector $\mathbf{x}_i$ and updates its local state trajectory $\overline{\mathbf{x}}_i$ using the most up-to-date candidate input sequence $\overline{\mathbf{u}}_i$. In step 2, this updated information is communicated between neighbors. As a result, prior to step 3, agent i possesses the most up-to-date trajectories and stationary vectors $\overline{\mathbf{x}}_j$,$\overline{\mathbf{u}}_j$, $\mathbf{x}_j^s$, and $\mathbf{u}_j^s$ of neighbors $j \in \mathscr{N}_{-i|J}$. The assumptions $\tilde{\mathbf{x}}_j$, $\tilde{\mathbf{u}}_j$, $\hat{\mathbf{x}}_j^s$, and $\hat{\mathbf{u}}_j^s$ for all $j \in \mathscr{N}_{-i|J}$ are also updated as a result.

Step 3 initiates an iterative process within the current sampling time-step with the objective of identifying an appropriate candidate stationary point $(\mathbf{x}_i^s, \mathbf{u}_i^s)$ to serve as a terminal constraint in future steps. In step 3(a), agent i cycles through all hierarchy levels sequentially and, during its allocated hierarchy level, solves Problem (5.32) to obtain an optimal stationary point $(\breve{\mathbf{x}}_i^s, \breve{\mathbf{u}}_i^s)$. These vectors, along with the assumptions $\hat{\mathbf{x}}_j^s$ and $\hat{\mathbf{u}}_j^s$ for all $j \in \mathscr{N}_{-i|J}$, are then used to calculate the naive stage cost function value $\hat{V}_i^s$ to be used for comparison to $\breve{V}_i^s$ later on. Finally, agent i exchanges optimal stationary vector information with neighboring agents and updates its assumptions $\hat{\mathbf{x}}_j^s$ and $\hat{\mathbf{u}}_j^s$ for all $j \in \mathscr{N}_{-i|J}$. Outside of its hierarchy level, agent i remains idle and only receives information from neighbors and updates its assumptions.

After all hierarchy levels have been cycled through, agent i will have received $\breve{\mathbf{x}}_j^s$ and $\breve{\mathbf{u}}_j^s$ from all neighbors $j \in \mathscr{N}_{-i|J}$. Step 3(b) requires the computation of the informed stage cost function value $\breve{V}_i^s$. If $\breve{V}_i^s > \hat{V}_i^s$, then agent i is in conflict with its neighbors while attempting to identify an optimal stationary point, and it is necessary for its hierarchy level q_i to be randomly mutated. Further, if conflict is encountered, then agent i resets it optimal stationary vectors $\breve{\mathbf{x}}_i^s$ and $\breve{\mathbf{u}}_i^s$ to the candidate stationary vectors $\mathbf{x}_i^s$ and $\mathbf{u}_i^s$. This reset essentially erases $\breve{\mathbf{x}}_i^s$ and $\breve{\mathbf{u}}_i^s$ so that conflict-yielding optimal solutions are no longer communicated. If $\breve{V}_i^s \leq \hat{V}_i^s$, then agent i is operating free of conflict, and its current hierarchy level may be maintained. Further, since the recently computed stationary vectors $\breve{\mathbf{x}}_i^s$ and $\breve{\mathbf{u}}_i^s$ did not yield conflict, they may replace the candidate stationary vectors $\mathbf{x}_i^s$ and $\mathbf{u}_i^s$. As a result, agent i updates $\mathbf{x}_i^s$ and $\mathbf{u}_i^s$ using $\breve{\mathbf{x}}_i^s$ and $\breve{\mathbf{u}}_i^s$.

Step 3 is terminated once the maximum number of iterations N_p has been reached. Step 4 essentially repeats step 3, except the candidate terminal state vector $\mathbf{x}_i^s$ has now been updated, and the objective is to update candidate state and input trajectories $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{u}}_i$. The maximum number of iterations available for this step is also N_p. Step 5 requires that the control input vector corresponding to the first time-step along the prediction horizon be applied to the system. Additionally, the candidate input trajectory $\overline{\mathbf{u}}_i$ for the next sampling time-step is constructed by concatenating it with the candidate stationary input vector $\mathbf{u}_i^s$.

5.3.3 Closed-loop properties

The current subsection establishes closed-loop properties for Algorithm 1. We first demonstrate that, regardless of the interaction topology of a multi-agent system, all conflicts may be resolved in a finite number of iterations with some probability. We then address convergence, feasibility, and stability.

Theorem 1. *(Conflict resolution) There exist a finite number of iterations after which, with some probability greater than zero, Inequalities (5.22) and (5.23) are guaranteed to be satisfied at each iteration within a single time-step.*

Proof. This proof consists of two parts; (i) it is first necessary to prove that, for any interconnected system, at least one social hierarchy exists that will ensure system-wide conflict resolution; (ii) it is then proved that the probability of agents self-organizing according to such a social hierarchy is greater than zero during any iteration.

Part I: Consider a multi-agent system wherein all agents possess the same stage cost function $J(\cdot)$ defined as follows:

$$J(\mathbf{x},\mathbf{u}) := \sum_{i\in\mathscr{I}} \ell_i(\mathbf{x}_i,\mathbf{u}_i,\mathbf{x}_{-i},\mathbf{u}_{-i}), \tag{5.34}$$

where $\mathbf{x} \in \mathbb{R}^{\sum_{i\in\mathscr{I}} n_i}$ and $\mathbf{u} \in \mathbb{R}^{\sum_{i\in\mathscr{I}} m_i}$ contain the n_i states and m_i inputs of all agents $i \in \mathscr{I}$, and $J(\cdot)$ is global-cooperative in that it considers the local interests $\ell_i(\cdot)$ of all agents $i \in \mathscr{I}$. The resulting dynamic optimization problem over the prediction horizon H for any agent i is therefore defined as follows:

$$\min_{\overline{\mathbf{u}}_i,\overline{\mathbf{x}}_i} \sum_{k=0}^{H-1} J(\mathbf{x}^k,\mathbf{u}^k), \tag{5.35}$$

with constraints similar to those of Problem (5.24). The vectors $\mathbf{x}^k$ and $\mathbf{u}^k$ denote system-wide states and inputs at time-step k along the prediction horizon. The stationary optimization problem for any agent i is defined as follows:

$$\min_{\overline{\mathbf{u}}_i,\mathbf{u}_i^s,\overline{\mathbf{x}}_i,\mathbf{x}_i^s} J(\mathbf{x}^s,\mathbf{u}^s), \tag{5.36}$$

with constraints similar to those of Problem (5.32). The vectors $\mathbf{x}^s$ and $\mathbf{u}^s$ denote system-wide steady states and inputs. We also define $V(\cdot)$ and $V^s(\cdot)$ as follows:

$$\begin{aligned} V(\bar{\mathbf{x}},\bar{\mathbf{u}}) &:= \sum_{k=0}^{H-1} J(\mathbf{x}^k,\mathbf{u}^k), && (5.37)\\ V^s(\mathbf{x}^s,\mathbf{u}^s) &:= J(\mathbf{x}^s,\mathbf{u}^s), && (5.38) \end{aligned}$$

where $\bar{\mathbf{x}} = \left(\mathbf{x}^0,\mathbf{x}^1,\cdots,\mathbf{x}^H\right)$ and $\bar{\mathbf{u}} = \left(\mathbf{u}^0,\mathbf{u}^1,\cdots,\mathbf{x}^{H-1}\right)$ denote system-wide state and input trajectories over the prediction horizon.

As a reference case, let the agents operate in a fully sequential manner such that, at any given iteration, only one agent solves its local optimization problem. Focusing on the negotiation of a stationary point for now, only agent i updates $\mathbf{x}_i^s$ and $\mathbf{u}_i^s$ at some iteration p, then transmits this information to all other agents. Furthermore, let recursive feasibility, which is established independently in Theorem 3, be presumed, and let Assumptions 2, 4, and 5 concerning continuity and bounded minima hold. Under these conditions, upon solving Problem (5.36), each agent $i \in \mathscr{I}$ is guaranteed to shift $\mathbf{x}^s$ and $\mathbf{u}^s$ until the gradient of $J(\cdot)$ projected along the variable space $(\mathbf{x}_i^s,\mathbf{u}_i^s)$ satisfies optimality conditions. Since this process occurs sequentially across all agents $i \in \mathscr{I}$, $V^s(\cdot)$ is guaranteed to decrease or remain unchanged after each subsequent agent's update to $\mathbf{x}_i^s$ and $\mathbf{u}_i^s$, which ensures conflict-free operation as per Definition 2.

We now prove that the above result may be achieved using social hierarchies that are not necessarily fully sequential (*i.e.* a subset of agents may solve their local optimization problems simultaneously) and also assuming neighborhood-cooperative stage cost functions as per Assumption 3. From the perspective of any agent $i \in \mathscr{I}$, $J(\cdot)$ may be arranged as follows:

$$\begin{aligned} J(\mathbf{x},\mathbf{u}) = \ell_i(\mathbf{x}_i,\mathbf{u}_i,\mathbf{x}_{-i},\mathbf{u}_{-i}) &+ \sum_{j\in\mathscr{N}_{+i}} \ell_j(\mathbf{x}_j,\mathbf{u}_j,\mathbf{x}_{-j},\mathbf{u}_{-j}) \\ &+ \sum_{\kappa\in\mathscr{I}\setminus\mathscr{N}_{+i},\kappa\neq i} \ell_\kappa(\mathbf{x}_\kappa,\mathbf{u}_\kappa,\mathbf{x}_{-\kappa},\mathbf{u}_{-\kappa}), \quad (5.39) \end{aligned}$$

where the three terms on the right-hand-side represent, from left to right, the local interests of agent i, the local interests of agents whose cost functions are influenced by agent i, and the local interests of all remaining agents whose cost functions are uninfluenced by agent i. Taking the gradient of $J(\cdot)$ along the variable space $(\mathbf{x}_i, \mathbf{u}_i)$ yields the following:

$$\nabla_{\mathbf{x}_i,\mathbf{u}_i} J(\mathbf{x},\mathbf{u}) = \nabla_{\mathbf{x}_i,\mathbf{u}_i} \ell_i(\mathbf{x}_i,\mathbf{u}_i,\mathbf{x}_{-i},\mathbf{u}_{-i}) + \nabla_{\mathbf{x}_i,\mathbf{u}_i} \sum_{j\in\mathscr{N}_{+i}} \ell_j(\mathbf{x}_j,\mathbf{u}_j,\mathbf{x}_{-j},\mathbf{u}_{-j}). \quad (5.40)$$

Note that gradient of $\ell_\kappa(\cdot)$ for all $\kappa \in \mathscr{I} \setminus \mathscr{N}_{+i}, \kappa \neq i$ is zero along $(\mathbf{x}_i, \mathbf{u}_i)$ since these expressions have no dependency upon the operation of agent i. The shape of $J(\cdot)$ along the variable space $(\mathbf{x}_i, \mathbf{u}_i)$ is therefore only influenced by $\mathbf{x}_j$ and $\mathbf{u}_j$ for all $j \in \mathscr{N}_{-i|J}$, since $\mathscr{N}_{-i|J} = \mathscr{N}_{-i} \cup \mathscr{N}_{+i} \cup \mathscr{N}_{-j} \forall j \in \mathscr{N}_{+i}$. This result yields two important consequences.

First, the local optimization problem of agent i is uninfluenced by the control actions of agents $\kappa \in \mathscr{I} \setminus \mathscr{N}_{-i|J}, \kappa \neq i$. Agent i may therefore update its control actions in parallel with agents $\kappa \in \mathscr{I} \setminus \mathscr{N}_{-i|J}, \kappa \neq i$, and the computed optimal trajectories are guaranteed to decrease or preserve $V^s(\cdot)$. Second, whether agent i employs the global stage cost function $J(\cdot)$, or the neighborhood-cooperative stage cost function $J_i(\cdot)$ defined in Eq. (5.11), the computed optimal trajectories remain unchanged. This property is true since $\nabla_{\mathbf{x}_i,\mathbf{u}_i} J(\cdot) = \nabla_{\mathbf{x}_i,\mathbf{u}_i} J_i(\cdot)$. Therefore, if Assumption 3 concerning neighborhood-cooperative cost functions holds, and if no two agents i and j such that $j \in \mathscr{N}_{-i|J}$ for all $i, j \in \mathscr{I}, j \neq i$ solve their local optimization problems simultaneously, then at least one social hierarchy exists that will satisfy Inequality (5.22) after each iteration. The above logic may be extended to step 4 in Algorithm 1 without modification. Part I of the proof is thus completed.

Part II: Due to its distributed nature, Algorithm 1 may lead some agents to resolve their conflicts earlier than others. However, we prove that, even in a worst-case scenario in which all agents initially encounter conflict, the probability that all conflicts will be resolved within a single iteration is greater than zero. Let N_s describe the quantity of possible social hierarchies that will resolve conflict among N agents. If agents must, with uniform probability, randomly choose among N_q hierarchy levels, the probability P that all conflicts are resolved within a single iteration is defined as follows:

$$P = N_s \left(\frac{1}{N_q}\right)^N. \quad (5.41)$$

Thus, as long as $N_s > 0$, $P > 0$ must also be true. Part I of this proof demonstrated that $N_s \geq 1$ for any interconnected system with neighborhood-cooperative cost functions. □

Theorem 2. *(Convergence) Let* $\mathbf{x}_i^{s,p}$ *and* $\overline{\mathbf{u}}_i^p$ *denote the candidate steady state vector and input trajectory* $\mathbf{x}_i^s$ *and* $\overline{\mathbf{u}}_i$ *held by agent i at iteration p. Given a sufficiently large number of iterations in Steps 3 and 4 in Algorithm 1, the following inequalities are guaranteed to be satisfied for all* $i \in \mathscr{I}$*:*

$$\begin{aligned} \left\| \mathbf{x}_i^{s,p} - \mathbf{x}_i^{s,p-1} \right\| &\leq \alpha \left\| \mathbf{x}_i^{s,p-1} \right\|, && (5.42) \\ \left\| \overline{\mathbf{u}}_i^{p} - \overline{\mathbf{u}}_i^{p-1} \right\| &\leq \beta \left\| \overline{\mathbf{u}}_i^{p-1} \right\|, && (5.43) \end{aligned}$$

where $\alpha > 0$ *and* $\beta > 0$ *represent fixed convergence tolerances.*

Proof. Theorem 1 guarantees that, with a large enough number of iterations, system-wide conflict-resolution may be achieved with some probability greater than zero. As per Definition 2, and focusing on step 3 from Algorithm 1, once system-wide conflict-free operation has been achieved, the inequality $\check{V}_i^s \leq \hat{V}_i^s$ is guaranteed to be satisfied after each iteration for any agent $i \in \mathscr{I}$. Let $\hat{V}_i^{s,p}$ and $\check{V}_i^{s,p}$ denote $\hat{V}_i^s$ and $\check{V}_i^s$ computed during iteration number p. Since $\hat{V}_i^p$ is computed using optimal state and input trajectories of agent i that have been updated during iteration p, then $\hat{V}_i^{s,p} \leq \check{V}_i^{s,p-1}$ must be true during conflict-free operation. Therefore, since $\check{V}_i^{s,p} \leq \hat{V}_i^{s,p}$ and $\hat{V}_i^{s,p} \leq \check{V}_i^{s,p-1}$, then $\check{V}_i^{s,p} \leq \check{V}_i^{s,p-1}$ must be true after conflicts have been resolved, which indicates that the value of $J_i(\cdot)$ computed using updated stationary points of all relevant agents is guaranteed to decrease with each subsequent iteration. If Assumption 4 concerning bounded minima holds, then $(\mathbf{x}_i^s, \mathbf{u}_i^s)$ is guaranteed to approach a local minimizer of $J_i(\cdot)$, thus satisfying Inequality (5.42). The above logic may be extended to step 4 of Algorithm 1 without modification; thus concluding the proof. □

Theorem 3. *(Recursive feasibility) The constraints of Problems (5.32) and (5.24) are guaranteed to be satisfied during each iteration at any sampling time* $k \geq 0$ *for all agents* $i \in \mathscr{I}$*.*

Proof. Provided that initial values of $\overline{\mathbf{x}}_i$, $\overline{\mathbf{u}}_i$, $\mathbf{x}_i^s$, and $\mathbf{u}_i^s$ established at the start of some time-step are feasible, then at least one viable set of values for $\overline{\mathbf{x}}_i$, $\overline{\mathbf{u}}_i$, $\mathbf{x}_i^s$, and $\mathbf{u}_i^s$ exists that satisfies Constraints (5.25) and (5.33), thus ensuring recursive feasibility at each subsequent iteration within the time-step. Further, since the absence of disturbances restricts state progression to $\overline{\mathbf{x}}_i$, then feasibility is preserved at subsequent time-steps. Guaranteeing recursive feasibility thus requires that $\overline{\mathbf{x}}_i$, $\overline{\mathbf{u}}_i$, $\mathbf{x}_i^s$, and $\mathbf{u}_i^s$ are initially feasible. Referring to Eq. (5.7), the existence of $\overline{\mathbf{x}}_i$, $\overline{\mathbf{u}}_i$, $\mathbf{x}_i^s$, and $\mathbf{u}_i^s$ such that Constraints (5.25) and (5.33) are satisfied requires that the set $\mathscr{X}_i^{0\to s}$ is not empty. Assumption 1 concerning weak controllability bounds the control input trajectory $\overline{\mathbf{u}}_i$ required to steer any initial state vector $\mathbf{x}_i^0 \in \mathscr{X}_i^{0\to s}$ to a reachable stationary point $(\mathbf{x}_i^s, \mathbf{u}_i^s)$ such that $(\mathbf{x}_i^0, \overline{\mathbf{u}}_i, \mathbf{x}_i^s) \in \mathscr{Z}_i^{0\to s}$. As a result, the set $\mathscr{X}_i^{0\to s}$ must be non-empty; thus concluding the proof. □

Theorem 4. *(Closed-loop stability) For all agents $i \in \mathscr{I}$, as $k \to \infty$, the measured state vector $\mathbf{x}_i$ will remain bounded within a set $\mathscr{X}_i^* \subset \mathscr{X}_i^{0\to s}$ surrounding a fixed stationary point $\mathbf{x}_i^* \in \mathscr{X}_i^s$. The set $\mathscr{X}_i^*$ is defined as follows:*

$$\mathscr{X}_i^* := \left\{ \mathbf{x}_i \in \mathscr{X}_i \mid \exists \overline{\mathbf{u}}_i \in \overline{\mathscr{U}}_i : (\mathbf{x}_i, \overline{\mathbf{u}}_i, \mathbf{x}_i^*) \in \mathscr{Z}_i^{0\to s} \right\}. \tag{5.44}$$

Proof. For all agents $i \in \mathscr{I}$, while updating candidate state and input vector trajectories $\overline{\mathbf{x}}_i$ and $\overline{\mathbf{u}}_i$, Constraint (5.25e) ensures that the state vector $\mathbf{x}_i$ always remains within a reachable set surrounding some steady state vector $\mathbf{x}_i^s$. Therefore, in order to prove that this reachable set ultimately maintains a fixed value $\mathscr{X}_i^*$, one must prove that the terminal steady state vector $\mathbf{x}_i^s$ approaches a fixed value $\mathbf{x}_i^*$ as $k \to \infty$ for all agents $i \in \mathscr{I}$.

We assume for the time being that, after a sufficient number of iterations, agents establish a social hierarchy that permanently resolves all conflicts. That is, Inequality (5.22) is guaranteed to be satisfied at each iteration within all subsequent time-steps, and the social hierarchy therefore ceases to change. We shall refer to such a social hierarchy as a *universal social hierarchy*. Let $\breve{V}_i^{s,k}|_{\mathrm{A}}$ and $\breve{V}_i^{s,k}|_{\mathrm{B}}$ denote $\breve{V}_i^s$ computed at the initial and final iterations, respectively, of time-step k. It is clear that, after a universal social hierarchy has been established, $\breve{V}_i^{s,k}|_{\mathrm{B}} \leq \breve{V}_i^{s,k}|_{\mathrm{A}}$ is satisfied at all subsequent time-steps. Since $\breve{V}_i^{s,k}|_{\mathrm{A}}$ is computed using $\breve{\mathbf{x}}_i^s$ and $\breve{\mathbf{u}}_i^s$ obtained by solving Problem (5.32) at the first iteration of time-step k, then $\breve{V}_i^{s,k}|_{\mathrm{A}} \leq \breve{V}_i^{s,k-1}|_{\mathrm{B}}$

must also be true. As a result, $\breve{V}_i^{s,k}|_{\mathrm{B}} \leq \breve{V}_i^{s,k-1}|_{\mathrm{B}}$ must be satisfied at all time-steps following the establishment of a universal social hierarchy. If Assumption 4 concerning bounded minima holds, $(\mathbf{x}_i^s, \mathbf{u}_i^s)$ is guaranteed to approach some fixed point $(\mathbf{x}_i^*, \mathbf{u}_i^*)$ for all $i \in \mathscr{I}$ as $k \to \infty$, where $(\mathbf{x}_i^*, \mathbf{u}_i^*)$ is some local minimizer of $J_i(\cdot)$.

The remaining task is to prove that a universal social hierarchy is in fact attainable after a sufficient number of iterations. The existence of at least one such social hierarchy has already been established in the proof for Theorem 1. Namely, if Assumption 3 concerning neighborhood-cooperative cost functions holds, and if no two agents i and j such that $j \in \mathscr{N}_{-i|J}$ for all $i, j \in \mathscr{I}, j \neq i$ make decisions simultaneously, then Inequality (5.22) is guaranteed to be satisfied at every iteration within any sampling time. If conflict persists, then agents will, after a sufficient number of iterations, self-organize according to a universal social hierarchy with some probability greater than zero; thus concluding the proof. □

5.3.4 Determining social hierarchy properties

One final issue that must be addressed concerns the selection of N_q. In order to guarantee that a universal social hierarchy is attainable, N_q must be large enough such that a social hierarchy wherein no neighboring pairs of agents operate in parallel is permissible. This goal invokes the vertex coloring problem from graph theory [35]. In brief, vertex coloring of a graph requires assigning colors to all nodes such that no two interconnected nodes share the same color. Returning to the context of the current chapter, each *node* signifies an agent, each *color* represents a specific level along a social hierarchy, and *interconnection* symbolizes cost function coupling. In graph theory, the *chromatic number* refers to the minimum number of colors required to complete the vertex coloring problem. Therefore, in the current context, N_q should be equal to or greater than the chromatic number of the system graph.

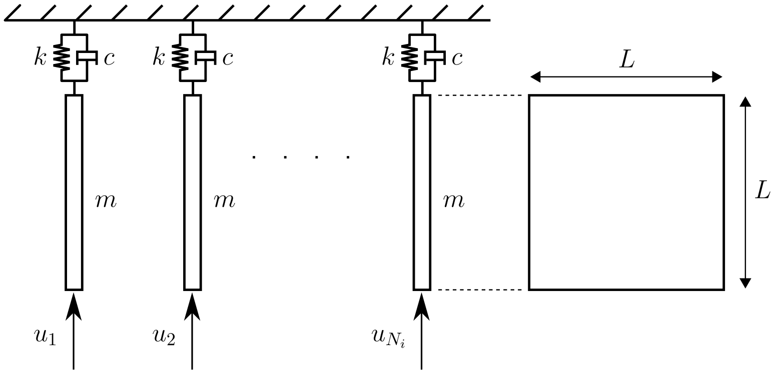


Figure 5.3: Schematic of the plate overlap problem used in the numerical example.

5.4 Numerical example

5.4.1 Problem description

Consider the mechanical system described in Figure 5.3. This setup consists of N square plates that are supported by spring-damper systems and perfectly aligned at equilibrium. Each square plate has a mass $m = 1.0\,\mathrm{kg}$, side length $L = 0.25\,\mathrm{m}$, and is supported by stiffness and damping coefficients $k = 1.0\,\mathrm{N/m}$ and $c = 1.0\,\mathrm{kg/s}$. Further, the vertical position of each plate i is controlled via an input force u_i that is regulated by agent i. The resulting continuous-time dynamics of each plate are expressed in state-space form as follows:

$$\begin{bmatrix} \dot{x}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_i \\ v_i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{u_i}{m} \end{bmatrix}, \tag{5.45}$$

where x_i and v_i denote the vertical position and velocity of plate i. For the remainder of the current section, we work with the discrete-time form of Eq. (5.45).

The economic control objective of each agent i is to, without excessive actuation, minimize the overlap area between its respective plate and those of neighboring agents $i-1$ and $i+1$. The resulting stage cost function of agent i is therefore

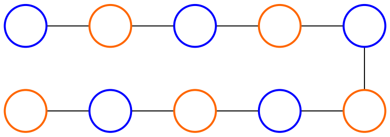


Figure 5.4: Interaction graph of the plate overlap problem with vertex coloring used to identify a universal social hierarchy.

expressed as follows:

$$J_i(x_i, u_i, x_{-i}) = \frac{1}{|\mathscr{N}_{-i}|} \sum_{j \in \mathscr{N}_{-i}} A_{i,j}(x_i, x_j) + u_i^{\mathrm{T}} u_i, \tag{5.46}$$

where the set $\mathscr{N}_{-i}$ contains the indices of all neighboring plates j that are physically adjacent to plate i, $|\mathscr{N}_{-i}|$ is the cardinality of the set $\mathscr{N}_{-i}$, and $A_{i,j}(\cdot)$ defines the overlap area between agents i and j as follows:

$$A_{i,j}(x_i, x_j) = \begin{cases} 0 & , \quad \left|x_i - x_j\right| \geq L, \\ L\left(L - \left|x_i - x_j\right|\right) & , \quad \left|x_i - x_j\right| < L. \end{cases} \tag{5.47}$$

Note that, although simple in terms of system dynamics, the numerical problem described above entails nonlinear and non-convex cost functions. Physically, non-convexity stems from the property that any two adjacent plates may be relocated in multiple ways to minimize their respective overlap areas.

5.4.2 Social hierarchy-based DEMPC properties

The interaction graph for an example problem with ten plates is shown in Figure 5.4. The solution to the vertex coloring problem for this example involves simply alternating the color of each subsequent node, which yields a chromatic number of two regardless of the quantity of vertices. An appropriate choice for the number of hierarchy levels is thus $N_q = 2$.

We implement prediction and control horizons of $H = 5$ time-steps and a sampling period of $1.0\,\mathrm{sec}$. Within a single sampling period, $N_p = 5$ iterations are permitted for the negotiation of optimal stationary points (*i.e.* step 3 in Algorithm 1), followed by five more iterations for trajectory optimization (*i.e.* step 4 in Algorithm 1). Finally, the actuation force of any agent is bounded as $-0.25\,\mathrm{N} \leq u_i \leq$

0.25 N. These values were selected to limit the steady-state displacement of each plate to a maximum of $L = 0.25\,\mathrm{m}$.

Finally, all plates are initially at rest with zero displacement from equilibrium (*i.e.* $\mathbf{x}_i = \mathbf{0}$ and $\mathbf{u}_i = \mathbf{0}$) and are therefore perfectly aligned with their neighbors. The state and input vector trajectories of all agents are initialized as $\overline{\mathbf{x}}_i = (\mathbf{0}, \cdots, \mathbf{0})$ and $\overline{\mathbf{u}}_i = (\mathbf{0}, \cdots, \mathbf{0})$.

5.4.3 Simulation results

Social hierarchy-based DEMPC – ten plates

The first results we present pertain to five simulations involving ten plates. The outcomes in these simulations will differ due to the element of randomness in the proposed coordination algorithm. In Figure 5.5, we show the variation in social hierarchy levels over the iteration number for all five simulations. Rather than displaying the actual hierarchy level of any particular agent, which would yield a cluttered image, we instead present the cumulative number of social hierarchy level changes. That is, let some cumulative counter start at zero for each simulation. Then, each time an agent alters its social hierarchy level, the cumulative count increases by one.

What is observed in Figure 5.5 is that, in all five simulations, the cumulative count of social hierarchy variations ultimately reaches a fixed value. This outcome indicates that, with a sufficient number of iterations, the agents sort themselves along a social hierarchy that guarantees conflict-free operation according to Definition 2 in all future iterations, thus validating Theorem 1 concerning conflict resolution.

Next, we plot the evolution of the global cost function V with respect to the iteration number in Figure 5.6 for all five simulations, with V computed as follows:

$$V = \sum_{i \in \mathscr{I}} \breve{V}_i. \tag{5.48}$$

Note that the evolution of V differs in each case due to the element of stochasticity inherent to the proposed algorithm. However, in all five simulations, a reduction in V to some locally-optimal value is evident after conflicts have been resolved.

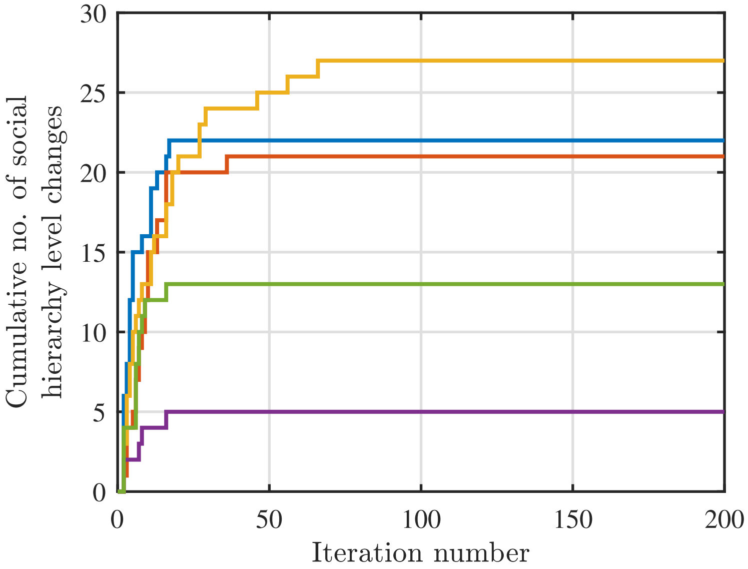


Figure 5.5: Evolution of the cumulative number of social hierarchy changes for five simulations of ten plates. Each color corresponds to a different simulation.

Further, within each time-step (*i.e.* within each 20 iteration interval), the reduction or preservation of V is apparent after agents have settled on an appropriate social hierarchy. This latter outcome validates Theorem 2 on convergence.

As a reference in Figure 5.6, we plot (using a black dotted line) the globally-optimal evolution of V, which is obtained by initializing the agents' hierarchy levels according to the universal social hierarchy from Figure 5.4. In this case, the agents do not alter their social hierarchy levels as conflict-free operation is guaranteed from the beginning of the simulation. As a result, V is reduced or preserved within each time-step immediately from the start of the simulation. This result validates the existence of a universal social hierarchy.

In Figure 5.7, we plot the evolution of stationary target positions over the iteration number for five simulations. Rather than displaying the stationary targets of individual agents, which would yield a cluttered image, we plot the mean stationary

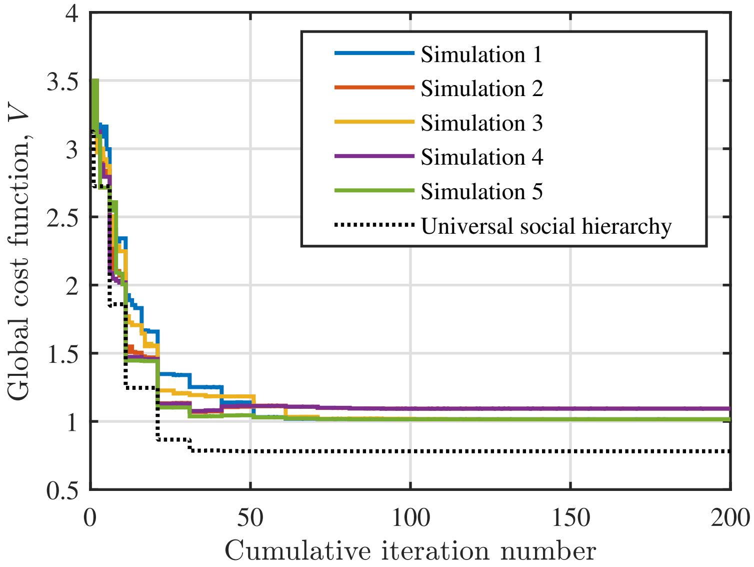


Figure 5.6: Evolution of the global cost function for five simulations of ten plates.

target $\overline{\mathbf{x}}_s$ of the entire plant, which is computed as follows:

$$\overline{\mathbf{x}}_s = \frac{1}{N} \sum_{i \in \mathscr{I}} \breve{x}_i^s. \tag{5.49}$$

It is evident from Figure 5.7 that values of $\breve{x}_i^s$ for all agents $i \in \mathscr{I}$ ultimately converge to fixed values in all simulations. This outcome validates Theorem 4 concerning bounded closed-loop stability, which requires fixed stationary targets.

Finally, we have plotted the locations of all plates at the final sampling time in each simulation in Figure 5.8. The color of each plate indicates its social hierarchy level (*i.e.* blue denotes $q_i = 1$, red denotes $q_i = 2$). The globally-optimal configuration would involve all adjacent plates being relocated in opposite directions; however, the social hierarchy-based DEMPC algorithm is only capable of finding a locally-optimal layout wherein some plates (*e.g.* plate 8 in simulation 1) must remain at the origin to minimize overlap with their neighbors.

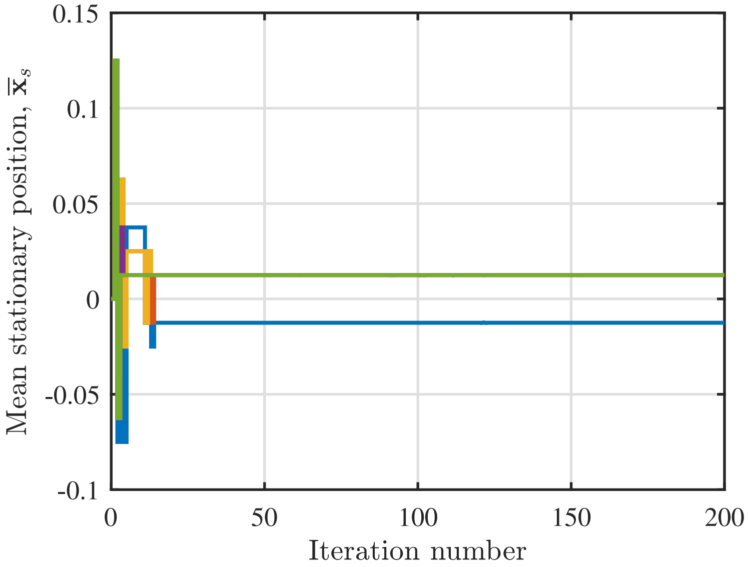


Figure 5.7: Evolution of the mean system-wide stationary target position for five different simulations involving ten plates.

Parallel vs. social hierarchy-based DEMPC

We compare the performance of the proposed social hierarchy-based DEMPC to a basic parallel DEMPC algorithm wherein all agents solve their local EMPC problems simultaneously and exchange stationary vectors and trajectories with their neighbors. In essence, a parallel DEMPC algorithm is identical to Algorithm 1, except with $N_q = 1$.

We first plot the evolution of V over the iteration number using a parallel DEMPC algorithm in Figure 5.9 for five simulations, each consisting of a different number of plates. The parallel DEMPC algorithm is in fact able to naturally resolve conflicts and ultimately decrease V to some locally minimum value. However, as the quantity of agents N increases, a greater number of iterations is required to decrease V to a locally-minimum value. For instance, with $N = 10$, V reaches a local minimum in 20 iterations. If $N = 80$ however, over 100 iterations are required.

In Figure 5.10, we plot the same information as Figure 5.9, except using the proposed social hierarchy-based DEMPC algorithm. The improvement is clear. The value of N has no discernible effect on the number of iterations necessary

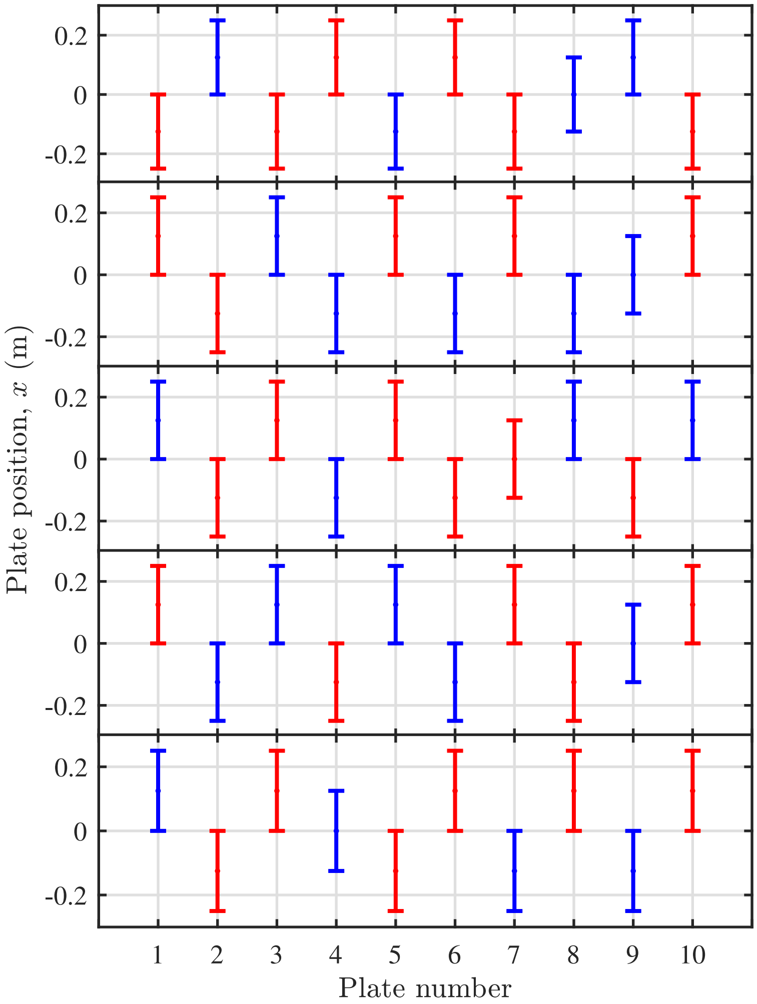


Figure 5.8: Plate locations at the final sampling time in (from top to bottom) simulations 1 to 5.

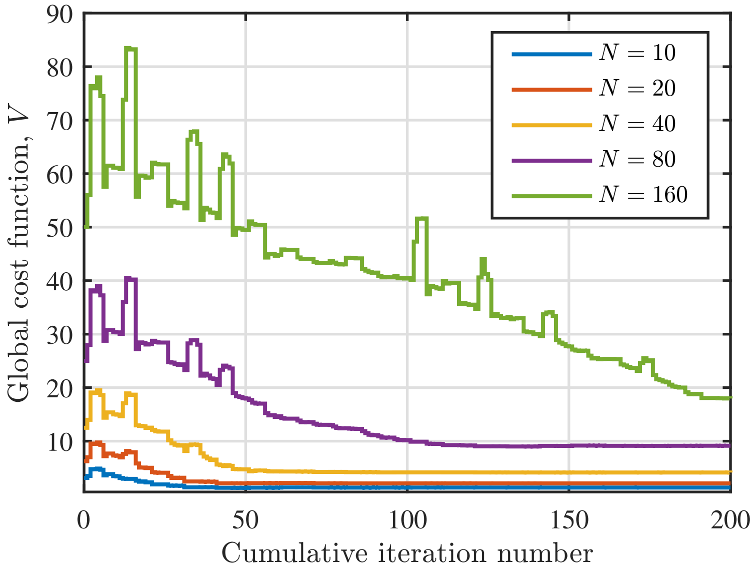


Figure 5.9: Evolution of the global cost function V using a parallel DEMPC algorithm for five simulations involving different quantities of plates.

for reducing V to a locally-minimum value. In each simulation, approximately 30 iterations are required for V to settle at some minimum value. This outcome results from the fact that the likelihood of any agent resolving conflict locally is dependent solely on its neighborhood interaction topology. Thus, raising N should not impact the number of iterations required for system-wide conflict resolution. This outcome is further validation of the scalability of the social hierarchy-based method.

5.5 Conclusions and recommendations for future research

We have presented a novel concept for addressing non-convexity in cost functions of distributed economic model predictive control systems with unknown terminal stationary targets. This concept involves agents self-organizing into a finite hierarchy using evolutionary principles, and ultimately enables agents to make decisions that are mutually beneficial with those of their neighbors. Theorems guarantee-

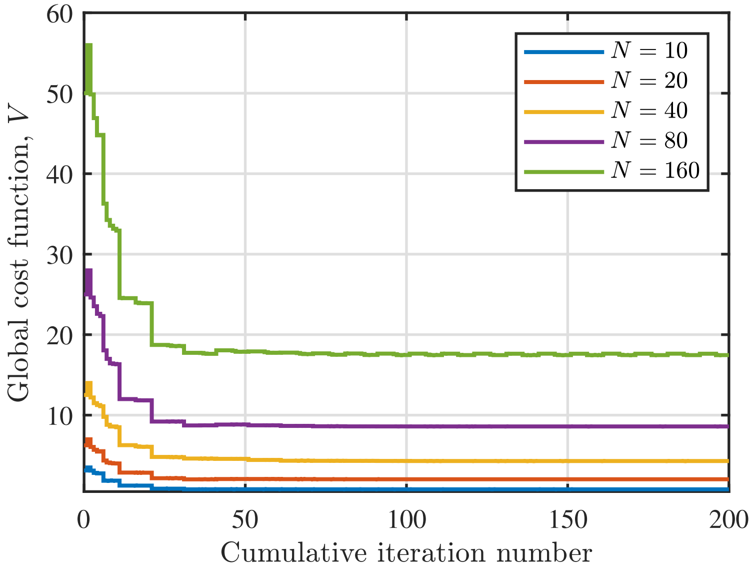


Figure 5.10: Evolution of the global cost function V using the proposed social hierarchy-based DEMPC algorithm for five simulations involving different quantities of plates.

ing convergence, recursive feasibility, and bounded closed-loop stability have also been provided for the proposed social hierarchy-based algorithm.

These theorems were validated using a numerical example involving a series of suspended square plates wherein each agent attempted to minimize the overlap area between its respective plate and those of its neighbors. Results showed that, across five simulations, the proposed algorithm was capable of establishing a social hierarchy that reduced the system-wide cost function to some locally-minimum value. Another observation from numerical results was that increasing the size of the distributed system (*i.e.* the number of plates and agents) had no discernible effect on the number of iterations required to minimize cost function values to local minima. This behavior was not observed when using a parallel DEMPC algorithm with no mechanism for addressing non-convexity.

Several research directions exist for further enhancing the proposed DEMPC algorithm; developing non-iterative algorithms using compatibility constraints as first proposed by Dunbar and Murray [53]; employing Lyapunov constraints to guarantee asymptotic stability rather than bounded stability; guaranteeing convergence, feasibility, and stability under the effects of coupled dynamics and constraints; ensuring robustness in the presence of bounded disturbances in the system dynamics and cost functions; application of the proposed algorithm to distributed systems with weak dynamic coupling such as autonomous vehicle trajectory planning and wind farm control.

Chapter 6

Social Hierarchy-based Distributed Economic Model Predictive Control of Floating Offshore Wind Farms

This chapter implements a recently developed social hierarchy-based distributed economic model predictive control (DEMPC) algorithm in floating offshore wind farms for the purpose of power maximization. The controller achieves this objective using the concept of *yaw and induction-based turbine repositioning* (YITuR), which minimizes the overlap areas between adjacent floating wind turbine rotors in real-time to minimize the wake effect. Floating wind farm dynamics and performance are predicted numerically using FOWFSim-Dyn. To ensure fast decision-making by the DEMPC algorithm, feed-forward neural networks are used to estimate floating wind turbine dynamics during the process of dynamic optimization. For simulated wind farms with layouts ranging from 1×2 to 1×5, an increase of $20\,\%$ in energy production is predicted when using YITuR instead of greedy operation. Increased variability in wind speed and direction is also studied and is shown to diminish controller performance due to rising errors in neural network predictions.

6.1 Introduction

This chapter outlines the implementation of distributed economic model predictive control (DEMPC) for the purpose of power maximization in floating offshore wind farms. This introductory section provides an overview of the control application along with a description of the DEMPC algorithm.

6.1.1 Application

Wind turbines that are aligned with the free stream wind suffer from a phenomenon termed the *wake effect*. Viscous interaction along the blades of an upstream machine generates a region downstream that is referred to as a *wake* in which the mean wind speed is reduced in comparison to that of the surrounding wind field. Any turbine that is located downstream and aligned with this wake encounters a lower average wind velocity relative to its upstream neighbor, and may thus operate with up to a 60 % reduction in power production [141].

One approach to mitigating the wake effect is *wind farm control*, which entails the operation of individual turbine actuators in a manner such that power production from the collective is increased. Two conventional wind farm control techniques have been widely investigated in the context of fixed-foundation wind farms; power de-rating, which involves sacrificing power production from upstream turbines to increase the mean wind speeds to which downstream machines are exposed; and wake steering, which requires operating upstream rotors with deliberate yaw misalignment to deflect their generated wakes in the cross-wind direction. The reader may refer to our comprehensive review article [102] for further information on the wake effect and wind farm control.

Floating wind farms render possible a third control strategy for power maximization. Namely, the mooring systems that anchor floating platforms to the seabed permit limited mobility of the platforms along the ocean surface. By operating a wind turbine with deliberate yaw misalignment, the direction of the net aerodynamic thrust force acting on its rotor may be altered, hence shifting its position along the ocean surface. As shown in Figure 6.1, this mobility may be exploited to reduce the overlap area between the wake generated by an upstream machine and the rotor of a downstream turbine. This wind farm control strategy is termed *yaw*

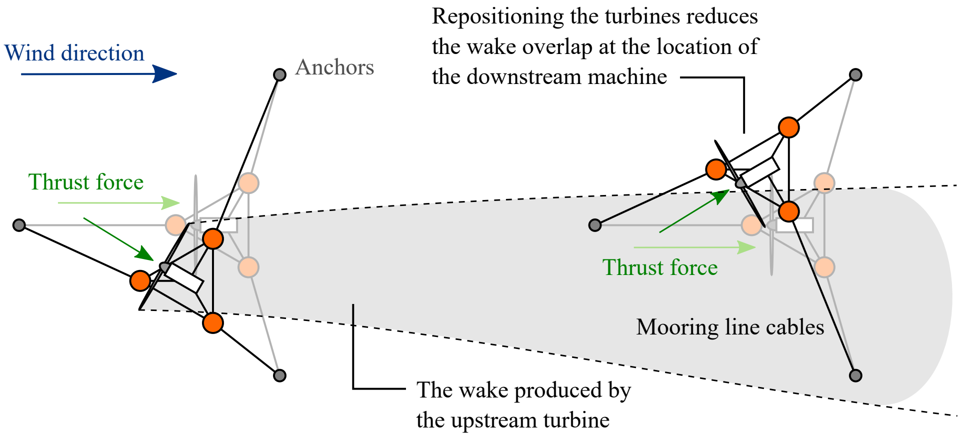

Wind direction
Anchors
Repositioning the turbines reduces the wake overlap at the location of the downstream machine
Thrust force
Thrust force
Mooring line cables
The wake produced by the upstream turbine

Figure 6.1: Schematic describing the use of aerodynamic thrust forces to passively relocate floating wind turbines in real-time.

and induction-based turbine repositioning (YITuR).

Our previous work [103] presented a steady-state model for predicting platform displacements and power production in floating offshore wind farms. Additionally, this work performed a stationary optimization study to evaluate the potential benefits of YITuR under steady-state conditions. To this end, we reported efficiency gains ranging from 21.8 to 42.7 % for wind farm sizes spanning from 2×2 to 7×7 provided that the mooring lines were long enough to permit crosswind relocation of roughly one rotor radius. Given this potential for increasing energy production, the aim of the current work is to transcend a steady-state analysis and to implement YITuR in real-time using optimal control techniques.

6.1.2 Controller

At any given sampling time, model predictive control (MPC) utilizes a mathematical model of a plant to solve a dynamic optimization problem over a finite prediction horizon. The output of this process is a sequence of optimal control inputs that minimize or maximize a prescribed cost function over the prediction horizon while satisfying constraints. Solely the first step in this sequence is then implemented, at which point, based on updated state and disturbance measurements, the dynamic optimization problem is recomputed at the next sampling time. More detail on

MPC is available in review articles by Mayne [125] and Mayne *et al.* [126].

The turbine repositioning wind farm control problem addressed in this chapter does not simply involve set-point stabilization; the control objective is to minimize overlap areas between the rotors of adjacent floating wind turbines. Given this property, economic model predictive control (EMPC) is a prime candidate for automation. EMPC computes optimal control actions by minimizing a *generalized cost function* that is a quantifier of plant performance over a finite prediction horizon. In comparison, traditional MPC minimizes a positive-definite cost function that simply penalizes deviation from a prescribed set-point. More detail on EMPC is available in the review article by Ellis *et al.* [55].

In order to promote scalability (*i.e.* computation time must not be affected by the size of the plant), it is necessary to distribute the optimal control problem among the various operational turbines within the plant; hence the employment of DEMPC in our approach. DEMPC eliminates the computational burden encountered by centralized EMPC as each agent (*i.e.* floating wind turbine) computes its own optimal control input sequence. In order to reach an optimal decision, every agent must make assumptions as to the state and control input trajectories of neighbors with which it is coupled. A major component of any DEMPC scheme is therefore the coordination algorithm that is used to update the assumptions that each agent holds regarding the future operation of its neighbors. More detail on DEMPC is available in the review article by Müller and Allgöwer [134].

The turbine repositioning problem possesses two properties that pose challenges in the implementation of DEMPC. The first is that objective functions are non-convex. There exist multiple platform displacement paths that reduce the overlap areas between the rotors of adjacent floating turbines. In standard parallel DEMPC algorithms, convergence of agent decisions cannot be guaranteed in the presence of non-convex cost functions [116]; more elaborate sequential [109, 151], negotiation-based [120, 135, 170], and group-based [12, 117, 144] coordination algorithms are therefore required to establish convergence. These algorithms suffer from scalability issues however (see Chapter 5). The second property that poses a challenge is that steady-state terminal set-points are not known *a priori* in the current application. In brief, terminal set-points are necessary for achieving stability in EMPC as they ensure that the computed optimal control actions always lead the system to a stable steady-state [55]. Existing DEMPC methods either assume that these set-points are known *a priori* [184], are determined using centralized optimization [6, 36, 50, 111, 112, 185], or are computed in a distributed manner assuming convex cost functions [108]. Our previous work (see Chapter 5) overcame these drawbacks and presented a DEMPC algorithm that used the concept of social hierarchies to guarantee convergence and stability in the presence of non-convex cost functions and unknown terminal set-points in a truly distributed and scalable manner (*i.e.* no centralized functionalities were required).

6.1.3 Contributions

The main contribution of this chapter is the first implementation of a social hierarchy-based DEMPC algorithm for the purpose of power maximization via YITuR in floating offshore wind farms. A secondary contribution includes the first use of feed-forward artificial neural networks to generate a computationally inexpensive control-oriented model of floating wind turbine dynamics. These neural networks serve as the mathematical model used in the dynamic optimization process of the DEMPC algorithm. The final contribution is the first demonstration of the benefits of YITuR implemented in real-time under time-varying wind conditions.

6.1.4 Chapter organization

The chapter is organized as follows: Section 6.2 describes FOWFSim-Dyn, which is a simulator used to predict platform motion, wake aerodynamics, and wind farm power production; Section 6.3 discusses the social hierarchy-based DEMPC objective and algorithm, the use of feed-forward neural networks to approximate turbine dynamics, and the impacts of disturbances and model uncertainty on controller performance; Section 6.4 provides simulation results of controller implementation corresponding to different wind conditions and turbine array sizes; and Section 6.5 concludes the chapter with summaries of major findings and recommendations for future research.

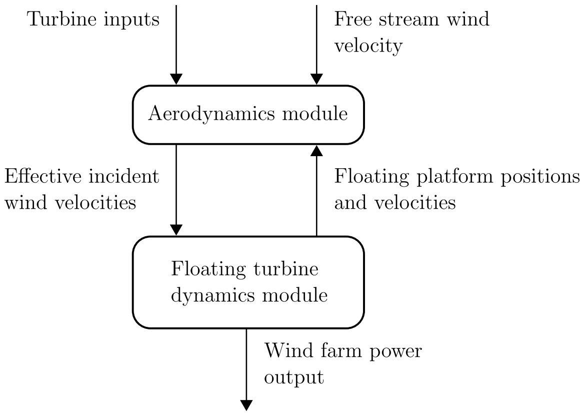

Turbine inputs
Free stream wind velocity
Aerodynamics module
Effective incident wind velocities
Floating platform positions and velocities
Floating turbine dynamics module
Wind farm power output

Figure 6.2: Block diagram showing the computation modules of FOWFSim-Dyn along with information transfer routes.

6.2 Simulation tool - FOWFSim-Dyn

This section briefly describes FOWFSim-Dym, which is a tool used to simulate floating offshore wind farms in the current work. Full details of this software are available in Chapter 4. A system diagram of FOWFSim-Dyn is shown in Figure 6.2. Two high-level modules are tasked with predicting floating platform motion coupled with wake aerodynamics. The following subsections delineate each of these modules.

6.2.1 Floating turbine dynamics module

The purpose of the floating turbine dynamics module is to predict the positions, velocities, and power outputs of all floating wind turbines within the wind farm. As shown in Figure 6.3, a floating wind farm is treated as system of particles distributed along the two-dimensional ocean surface. The predominant free stream wind direction is aligned with the global $\hat{x}$ axis; however, FOWFSim-Dyn is capable of simulating fluctuations in the free stream wind direction relative to the $\hat{x}$ axis. The wind velocity vector is denoted as $\mathbf{V}_\infty$.

The variables x_i and y_i describe the positions of turbine i along the $\hat{x}$ and $\hat{y}$ axes over time. The velocities of the platform of turbine i in these directions are denoted by $v_{x,i}$ and $v_{y,i}$. Finally, turbine i possesses two inputs; the rotor's axial induction

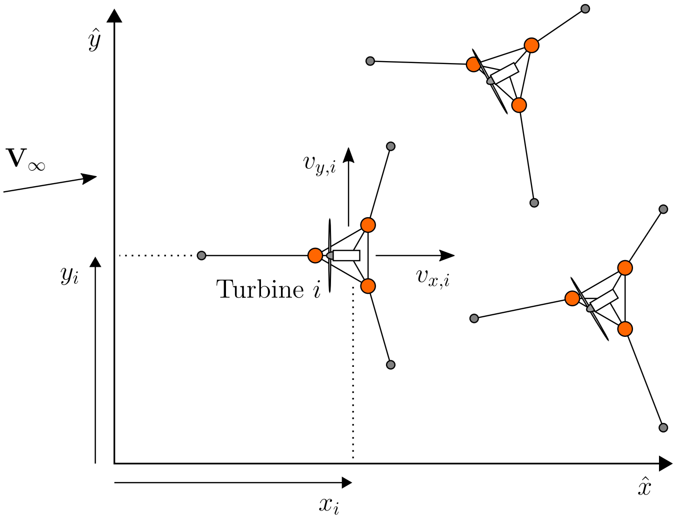


Figure 6.3: Schematic of the general floating offshore wind farm used in FOWFSim-Dyn.

factor a_i and its yaw angle γ_i measured positive counter-clockwise relative to the $\hat{x}$ axis.

The motions of the floating platforms are determined by summing the aerodynamic thrust forces, hydrodynamic drag and added mass forces, and mooring line tensions acting on each turbine. Hydrodynamic drag and added mass forces are calculated using Morison's equation [152], wherein the drag and added mass contributions of all submerged components are summed to provide the total hydrodynamic force. Mooring line tensions are obtained by solving the problem of a static catenary that is either partially resting along or fully lifted above the seabed [103].

Aerodynamic thrust forces are estimated by treating wind turbine rotors as actuator discs and computing thrust coefficients according to vortex cylinder theory [29]. Computation of aerodynamic thrust forces also requires estimates of the effective wind velocity that is incident upon the rotors of all turbines. The wake aerodynamics module addresses this requirement.

6.2.2 Aerodynamics module

The wake aerodynamics module predicts characteristics of the wakes generated by each turbine over time. These characteristics include the wake centerline position, wake diameter, and the average wake velocities in the $\hat{x}$ and $\hat{y}$ directions. The evolution of these properties over space and time is captured using one-dimensional momentum conservation with an assumed wake expansion rate.

The boundary conditions to this problem include the initial wake velocities at the location of the turbine that generates each wake. The initial wake velocities in the downwind and crosswind direction are estimated using momentum balance methods proposed by Bastankhah *et al.* [18] and Jiménez *et al.* [93].

The effective wind velocity that is incident upon the rotor of a downstream turbine is computed using the principle of equivalent kinetic power deficit [98]. Namely, the net kinetic power deficit at any downstream location is assumed to equate to the sum of kinetic power deficits contributed by all upstream wakes. Further, wake velocity profiles are modeled using Gaussian distributions as per experimental results reported by Bastankhah *et al.* [18].

6.3 Controller design

This section first describes the optimal control objective for maximizing power production in floating offshore wind farms. Then, the social hierarchy-based DEMPC algorithm (see Chapter 5) is briefly described. Next, the use of feed-forward neural networks for estimating floating platform motion during dynamic optimization processes is discussed. These neural networks treat aerodynamic coupling between turbines as disturbances. The current section therefore also discusses the impact of disturbance on the stability and performance of EMPC.

6.3.1 Controller objective functions

Simply put, the objective of DEMPC is to indirectly maximize power production by minimizing the overlap areas between the rotors of adjacent turbines. Power production is not directly maximized since this approach would require the identification of functions that estimate the power output of each turbine using some machine learning scheme. Identification of such functions is challenging since, due to the transport phenomenon, the power produced by any turbine is strongly influenced by the *history* of the states and inputs of its upstream neighbors. In other words, the states and inputs that upstream neighbors would have possessed in the past influence the power produced by a downstream turbine at the present moment. Identification of a power function is therefore deferred to future work who's primary focus is dynamic system identification of floating offshore wind farms. In the current work, the indirect approach of rotor area overlap minimization is taken.

The resulting dynamic optimization problem to be solved by the local EMPC function of turbine i takes the following form:

$$\min_{\bar{\mathbf{x}}_i, \bar{\mathbf{u}}_i} \sum_{k=0}^{H-1} \left\{ \Delta\mathbf{u}_{i,k}^{\mathrm{T}} \mathbf{Q} \Delta\mathbf{u}_{i,k} + \frac{1}{|\mathscr{N}_i|} \sum_{j \in \mathscr{N}_i} \left[A_{\mathrm{OL},i\to j}(\mathbf{x}_{i,k}, \mathbf{x}_{j,k}) + \Delta\mathbf{u}_{j,k}^{\mathrm{T}} \mathbf{Q} \Delta\mathbf{u}_{j,k} \right] \right\}, \quad (6.1)$$

subject to

$$\begin{aligned} \mathbf{x}_{i,0} &= \mathbf{x}_i, & (6.2a) \\ \mathbf{x}_{i,k+1} &= \mathbf{f}_i(\mathbf{x}_{i,k}, \mathbf{u}_{i,k}), & (6.2b) \\ a_{i,k} &= \frac{1}{3}, & (6.2c) \\ |\gamma_{i,k}| &\leq 10\,\mathrm{deg}, & (6.2d) \\ \mathbf{x}_{i,H} &= \mathbf{x}_{i,s}, & (6.2e) \end{aligned}$$

where H is the length of the prediction horizon, and k is the time-step number along the prediction horizon. The vector $\mathbf{x}_i$ contains the most recent state measurements of turbine i as follows:

$$\mathbf{x}_i := \begin{bmatrix} x_i & y_i & v_{x,i} & v_{y,i} \end{bmatrix}^{\mathrm{T}}, \quad (6.3)$$

and the vectors $\mathbf{x}_{i,k}$ and $\mathbf{u}_{i,k}$ contain the candidate states and inputs of turbine i at time-step number k along the prediction horizon as follows:

$$\begin{aligned} \mathbf{x}_{i,k} &:= \begin{bmatrix} x_{i,k} & y_{i,k} & v_{x,i,k} & v_{y,i,k} \end{bmatrix}^{\mathrm{T}}, & (6.4) \\ \mathbf{u}_{i,k} &:= \begin{bmatrix} a_{i,k} & \gamma_{i,k} \end{bmatrix}^{\mathrm{T}}. & (6.5) \end{aligned}$$

The terms $\overline{\mathbf{x}}_i := (\mathbf{x}_{i,0}, \mathbf{x}_{i,1}, \cdots, \mathbf{x}_{i,H})$ and $\overline{\mathbf{u}}_i := (\mathbf{u}_{i,0}, \mathbf{u}_{i,1}, \cdots, \mathbf{u}_{i,H-1})$ denote the candidate state and input trajectories of turbine i. The term $\Delta\mathbf{u}_{i,k}$ symbolizes the input vector of turbine i at time-step number k measured relative to the reference $\left[\frac{1}{3}\ 0\,\text{deg}\right]^{\mathrm{T}}$. This reference is used so that only deviation of the axial induction factor relative to its optimal value of $\frac{1}{3}$ is penalized in the cost function. $\mathbf{Q}$ is a weighting matrix that regulates the significance of this penalty. The set $\mathcal{N}_i$ contains the indices of all agents that are physically adjacent to turbine i. Finally, $A_{\mathrm{OL},i\to j}(\cdot)$ is a function that computes the overlap area between the rotors of turbines i and j and then normalizes this value based on the rotor swept area.

Constraint (6.2a) simply states that the initial condition along the prediction horizon must equate to the latest state measurement. Constraint (6.2b) relates the candidate state and input vector trajectories via a model of the dynamics of turbine i. This dynamic model $\mathbf{f}_i(\cdot)$ is estimated using feed-forward neural networks in Section 6.3.2. Finally, constraints (6.2c) and (6.2d) limit variations of the axial induction factor and yaw angle of turbine i. In the case of yaw angles, values are limited to $\pm 10\,\text{deg}$ for operational safety. In the current work, the axial induction factors of all turbines are held fixed at the optimal operating point of $\frac{1}{3}$. In future works that attempt to directly maximize power production, the axial induction factor may be permitted to vary since its impact is considered in the power estimation function.

Constraint (6.2e) is the terminal set-point. It requires that the optimal state trajectory $\overline{\mathbf{x}}_i$ ends at a stationary state vector $\mathbf{x}_{i,s}$, which is a candidate steady-state. The computation of $\mathbf{x}_{i,s}$ requires solving the following optimization problem at each time-step:

$$\min_{\overline{\mathbf{x}}_i, \mathbf{x}_{i,s}, \overline{\mathbf{u}}_i, \mathbf{u}_{i,s}} \Delta\mathbf{u}_{i,s}^{\mathrm{T}}\mathbf{Q}\Delta\mathbf{u}_{i,s} + \frac{1}{|\mathcal{N}_i|}\sum_{j\in\mathcal{N}_i}\left[A_{\mathrm{OL},i\to j}(\mathbf{x}_{i,s}, \mathbf{x}_{j,s}) + \Delta\mathbf{u}_{j,s}^{\mathrm{T}}\mathbf{Q}\Delta\mathbf{u}_{j,s}\right], \tag{6.6}$$

subject to constraints in Eq. (6.2) along with

$$\begin{aligned} \mathbf{x}_{i,s} &= \mathbf{f}_i(\mathbf{x}_{i,s}, \mathbf{u}_{i,s}), && \text{(6.7a)} \\ a_{i,s} &= \frac{1}{3}, && \text{(6.7b)} \\ |\gamma_{i,s}| &\leq 10\,\text{deg}, && \text{(6.7c)} \end{aligned}$$

where $\mathbf{u}_{i,s}$ is the candidate optimal stationary input vector of turbine i, and $\Delta\mathbf{u}_{i,s}$ denotes $\mathbf{u}_{i,s}$ measured relative to the reference $\left[\frac{1}{3}\ 0\,\mathrm{deg}\right]^{\mathrm{T}}$. Constraint (6.7a) requires that $\mathbf{x}_{i,s}$ and $\mathbf{u}_{i,s}$ correspond to a stationary point. Similar to Problem (6.1), constraints (6.7b) and (6.7c) serve as limits on variations of the axial induction factor and yaw angle at steady-state.

The difference between Problems (6.1) and (6.6) is that the latter does not minimize the stage cost function throughout the prediction horizon. Instead, it only minimizes overlap areas and actuator deviation at steady-state. It therefore computes an optimal stationary point independent of the trajectory that leads to this point. However, since the constraints require that $\mathbf{x}_{i,H} = \mathbf{x}_{i,s}$, the computed optimal steady-state is guaranteed to be reachable given the current state measurement. The vector $\mathbf{x}_{i,s}$ is therefore a candidate *reachable* steady-state.

6.3.2 Feed-forward artificial neural network

In order to hasten the dynamic optimization process of the DEMPC algorithm, feed-forward neural networks are used to approximate the dynamic model $\mathbf{f}_i(\cdot)$ for any turbine i. This estimated model is solely a function of the states and inputs of turbine i and thus treats aerodynamic coupling as disturbance or model uncertainty.

Each neural network takes the form shown in Figure 6.4. The six input neurons represent the states $\mathbf{x}_{i,k}$ and inputs $\mathbf{u}_{i,k}$ of the corresponding turbine at some time-step number k along the prediction horizon. This input layer feeds into a single hidden layer consisting of 20 neurons. The output layer contains four neurons representing the turbine states $\mathbf{x}_{i,k+1}$ at the subsequent time-step number $k+1$.

To generate training data for tuning the neural networks, FOWFSim-Dyn simulations consisting of 10^5 time-steps with sampling periods of 60 sec were completed for each wind farm configuration that was examined. The free stream wind velocity was fixed at 8 m/s in the direction of the positive $\hat{x}$ axis. The initial conditions of each simulation were computed by operating all turbines with an axial induction factor of $\frac{1}{3}$ and a yaw angle of 0 deg. Then, over the course of the simulations, turbine inputs were randomly varied with uniform probability of 0.1 (*i.e.* there was a 10 % chance that the input values would change at each new time-step). The logic behind this variation approach was to allow enough time for the

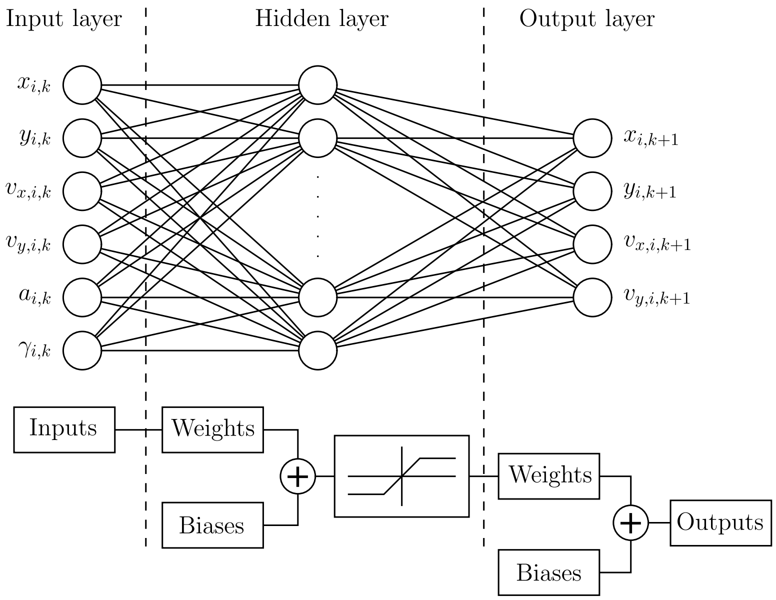


Figure 6.4: Schematic of the local neural network structure used to identify the dynamic model of each floating wind turbine.

turbine platforms to respond to each change in input values. Axial induction factors were varied between 0.2 and 0.4, while yaw angles were varied between -20 and $+20$ deg.

Sample validation data is provided for the two-turbine floating wind farm described in Figure 6.5. The turbine and platform designs are detailed in the works of Jonkman *et al.* [96] and Robertson *et al.* [152], except mooring line lengths have been increased from 835 to 950 m to increase platform range of motion. In their neutral positions, the platforms are spaced seven rotor diameters apart, which equates to 882 m.

For validation, FOWFSim-Dyn and the neural networks were given the same randomly generated initial condition and turbine input sequences over 60 timesteps (*i.e.* 3,600 sec). Their predictions of the system response are compared for both turbines 1 and 2 in Figure 6.6 and Figure 6.7, respectively. These plots correspond to a single run, and have been provided for qualitative inspection. It is apparent that neural network predictions for turbine 1 more closely match simulation results. This outcome is expected since turbine 1 is the leading machine and is

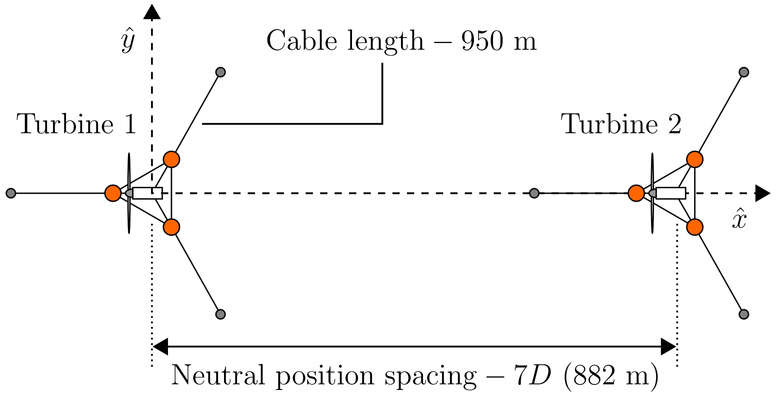


Figure 6.5: Schematic of a two-turbine floating offshore wind farm simulated in the current work. Wind turbine and floating platform properties are based on the National Renewable Energy Laboratory's (NREL's) baseline designs presented by Jonkman *et al.* [96] and Robertson *et al.* [152]. The sole modification is that mooring line lengths have been increased from 835 to 950 m to permit greater platform displacement.

thus uninfluenced by aerodynamic coupling. The behaviour of turbine 2 is affected by the states and inputs of turbine 1 however, which induce greater prediction errors in the behavior of turbine 2 since its neural network does not consider dynamic coupling.

For a quantitative assessment of neural network performance, ten different runs were completed to acquire the average root-mean-square-error (RMSE) of each neural network output. This information is listed in Table 6.1. Neural network predictions of x_i deviate from FOWFSim-Dyn results with RMSE values of 0.94 and 5.79 m for turbines 1 and 2, respectively. The greater RMSE associated with turbine 2 is attributed to the absence of dynamic coupling in the neural networks. This trend reverses in the case of y_i predictions. Since turbine 1 is unaffected by any upstream wakes, it encounters higher wind speeds and thus displaces a greater distance in the crosswind direction relative to turbine 2 when operating with yaw misalignment. As a result, the RMSE value of y_1 is inflated in comparison to that of y_2. These arguments apply equally well with regards to platform velocity RMSE values. Successful control of the current system therefore requires that local EMPC algorithms are capable of rejecting these prediction errors. A description of the so-

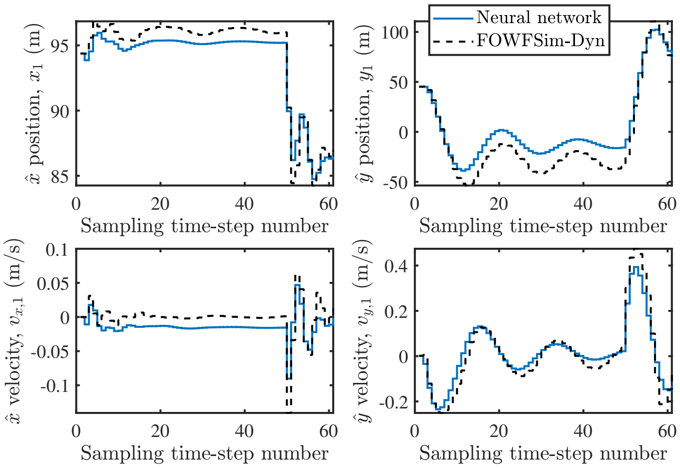


Figure 6.6: Sample validation data for the neural network of the upstream machine (*i.e.* turbine 1) in a two-turbine floating wind farm.

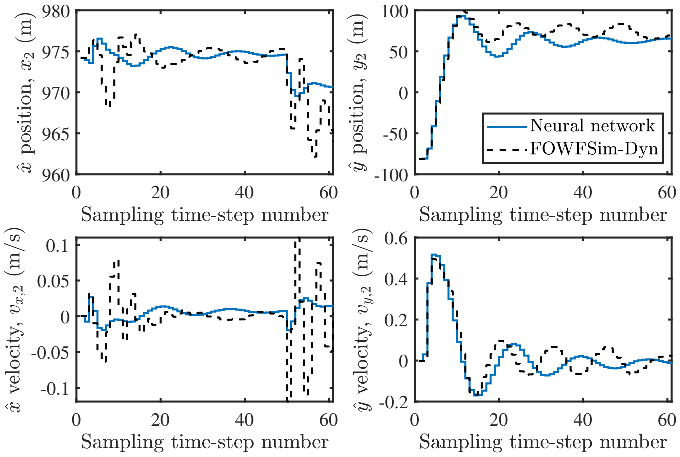


Figure 6.7: Sample validation data for the neural network of the upstream machine (*i.e.* turbine 2) in a two-turbine floating wind farm.

Table 6.1: List of RMSE values between neural network predictions and FOWFSim-Dyn predictions.

Turbine	x_i (m)	y_i (m)	$v_{x,i}$ (m/s)	$v_{y,i}$ (m/s)
1	0.94	14.68	0.02	0.08
2	5.79	11.89	0.05	0.06

cial hierarchy-based DEMPC scheme now follows.

6.3.3 Social hierarchy-based DEMPC algorithm

If multiple optimal solutions exist in an optimal control problem (*i.e.* the cost functions are non-convex), and decisions are made in a synchronous and distributed manner by multiple agents, then there exists a potential for conflicting decisions (see Chapter 5). It is due to this logic that convergence of the decisions made by multiple agents cannot be guaranteed in parallel distributed optimal control problems [116]. The social hierarchy-based DEMPC algorithm used in this work was therefore originally developed to address non-convexity in distributed optimal control problems. The essential function of this algorithm is that all agents self-organize in a fully distributed manner into a social hierarchy that dictates the order in which each agent makes a decision.

The algorithm borrows from evolutionary principles. Namely, if an agent's current level along a social hierarchy yields decisions that conflict with those of its neighbors, then the agent's current level is detrimental to performance and must be randomly varied. The method with which conflicting decisions are identified is now described. At any iteration, each agent possesses assumed values of the state and input trajectories of its neighbors. Based on these assumptions, and given some existing social hierarchy that determines the order in which agents make decisions, the agent updates its own optimal trajectories and calculates the resulting value of its neighborhood cost function. We refer to this value as the *naive* cost function. After receiving updated optimal state and input trajectories from its neighbors, the agent recomputes the value its neighborhood cost function. We refer to this updated value as the *informed* cost function. If the informed cost function is improved in comparison to the naive cost function, then the agent's decisions are mutually beneficial with those of its neighbors. Otherwise, the decisions are conflicting and the agent must randomly change its place along the social hierarchy. It was proved in our previous work that, for any interaction topology, at least one social hierarchy exists that enables the agents to avoid conflicting decisions; thus establishing a guarantee of convergence.

Another unique property of this algorithm is that it was developed without the presumption of terminal set-points that are known *a priori*. Instead, it was intended for the agents to determine these terminal targets in a fully distributed manner. With mild similarity to the approach taken by Kohler et al. [108], the agents compute their terminal set-points at each time-step by first solving an optimization problem which minimizes their cost functions at steady-state. This steady-state set-point is utilized as a terminal constraint in the subsequent optimization problem that yields optimal dynamic state and input trajectories. The computation of the steady-state set-points also makes use of the social hierarchy-based approach to eliminate conflicting decisions.

6.3.4 Effects of dynamic coupling

In the current work, feed-forward neural networks are used to locally estimate the dynamic model of each floating wind turbine. These estimated models are better suited for dynamic optimization due to their low computation time requirements. Since these neural networks are tuned locally, they are solely functions of the inputs and states of their corresponding turbines. They therefore do not consider the inputs and states of turbines with which aerodynamic coupling exists. Dynamic coupling may thus be interpreted as a source of disturbance or model uncertainty in our work.

EMPC is inherently capable of rejecting disturbance or model uncertainty that is appropriately bounded in magnitude given the dynamics of the plant. The impact of disturbances in EMPC is first demonstrated in the context of terminal equality constraint satisfaction in Figure 6.8. Let the set $\mathscr{X}_r$ contain all initial state vectors $\mathbf{x}_0$ from which a terminal set-point $\mathbf{x}_s$ is reachable in H time-steps while satisfying process constraints. As long as $\mathbf{x}_0$ is contained within the set $\mathscr{X}_r$, then an input trajectory $\overline{\mathbf{u}} = (\mathbf{u}_0, \mathbf{u}_1, \cdots, \mathbf{u}_{H-1})$ exists that generates a state trajectory $\overline{\mathbf{x}} = (\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_H)$ which ends at $\mathbf{x}_H = \mathbf{x}_s$. Disturbance or model uncertainty

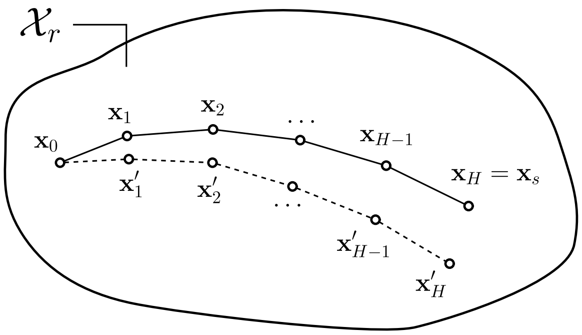


Figure 6.8: Schematic showing the impacts of disturbance/model uncertainty on terminal equality constraint satisfaction in EMPC

causes a deflection of the state trajectory from $\overline{\mathbf{x}}$ to $\overline{\mathbf{x}}' = (\mathbf{x}_0, \mathbf{x}_1', \cdots, \mathbf{x}_H')$ as the system moves forward in time. Provided that the magnitude of disturbance or model uncertainty is appropriately bounded, then $\mathbf{x}_H'$ will not leave the set $\mathscr{X}_r$. As a result, even if the system follows the deflected path from $\mathbf{x}_0$ to $\mathbf{x}_H'$, then a path from $\mathbf{x}_H'$ to $\mathbf{x}_s$ is guaranteed to exist. If, on the other hand, the magnitude of disturbance is large enough to cause $\mathbf{x}_H'$ to leave $\mathscr{X}_r$, the satisfaction of the terminal constraint cannot be guaranteed and system redesign is recommended.

Next, the impact of disturbance on the satisfaction of trajectory constraints is discussed. Let the set $\mathscr{X}$ contain all acceptable values of the state vector $\mathbf{x}_k$ for all $k \in \mathbb{I}_{0:H}$. In other words, the state trajectory $\overline{\mathbf{x}}$ must always be contained within $\mathscr{X}$. As before, disturbance or model uncertainty deflects $\overline{\mathbf{x}}$ to $\overline{\mathbf{x}}'$ as the system moves forward in time. Provided that the magnitude of disturbance is appropriately bounded, then $\overline{\mathbf{x}}'$ will not leave the set $\mathscr{X}$ and state constraints will be satisfied. This expectation is conservative however. If the magnitude of disturbance is not appropriately bounded, then the technique of constraint tightening [37] may be used to ensure constraint satisfaction.

The mechanism of constraint tightening is demonstrated in Figure 6.9. Once again, it is required that $\overline{\mathbf{x}}$ remains within the set $\mathscr{X}$. To ensure this outcome, the set $\mathscr{X}$ may be duplicated and *tightened* to form the set $\mathscr{X}'$. The dynamic optimization problem must then be required to maintain $\overline{\mathbf{x}}$ within $\mathscr{X}'$. If $\mathscr{X}'$ is sized accordingly, then, even with the worst-case disturbance scenario, the deflected state trajectory $\overline{\mathbf{x}}'$ is guaranteed to remain within the original process constraint set $\mathscr{X}$.

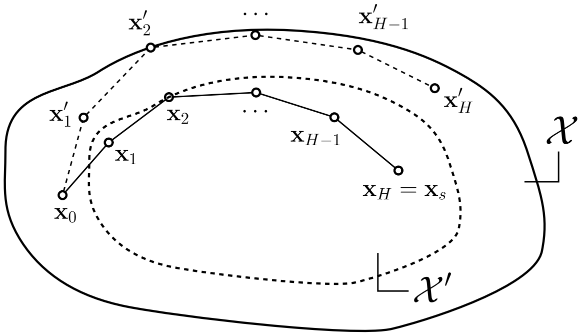


Figure 6.9: Schematic demonstrating the method of constraint tightening for process constraint satisfaction in EMPC.

6.4 Results and discussion

The current section presents simulation results corresponding to the implementation of the DEMPC scheme outlined in Section 6.3 on floating wind farms of different sizes. Each simulated wind farm possesses a row configuration similar to that shown in Figure 6.5. Inter-turbine neutral position spacings are seven rotor diameters and mooring line cable lengths are 950 m, while the sizes of the wind farms range from 1×2 to 1×5. First, the impact of wind speed and direction variation on controller performance is assessed for a 1×2 wind farm. Then, the performance and behaviour of larger array sizes are investigated.

6.4.1 Wind velocity variation

The neural networks described in Section 6.3.2 were tuned using data corresponding to 8 m/s free stream wind aligned with the positive $\hat{x}$ axis. As the free stream wind speed and direction deviate from this condition, the error in neural network outputs will increase and cause detriment to controller performance. We therefore assess in this subsection the impacts of free stream wind velocity fluctuations on platform motion and optimal power production.

To quantify velocity fluctuations, we define the parameter $0 \leq \sigma_\infty \leq 1$ and use it to randomly perturb, with uniform probability, the ten-minute averaged free stream wind velocity vector. Therefore, at some ten-minute mark t, the free stream wind

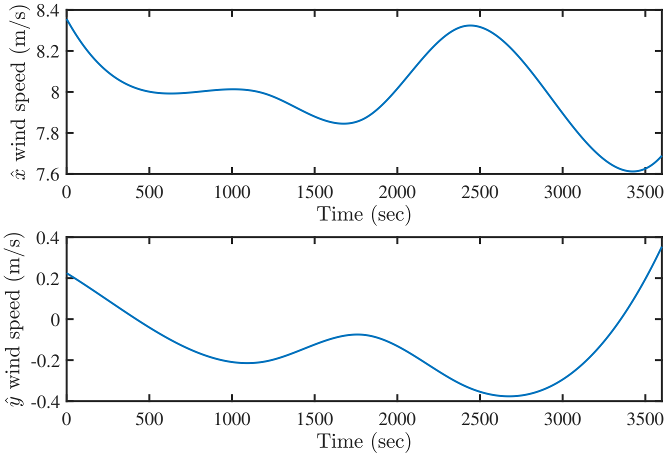


Figure 6.10: Sample randomly generated free stream wind speed evolutions in the $\hat{x}$ and $\hat{y}$ directions corresponding to a perturbation parameter of $\sigma_\infty = 5\,\%$.

velocity vector $\mathbf{V}_\infty(t)$ is computed as follows:

$$\mathbf{V}_\infty(t) = \overline{\mathbf{V}}_\infty + \begin{bmatrix} \text{rand}_{\pm\sigma_\infty \left\|\overline{\mathbf{V}}_\infty\right\|} \\ \text{rand}_{\pm\sigma_\infty \left\|\overline{\mathbf{V}}_\infty\right\|} \end{bmatrix}, \tag{6.8}$$

where $\overline{\mathbf{V}}_\infty$ is the baseline free stream velocity vector of 8 m/s aligned with the positive $\hat{x}$ axis, and $\text{rand}_{\pm\sigma_\infty \left\|\overline{\mathbf{V}}_\infty\right\|}$ is a function that generates a random number between $-\sigma_\infty \left\|\overline{\mathbf{V}}_\infty\right\|$ and $+\sigma_\infty \left\|\overline{\mathbf{V}}_\infty\right\|$. Spline interpolation is then used to generate smoother curves with 0.1 sec resolution using these randomly generated ten-minute averages. Sample free stream wind velocity curves in the $\hat{x}$ and $\hat{y}$ directions corresponding to $\sigma_\infty = 5\,\%$ are plotted in Figure 6.10.

6.4.2 Social hierarchy-based DEMPC properties

The relevant properties of the social hierarchy-based DEMPC algorithm are listed in Table 6.2. Only two social hierarchy levels are used since, in order to prevent conflicting decisions, each turbine must be given the opportunity to compute its optimal trajectory out of synchrony with its adjacent neighbors. Within each

Table 6.2: List of properties relevant to the social hierarchy-based DEMPC algorithm.

Property	Value
Number of social hierarchy levels	2
Number of iterations per sampling period	3
Number of prediction horizon time-steps, H	5
Input weight matrix, $\mathbf{Q}$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

sampling period, each turbine is granted three iterations to compute its optimal stationary point by solving Problem (6.6), as well as three additional iterations to compute its optimal state and input trajectories by solving Problem (6.1).

The length of the prediction horizon is set to $H = 5$. Since the sampling period of each controller is 60 sec, this value of H permits prediction of 5 min into future. Finally, the input weight matrix $\mathbf{Q}$ used in Problems (6.6) and (6.1) is set the identity matrix. Since the overlap areas between turbine rotors are normalized in the cost functions of Problems (6.1) and (6.6), input weights of unity render the optimization problem approximately circular. An identity matrix was therefore used a starting point for $\mathbf{Q}$ and the corresponding results were deemed acceptable.

Regardless of the wind farm size, the time required by the DEMPC algorithm to compute optimal state and input trajectories *per turbine* ranged from 10 to 50 sec. This range of durations is smaller than the sampling period of 60 sec; the social hierarchy-based DEMPC algorithm may therefore be implemented in real-time. It is important to note that, if a sequential DEMPC algorithm [109, 151] had been used, then the total computation time would be a summation of the durations required by all turbines. This property exists since a sequential algorithm requires that individual turbines take turns to make decisions. In other words, even for a two-turbine wind farm, the total required computation time would range from 20 to 100 sec, which exceeds the 60 sec sampling period.

6.4.3 Effects of wind velocity variations

The impact of time-varying free stream wind velocity on controller performance is assessed for a 1×2 wind farm. Evolutions of turbine $\hat{y}$ positions for different σ_∞ values are shown in Figure 6.11. As desired, the two floating wind turbines relocate

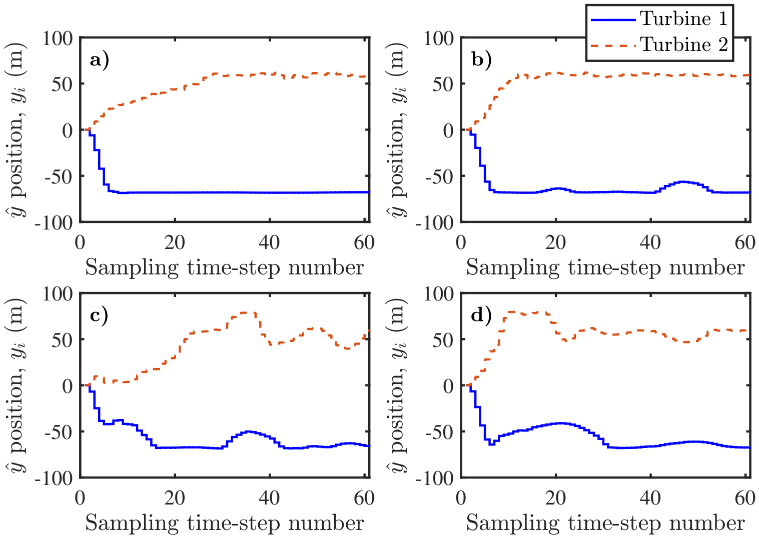


Figure 6.11: Evolution of $\hat{y}$ positions in a two-turbine floating wind farm that is optimally controlled by the DEMPC from Section 6.3. The four plots correspond to different wind velocity perturbations of a) $\sigma_\infty = 5\,\%$, b) $\sigma_\infty = 10\,\%$, c) $\sigma_\infty = 15\,\%$, and d) $\sigma_\infty = 20\,\%$.

in opposite directions in all cases to minimize their respective rotor overlap areas. Specifically, the turbines shift by approximately one rotor radius (*i.e.* 60 m), which results in a total clearance of one rotor diameter between their nacelle centerlines.

The impact of neural network prediction error is also evident in two aspects. First, regardless of the wind velocity perturbation parameter σ_∞, the downstream turbine's $\hat{y}$ position shows greater variability over time. The reason is that the dynamics of the downstream turbine (*i.e.* turbine 2) are influenced by the states and inputs of the upstream machine. Since this coupling was not explicitly considered in the machine learning process, the neural network of turbine 2 is subject to greater prediction error. This error translates to deflections of optimal state trajectories as discussed in Section 6.3.4.

Second, as the wind velocity perturbation σ_∞ is increased from 5 to 20 %, the $\hat{y}$ positions of both turbines show greater variability over time. This outcome is also explained by neural network prediction error. The neural networks have been tuned using data corresponding to a free stream wind velocity of 8 m/s aligned with the $\hat{x}$ axis. As the wind velocity deviates from this reference, the neural network uncertainties are exacerbated, once again leading to deflections in state trajectories relative to computes optimal values.

Evolutions of wind farm power production for different σ_∞ values are shown in Figure 6.12. This figure displays power production trends corresponding to both DEMPC and *greedy*[1] operation subject to identical free stream wind conditions. The greedy performance curves demonstrate the amount of power produced without wind farm control. Any rise in power production in the greedy case is consequently purely a result of free stream wind speed fluctuations. The benefit incurred by implementing YITuR is therefore quantified by the difference between the DEMPC and greedy power trajectories. With 5 % variability in the wind velocity, YITuR yields an 18.4 % gain in energy production relative to greedy operation throughout the hour-long simulation.

An additional observation from Figure 6.12 is that the increased variability of the free stream wind velocity diminishes wind farm controller performance. As σ_∞ is raised from 5 to 20 %, the relative gain in wind farm energy production obtained by switching from greedy operation to optimal control decreases from 18.4 to 7.3 %. This outcome is partly caused by the aforementioned rise in neural network prediction error that results from wind speed and direction deviations from the values used for neural network tuning. As seen in Figure 6.11, this error causes turbine $\hat{y}$ displacements to decrease for brief periods, which increases rotor overlap areas. Another factor is that, as the free stream wind direction varies and misaligns from the row of turbines, wake overlap naturally subsides. As a result, wind farm control fails to yield much benefit relative to greedy operation.

Platform displacements in the $\hat{x}$ direction are not considered in the DEMPC optimization problems; their progressions are therefore entirely dictated by the free stream wind velocity and aerodynamic coupling. Insight may nonetheless be gained by examining their trends. Evolutions of turbine $\hat{x}$ displacements (*i.e.* relative to neutral positions) are shown in Figure 6.13. The first observation is that turbine 2 consistently displaces a smaller distance in the downwind direction in

[1] Greedy operation implies the absence of wind farm control. That is to say, each wind turbine is operated with an axial induction factor and yaw angle that maximize its own power output; these greedy values are $a_i = \frac{1}{3}$ and $\gamma_i = 0\,\text{deg}$.

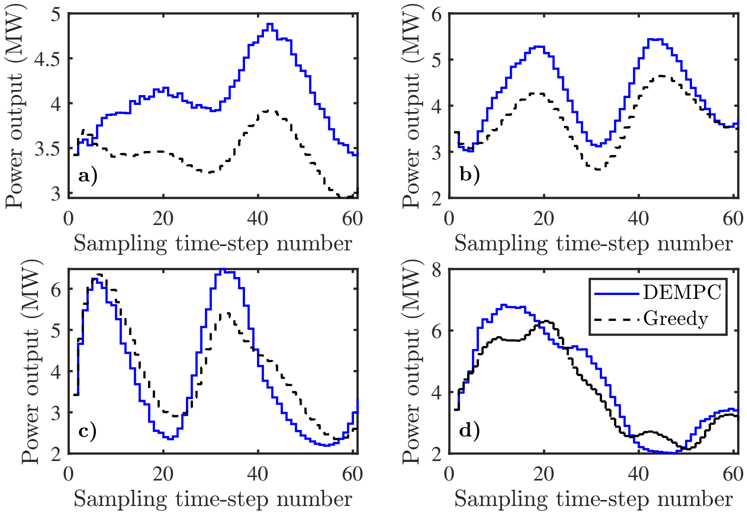


Figure 6.12: Evolution of power production in a two-turbine floating wind farm that is optimally controlled by the DEMPC from Section 6.3. For comparison, power production under greedy operation is also demonstrated. The four plots correspond to different wind velocity perturbations of a) $\sigma_\infty = 5\,\%$, b) $\sigma_\infty = 10\,\%$, c) $\sigma_\infty = 15\,\%$, and d) $\sigma_\infty = 20\,\%$.

comparison to turbine 1 for all wind velocity scenarios. The reason is simply that, as the downstream machine, turbine 2 encounters slower wind speeds on average due to the presence of the wake produced by its upstream neighbor.

A second trend from Figure 6.13 shows that the downwind displacement of turbine 2 is more volatile over time. This behaviour is once again the result of aerodynamic coupling. As turbine 1 adjusts its yaw angle and relocates its platform, the resultant effects on the wind field are transported downwind to the location of turbine 2. The downstream machine therefore encounters greater variability in the effective wind velocity that is incident upon its rotor. Furthermore, larger variations in the free stream wind velocity cause greater volatility in the downwind displacement of both turbines. In the case of $\sigma_\infty = 5\,\%$, the $\hat{x}$ displacement of turbine 2 ranges from 85 to 100 m. This range expands to 75 to 105 m at $\sigma_\infty = 20\,\%$. This outcome indicates that some additional control function may be required to limit excessive platform displacements in the downwind direction. This function may

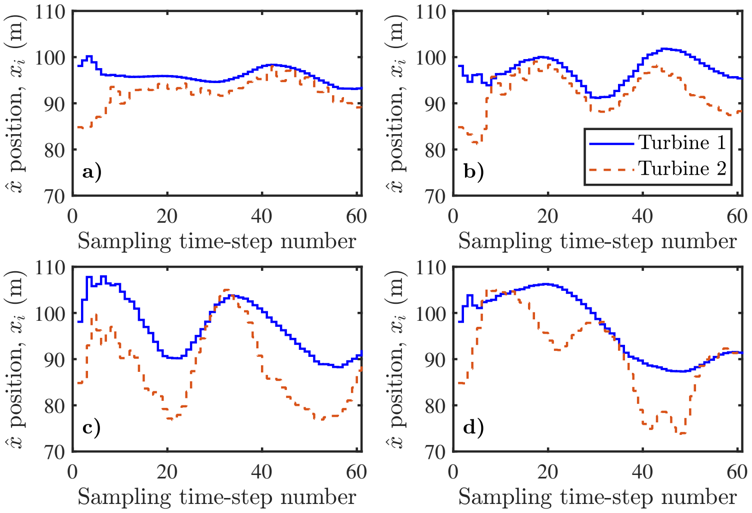


Figure 6.13: Evolution of $\hat{x}$ positions in a two-turbine floating wind farm that is optimally controlled by the DEMPC from Section 6.3. The four plots correspond to different wind velocity perturbations of a) $\sigma_\infty = 5\,\%$, b) $\sigma_\infty = 10\,\%$, c) $\sigma_\infty = 15\,\%$, and d) $\sigma_\infty = 20\,\%$.

take the form of constraints in the DEMPC optimization process or, alternatively, downwind position regulation may be achieved via wind turbine-level control systems.

6.4.4 Performance of different wind farm sizes

In this subsection, the impact of increasing the wind farm array size is assessed while maintaining $\sigma_\infty = 5\,\%$. Evolutions of wind farm power production for wind farm sizes ranging from 1×2 to 1×5 are shown in Figure 6.12 for both optimal operation using DEMPC and greedy operation. In all cases, the energy produced by the wind farm over the simulated hour increases by approximately 20 % when optimal control is used in place of greedy operation. This result contradicts the findings of our stationary optimization study [103], wherein the relative gain in wind farm efficiency increased from 21.8 to 42.7 % as the wind farm size was varied from 2×2 to 7×7.

One possible explanation for this outcome is that, as the wind farm size in-

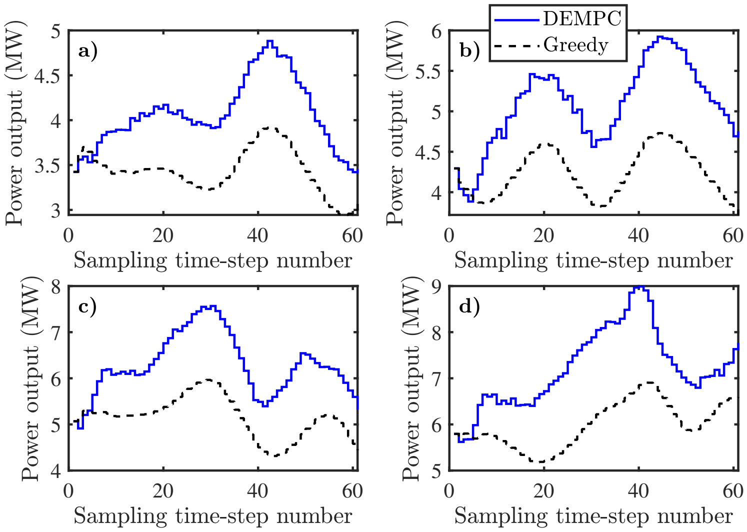


Figure 6.14: Evolution of power production in floating wind farms of different sizes that are optimally controlled by the DEMPC from Section 6.3. For comparison, power production under greedy operation is also demonstrated. The wind velocity perturbation is fixed at $\sigma_\infty = 5\,\%$. The four plots correspond to different wind farm configurations of a) 1×2, b) 1×3, c) 1×4, and d) 1×5.

creases, the trailing machines are subject to dynamic coupling from a larger number of upstream counterparts. As a result, their neural networks are subject to greater prediction error, which renders their optimal controllers less effective. This explanation is validated by plotting evolutions of turbine $\hat{y}$ positions for the different wind farm sizes in Figure 6.15. It is evident that, although σ_∞ remains unchanged, the $\hat{y}$ position of each subsequent downstream turbine displays greater variability in comparison to those of its upstream neighbors. The $\hat{y}$ position trajectory of turbine 5 in Figure 6.15d defies this trend; however, the $\hat{y}$ position trajectory of turbine 3 is consistently more volatile than those of turbines 1 and 2 in all relevant plots, and the $\hat{y}$ position trajectory of turbine 4 is consistently more volatile that that of turbine 3 in all relevant plots. This increased volatility results in longer moments of increased overlap between turbine rotors.

A final observation from Figure 6.15 that is worth point out is the fact that the social hierarchy-based DEMPC algorithm successfully prevents conflicting deci-

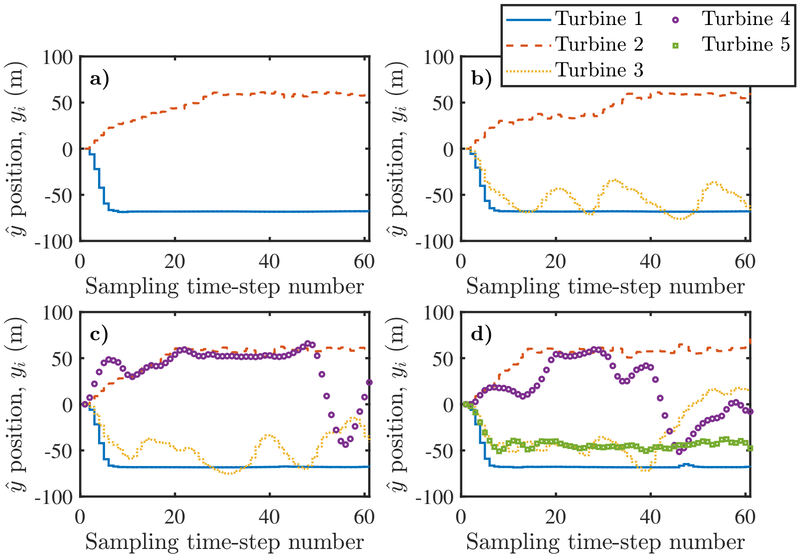


Figure 6.15: Evolution of $\hat{y}$ positions in floating wind farms of different sizes that are optimally controlled by the DEMPC from Section 6.3. The wind velocity perturbation is fixed at $\sigma_\infty = 5\,\%$. The four plots correspond to different wind farm configurations of a) 1×2, b) 1×3, c) 1×4, and d) 1×5.

sions in all simulated cases. For each wind farm size, neighboring floating wind turbines are relocated in opposite directions, which is the most effective modified layout for minimizing adjacent overlap areas.

Evolutions of turbine $\hat{x}$ positions for the different wind farm sizes are plotted in Figure 6.16. As observed from $\hat{x}$ displacements in the case of the two-turbine wind farm from Figure 6.13, downstream machines shift a smaller distance on average in the downwind direction due to reduced incident wind speeds while also experiencing greater volatility in $\hat{x}$ displacements over time due to aerodynamic coupling. However, it is observed in Figure 6.16 that, with each subsequent downstream machine, additional downwind displacements gradually diminish. For instance, over the course of the 3,600 sec simulation from Figure 6.16d, the average downwind displacement of turbine 1 is the largest at 96.1 m. This value drops to 90.9 and 86.0 m for turbines 2 and 3, and then to 82.8 m for both turbines 4 and 5. Clearly, the wind velocity deficit resulting from the compounding of overlapping wakes

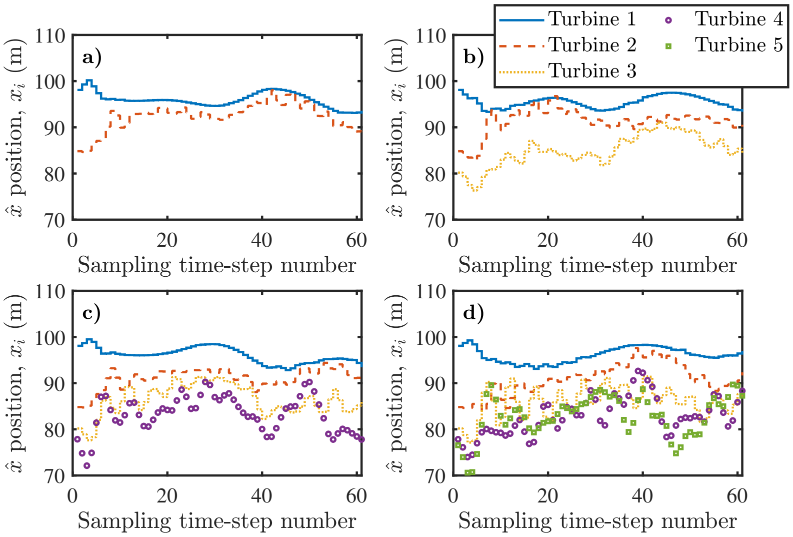


Figure 6.16: Evolution of $\hat{x}$ positions in floating wind farms of different sizes that are optimally controlled by the DEMPC from Section 6.3. The wind velocity perturbation is fixed at $\sigma_\infty = 5\,\%$. The four plots correspond to different wind farm configurations of a) 1×2, b) 1×3, c) 1×4, and d) 1×5.

ultimately converges to a finite value. This result is significant in the context of turbine repositioning. It indicates that, regardless of the size of the wind farm, turbines that are located far downstream may still encounter large enough wind speeds to generate sufficient platform displacements for rotor overlap reduction.

6.5 Conclusion

This chapter presented the first real-time implementation of wind farm control for power maximization in floating offshore wind farms. The specific mechanism for power maximization involved relocating wind turbines using aerodynamic forces to minimize overlap areas between the rotors of adjacent turbines.

Floating wind farms were simulated using FOWFSim-Dyn (see Chapter 4), which is a simulation tool designed specifically for capturing floating platform motion coupled with wind farm aerodynamics. The wind farm controller was based on a social hierarchy-based distributed model predictive control (DEMPC) algorithm (see Chapter 5) developed to address convergence challenges in non-convex distributed optimal control problems. Finally, feed-forward neural networks were used to model floating turbine dynamics during the dynamic optimization process of the DEMPC algorithm.

Conclusions and recommendations that have been derived from the results of this work are now discussed. Neural networks were effectively used to predict floating wind turbine behaviour for wind farm rows with layouts of 1×3 or smaller while neglecting dynamic coupling. Future work should establish error bounds on neural network predictions while taking into consideration the effects of dynamic coupling for improved performance in larger wind farms.

Deviations in the free stream wind velocity from the baseline value at which neural networks were tuned diminished controller performance. Future work may consider the free stream wind speed and direction as inputs to the neural networks or, alternatively, unique neural networks may be identified for multiple wind speed and direction ranges.

In larger wind farms, downstream machines demonstrated high variability in platform positions over time. This outcome resulted from increased neural network prediction error due to dynamic coupling. Once the bounds on these uncertainties have been established, future work may implement constraint tightening methods to limit the variability of floating platform positions.

The current controller maximized wind farm power production indirectly by minimizing the overlap areas between the rotors of adjacent floating turbines. Future work should use some machine learning scheme to identify power output functions for each turbine to serve as cost functions in DEMPC, thus permitting direct maximization of wind farm power production.

In the current work, a single row of wind turbines aligned with the predominant free stream wind direction was examined. Identifying which pairs of turbines encounter strong aerodynamic coupling was therefore a trivial task. In more complex wind farm configurations, such as gridded layouts, the pairings of turbines that interact aerodynamically is determined by the wind direction at any given instant in time. Future work may therefore extend the cost functions used by the DEMPC algorithm to capture the effects of wind direction on turbine pairings.

The current chapter was concerned solely with power maximization as the wind farm control objective. Future work may assess alternative objectives. For instance, at sufficiently large wind speeds, the primary concern of a wind farm operator shifts from power maximization to load reduction [107]. This shift may be accounted for via adjustments to the cost functions used in optimal control algorithms. Specifically, cost functions may be defined such that the turbines track power set-points while minimizing blade and tower vibrations among their neighborhoods. The role that YITuR may play in such a control problem should be investigated.

The optimal state and input trajectories computed using DEMPC in the current work should not be implemented directly. Instead, they should serve as set-points for turbine-level controllers that consider additional dynamic phenomena such as platform oscillation. Future work should therefore integrate wind farm-level control with lower controllers capable of rejecting disturbances caused by waves and wind gusts with the aim of minimizing turbine loads and platform oscillation.

Finally, all state information was assumed to be available in the current work. Future work should examine sensing and observer techniques for estimating floating platform positions and velocities.

Chapter 7

Conclusion

This thesis assessed the potential of real-time turbine repositioning for increasing power production in floating offshore wind farms. A thorough review of existing wind farm control literature was first presented in order to establish an understanding of the status of research as well as major gaps. This step was followed by the development of FOWFSim, which is a steady-state model for predicting platform displacement and power production in floating offshore wind farms. This model was used to conduct steady-state optimization studies to determine the potential of the wind farm control method of *yaw and induction-based turbine repositioning* (YITuR) under idealized static wind conditions. Next, a parametric dynamic model of floating offshore wind farms named FOWFSim-Dyn was developed to permit real-time control system evaluations.

The specific properties of the turbine repositioning control problems rendered existing optimal control algorithms inadequate. As a result, the development of a novel distributed economic model predictive control (DEMPC) algorithm that overcame the shortcomings of existing theory was the next necessary step. Finally, the real-time performance this DEMPC algorithm was evaluated via implementation using FOWFSim-Dyn. The following sections summarize the major findings from these research steps and provide recommendations for future work. The final subsection directly answers the research question posed in the introduction of this thesis by addressing the potential and challenges associated with the practical implementation of YITuR wind farm control.

7.1 Steady-state assessment of YITuR

The primary conclusion from the steady-state assessment of YITuR was that this wind farm control strategy has the potential to raise wind farm efficiency by more than 20 %. However, its performance and viability were determined to be strongly dependent upon wind conditions and mooring system design parameters. Future work in this area should therefore involve the development of optimization software that not only computes ideal turbine operating parameters, but also mooring system properties such as cable length and anchor locations in order to maximize annual energy production. Further, such software should consider site-specific wind velocity data.

7.2 Dynamic modeling of floating offshore wind farms

The primary purpose for developing FOWFSim-Dyn was the absence of high-fidelity simulation tools with similar capabilities and the lack of access to scaled floating wind farm experiments. Consequently, only steady-state predictions of wake propagation were validated using published experimental results from wind tunnel tests. Validation of dynamic wake transport and platform motion have not been performed. Future work in this area should therefore involve the development of high-fidelity floating wind farm simulators that utilize large eddy simulations (LES) to capture wake behaviour and couple this solver with a three-dimensional floating platform dynamics module. Such a simulation tool, along with scaled experimental techniques may be used for further validation of FOWFSim-Dyn.

7.3 Real-time YITuR controller implementation

It was demonstrated during the implementation of a YITuR controller that feed-forward neural networks may serve as appropriate surrogate models for predicting floating platform motion during the dynamic optimization process of DEMPC. However, the error in predictions was strongly influenced by aerodynamic coupling between turbines and time-varying wind conditions. As a result, future work on floating wind farm data-driven modeling should take into account several recommendations. First, coupling effects between turbines should be considered in the machine learning process. With this capability, each machine may better predict its future trajectory once informed of the optimal decisions and actions of its upstream neighbors. Alternatively, coupling may continue to be treated as a model uncertainty; however, more extensive studies should be carried out to determine the bounds on these uncertainties. Finally, the current condition of the free stream wind should factor into surrogate model outputs. This outcome may be achieved by directly including the free stream wind speed and direction as model inputs, or by tuning various neural networks for different ranges of wind velocity.

The prediction error due to aerodynamic coupling resulted in significant platform motion and deterioration of performance among downstream machines. It is therefore recommended that the optimal control problem be augmented with constraints that limit platform motion over time. Furthermore, satisfaction of these constraints should be guaranteed using constraint tightening approaches once error bounds on neural network predictions have been established.

The current implementation of YITuR control maximized wind farm power production indirectly by minimizing rotor overlap areas between adjacent floating wind turbines. Although this approach yielded gains in energy production exceeding 20 % in scenarios of low wind speed and direction variability, a more direct power maximization approach should be pursued. Namely, future research should focus on the development of surrogate models that predict the power output of a floating wind turbine based on the inputs and states of the turbine itself, along with those of its upstream neighbors. Once such a function is established, the distributed economic model predictive control theory developed in this work may be implemented without modification for the purpose of wind farm power maximization.

The objective function used in Chapter 6 may be extended in two ways. First, the pairs of turbines that interact aerodynamically may be defined as a function of the wind direction. In this manner, as the wind direction changes, any given turbine may update the set of neighboring machines with which it should minimize rotor overlap. The current work only considered a single row of wind turbines that are aligned with the free stream wind. The modification to objective functions mentioned above will generalize YITuR to any wind farm configuration. Second, the cost function used in DEMPC may be extended to accommodate the alternative wind farm control objective of power set-point tracking and load reduction. This mode of wind farm control is relevant in cases where the wind speed is excessively high and power maximization is no longer of primary concern.

The YITuR controller developed in this work is high-level and thus solely considers floating turbine states and inputs that are consequential at the scale of the wind farm. Motions such as platform oscillation and blade vibration are not factored into the optimal control problem since their effects on wind farm power production are deemed to be insignificant. Nonetheless, platform oscillation and blade vibration are consequential in the context of individual turbine loads and fatigue. As a result, future work should develop turbine-level controllers that track the optimal trajectories computed by the higher-level YITuR controller while minimizing platform oscillation and blade vibration.

Finally, it was assumed throughout this thesis that floating wind turbine state information, such as platform positions and velocities, is directly measurable. Physical implementation of a YITuR controller will require appropriate sensing and state estimation techniques that reliably deliver this information. Future work should therefore assess the potential of different measurement and estimation technologies using scaled floating wind farm experiments.

7.4 Final comments on YITuR

Although the analysis performed in this work indicates that YITuR has the potential to raise the efficiency of floating offshore wind farms by 20 %, wind farm developers may be hesitant to implement such a control technique in response to several challenges. First, due to the large range of motion of the floating platforms, gaining access to wind turbines will be a potentially unsafe and involved task for maintenance crews. Second, the large range of platform motion will render the ocean surface area spanned by the wind farm unsafe for naval vessel travel. Third, lengthening mooring lines to expand movable ranges induces additional capital expenses and alters the oscillatory responses of the floating platforms in the presence of waves, wind velocity fluctuations, and aerodynamic coupling effects between turbines. Finally, raising wind farm power production by reducing wake overlap may exacerbate fatigue damage in turbines with partially overlapped rotors.

Further investigation of these challenges will illuminate the practicality and feasibility of YITuR operation. If the complications associated with large platform displacements are deemed to be a valid trade-off for the added revenue stemming from increased efficiency, then a case could potentially be presented for YITuR wind farm control. Regardless of this outcome however, completion of the current project has been a worthwhile research endeavor. It yielded the first steady-state and dynamic models of floating offshore wind farms, which will promote further development of floating offshore wind technology via design, optimization, and controller assessment studies. Further, tackling the repositioning problem necessitated the development of a novel distributed control system that will have implications in a wide range of multi-agent systems such as autonomous vehicle platoons, traffic networks, and multi-robot systems.

Bibliography

[1] M. Abbes and M. Allagui. Centralized control strategy for energy maximization of large array wind turbines. *Sustainable Cities and Society*, 25:82–89, 2016. ISSN 22106707. doi:10.1016/j.scs.2015.11.007. → pages 37, 237

[2] M. S. Adaramola and P. Å. Krogstad. Experimental investigation of wake effects on wind turbine performance. *Renewable Energy*, 36(8): 2078–2086, 2011. ISSN 09601481. doi:10.1016/j.renene.2011.01.024. → pages 36, 46, 69, 236, 242

[3] M. A. Ahmad, S. Azuma, and T. Sugie. A model-free approach for maximizing power production of wind farm using multi-resolution Simultaneous Perturbation Stochastic Approximation. *Energies*, 7: 5624–5646, 2014. ISSN 19961073. doi:10.3390/en7095624. → pages 39, 59, 64, 69, 105, 238

[4] J. Ainslie. Calculating the flowfield in the wake of wind turbines. *Journal of Wind Engineering and Industrial Aerodynamics*, 24:213–224, 1988. → pages 31, 231

[5] W. Al-Gherwi, H. Budman, and A. Elkamel. Robust distributed model predictive control: A review and recent developments. *Canadian Journal of Chemical Engineering*, 89(5):1176–1190, 2011. ISSN 00084034. doi:10.1002/cjce.20555. → page 139

[6] F. Albalawi, H. Durand, and P. D. Christofides. Distributed economic model predictive control with Safeness-Index based constraints for nonlinear systems. *Systems and Control Letters*, 110:21–28, 2017. ISSN 01676911. doi:10.1016/j.sysconle.2017.10.002. → pages 140, 178

[7] D. Allaerts and J. Meyers. Large eddy simulation of a large wind-turbine array in a conventionally neutral atmospheric boundary layer. *Physics of Fluids*, 27(6), 2015. ISSN 10897666. doi:10.1063/1.4922339. → page 30

[8] O. Anaya-Lara, J. O. Tande, K. Uhlen, and K. Merz. *Offshore Wind Energy Technology*. John Wiley & Sons Ltd, 2018. → page 10

[9] J. Annoni, P. M. O. Gebraad, A. K. Scholbrock, P. A. Fleming, and J.-W. van Wingerden. Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model. *Wind Energy*, 19(6): 1135–1150, 2016. doi:10.1002/we.1891. → pages 33, 34, 35, 36, 66, 69, 235

[10] J. Annoni, C. Bay, K. Johnson, E. Dall'Anese, E. Quon, T. Kemper, and P. Fleming. A Framework for Autonomous Wind Farms: Wind Direction Consensus. *Accepted for review in Wind Energy Science Discussions*, n/a, 2018. doi:10.5194/wes-2018-60. → pages 20, 22, 65

[11] J. Annoni, C. Bay, T. Taylor, L. Pao, P. Fleming, and K. Johnson. Efficient optimization of large wind farms for real-time control. In *Proceedings of the 2018 American Control Conference*, pages 6200–6205, 2018. doi:10.23919/ACC.2018.8430751. → page 241

[12] E. F. Asadi and A. Richards. Scalable distributed model predictive control for constrained systems. *Automatica*, 93:407–414, 2018. ISSN 00051098. doi:10.1016/j.automatica.2018.03.050. → pages 141, 177

[13] D. Aßmann, U. Laumanns, and D. Uh, editors. *Renewable Energy: A Gobal Review of Technologies, Policies and Markets*. Earthscan, London, UK, 2006. → page 8

[14] J. J. Barradas-Berglind and R. Wisniewski. Wind farm axial-induction factor optimization for power maximization and load alleviation. In *Proceedings of the 2016 European Control Conference*, pages 891–896, 2016. doi:10.1109/ECC.2016.7810402. → page 233

[15] J. Barreiro-Gomez, C. Ocampo-Martinez, F. Bianchi, and N. Quijano. Model-free control for wind farms using a gradient estimation-based algorithm. In *Proceedings of the 2015 European Control Conference*, pages 1516–1521, 2015. doi:10.1109/ECC.2015.7330753. → page 237

[16] J. Bartl and L. Sætran. Experimental testing of axial induction based control strategies for wake control and wind farm optimization. *Journal of Physics: Conference Series*, 753, 2016. doi:10.1088/1742-6596/753/3/032035. → pages 36, 236

[17] M. Bastankhah and F. Porté-Agel. A new analytical model for wind-turbine wakes. *Renewable Energy*, 70:116–123, 2014. doi:10.1016/j.renene.2014.01.002. → pages 27, 75

[18] M. Bastankhah and F. Porté-Agel. Experimental and theoretical study of wind turbine wakes in yawed conditions. *Journal of Fluid Mechanics*, 806: 506–541, 2016. ISSN 14697645. doi:10.1017/jfm.2016.595. → pages xx, 119, 120, 124, 129, 130, 181

[19] M. Bastankhah and F. Porté-Agel. Wind farm power optimization via yaw angle control: A wind tunnel study. *Journal of Renewable and Sustainable Energy*, 11(2), 2019. ISSN 19417012. doi:10.1063/1.5077038. → pages 47, 243

[20] A. Behnood, H. Gharavi, B. Vahidi, and G. H. Riahy. Optimal output power of not properly designed wind farms, considering wake effects. *International Journal of Electrical Power and Energy Systems*, 63:44–50, 2014. ISSN 01420615. doi:10.1016/j.ijepes.2014.05.052. → page 233

[21] E. Bitar and P. Seiler. Coordinated control of a wind turbine array for power maximization. In *Proceedings of the 2013 American Control Conference*, pages 2898–2904, 2013. doi:10.1109/ACC.2013.6580274. → page 233

[22] G. Bo, L. Yongqian, Y. Jie, L. Li, and K. Shun. A wind farm optimal control algorithm based on wake fast-calculation model. *Journal of Solar Energy Engineering*, 138(2):024501, 2016. ISSN 0199-6231. doi:10.1115/1.4032297. → page 234

[23] S. Boersma, P. M. Gebraad, M. Vali, B. M. Doekemeijer, and J. W. Van Wingerden. A control-oriented dynamic wind farm flow model: WFSim. *Journal of Physics: Conference Series*, 753(3), 2016. ISSN 17426596. doi:10.1088/1742-6596/753/3/032005. → pages 30, 31, 231

[24] S. Boersma, B. Doekemeijer, P. Gebraad, P. Fleming, J. Annoni, A. Scholbrock, J. Frederik, and J.-W. van Wingerden. A tutorial on control-oriented modeling and control of wind farms. *Proceedings of the American Control Conference*, pages 1–18, 2017. doi:10.23919/ACC.2017.7962923. → pages 10, 12, 13, 17, 21, 24, 25, 29, 69, 105

[25] S. Boersma, B. Doekemeijer, M. Vali, J. Meyers, and J.-W. van Wingerden. A control-oriented dynamic wind farm model: WFSim. *Wind Energy*

Science, 3(1):75–95, 2018. doi:10.5194/wes-3-75-2018. → pages 30, 31, 106, 231

[26] K. Boorsma. Heat and Flux - Analysis of Field Measurements. Technical report, Energy Research Center of the Netherlands, 2012. → pages 36, 43, 69, 236

[27] E. Bossanyi. Combining induction control and wake steering for wind farm energy and fatigue loads optimisation. *Journal of Physics: Conference Series*, 1037(3), 2018. ISSN 17426596. doi:10.1088/1742-6596/1037/3/032011. → pages 244, 245

[28] C. L. Bottasso, S. Cacciola, and J. Schreiber. Local wind speed estimation, with application to wake impingement detection. *Renewable Energy*, 116: 155–168, 2018. doi:10.1016/j.renene.2017.09.044. → page 22

[29] T. Burton, N. Jenkins, D. Sharpe, and E. Bossanyi. *Wind Energy Handbook*. John Wiley & Sons, Ltd, 2nd edition, 2011. doi:10.1002/9781119992714. → pages 16, 18, 68, 75, 112, 113, 119, 180

[30] R. L. Busby. *Wind Power: The Industry Grows Up*. PennWell Corporation, Tulsa, Oklahoma, 2012. → pages 2, 10, 11, 12, 17, 68

[31] F. Campagnolo, V. Petrovi, C. L. Bottasso, and A. Croce. Wind tunnel testing of wake control strategies. In *Proceedings of the 2016 American Control Conference*, pages 513–518, 2016. doi:10.1109/ACC.2016.7524965. → pages 36, 46, 69, 236, 243

[32] F. Campagnolo, V. Petrović, J. Schreiber, E. M. Nanos, A. Croce, and C. L. Bottasso. Wind tunnel testing of a closed-loop wake deflection controller for wind farm power maximization. *Journal of Physics: Conference Series*, 753, 2016. doi:10.1088/1742-6596/753/3/032006. → pages 50, 60, 62, 69, 245

[33] F. Campagnolo, J. Schreiber, A. M. Garcia, and C. L. Bottasso. Wind Tunnel Validation of a Wind Observer for Wind Farm Control. In *Proceedings of the 2017 International Ocean and Polar Engineering Conference*, pages 640–647, 2017. → page 22

[34] Carbon Trust staff. Holistic offshore wind farm control strategies to be put to the test, 2017. URL https://www.carbontrust.com/news/2017/12/wind-farm-control-trials/. → pages 13, 68

[35] G. Chartrand and P. Zhang. *Chromatic Graph Theory*. CRC Press, Taylor and Francis Group, New York, NY, 2008. ISBN 978-1-58488-801-7. → page 163

[36] X. Chen, M. Heidarinejad, J. Liu, and P. D. Christofides. Distributed economic MPC: Application to a nonlinear chemical process network. *Journal of Process Control*, 22(4):689–699, 2012. ISSN 09591524. doi:10.1016/j.jprocont.2012.01.016. → pages 140, 178

[37] L. Chisci, J. A. Rossiter, and G. Zappa. Systems with persistent disturbances: Predictive control with restricted constraints. *Automatica*, 37 (7):1019–1028, 2001. ISSN 00051098. doi:10.1016/S0005-1098(01)00051-6. → page 190

[38] P. D. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers and Chemical Engineering*, 51:21–41, 2013. ISSN 00981354. doi:10.1016/j.compchemeng.2012.05.011. → page 139

[39] M. Churchfield, P. Fleming, B. Bulder, and S. White. Wind Turbine Wake-Redirection Control at the Fishermen's Atlantic City Windfarm. Technical report, National Renewable Energy Laboratory, 2015. → pages 45, 69, 242

[40] M. J. Churchfield, S. Lee, J. Michalakes, and P. J. Moriarty. A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics. *Journal of Turbulence*, 13(14):1–32, 2012. ISSN 1468-5248. doi:10.1080/14685248.2012.668191. → pages 29, 44, 52

[41] U. Ciri, M. A. Rotea, and S. Leonardi. Model-free control of wind farms: A comparative study between individual and coordinated extremum seeking. *Renewable Energy*, 113:1033–1045, 2017. doi:10.1016/j.renene.2017.06.065. → pages 41, 240

[42] T. A. Clevenhult and F. Himmelman. *Added Turbulence and Optimal Power Distribution in Large Off-shore Wind Farms*. MASc Thesis. Lund University, 2010. → page 234

[43] G. Corten and P. Schaak. Heat and Flux - Increase of Wind Farm Production by Reduction of the Axial Induction. Technical report, Energy Research Center of the Netherlands, 2003. → pages 35, 236

[44] A. J. Coulling, A. J. Goupee, A. N. Robertson, J. M. Jonkman, and H. J. Dagher. Validation of a FAST semi-submersible floating wind turbine numerical model with DeepCwind test data. *Journal of Renewable and Sustainable Energy*, 5(2), 2013. ISSN 19417012. doi:10.1063/1.4796197. → page 126

[45] A. Crespo, J. Hernandez, E. Fraga, and C. Andreu. Experimental validation of the UPM computer code to calculate wind turbine wakes and comparison with other models. *Journal of Wind Engineering and Industrial Aerodynamics*, 27(1-3):77–88, 1988. ISSN 01676105. doi:10.1016/0167-6105(88)90025-6. → pages 31, 231

[46] J. Dahlberg and D. Medici. Potential improvement of wind turbine array efficiency by active wake control (AWC). In *Proceedings of the 2003 European Wind Energy Conference*, 2003. → page 241

[47] M. De-Prada-Gil, C. G. Alías, O. Gomis-Bellmunt, and A. Sumper. Maximum wind power plant generation by reducing the wake effect. *Energy Conversion and Management*, 101:73–84, 2015. ISSN 01968904. doi:10.1016/j.enconman.2015.05.035. → pages 69, 233

[48] D. Dilip and F. Porté-Agel. Wind turbine wake mitigation through bade pitch offset. *Energies*, 10(757), 2017. doi:10.3390/en10060757. → pages 33, 69, 235

[49] B. M. Doekemeijer, S. Boersma, L. Y. Pao, T. Knudsen, and J.-W. van Wingerden. Online model calibration for a simplified LES model in pursuit of real-time closed-loop wind farm control. *Wind Energy Science*, 3(2): 749–765, 2018. doi:10.5194/wes-3-749-2018. → pages 20, 23, 65

[50] P. A. Driessen, R. M. Hermans, and P. P. Van Den Bosch. Distributed economic model predictive control of networks in competitive environments. In *Proceedings of the IEEE Conference on Decision and Control*, pages 266–271, 2012. ISBN 9781467320665. doi:10.1109/CDC.2012.6426492. → pages 140, 178

[51] X. Du, U. Xi, and S. Li. Distributed model predictive control for large-scale systems. In *Proceedings of the American Control Conference*, pages 3142–3143, 2001. → page 150

[52] T. Duc, O. Coupiac, N. Girard, G. Giebel, and T. Göçmen. Local turbulence parameterization improves the Jensen wake model and its implementation for power optimization of an operating wind farm.

Accepted for review in Wind Energy Science Discussions, n/a, 2018. doi:10.5194/wes-2018-62. → page 234

[53] W. B. Dunbar and R. M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, 2006. ISSN 00051098. doi:10.1016/j.automatica.2005.12.008. → pages 150, 172

[54] J. Eilbrecht and O. Stursberg. Hierarchical solution of non-convex optimal control problems with application to autonomous driving. *European Journal of Control*, 50:188–197, 2019. ISSN 09473580. doi:10.1016/j.ejcon.2019.05.006. → page 140

[55] M. Ellis, H. Durand, and P. D. Christofides. A tutorial review of economic model predictive control methods. *Journal of Process Control*, 24(8): 1156–1178, 2014. ISSN 09591524. doi:10.1016/j.jprocont.2014.03.010. → pages 139, 177, 178

[56] G. Emanuel. *Analytical Fluid Dynamics*. Taylor & Francis Group, Boca Raton, Florida, 3rd edition, 2016. → pages 25, 29

[57] Environmental Protection Agency staff. Avoided emissions and generation tool (AVERT), 2018. URL https://www.epa.gov/statelocalenergy/avoided-emissions-and-generation-tool-avert. → page 8

[58] P. Fleming, P. M. Gebraad, S. Lee, J.-W. van Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty. Simulation comparison of wake mitigation control strategies for a two-turbine case. *Wind Energy*, 18(12):2135–2143, 2015. doi:10.1002/we.1810. → pages 16, 17, 18, 44, 45, 52, 65, 69, 242, 247

[59] P. Fleming, J. Annoni, J. J. Shah, L. Wang, S. Ananthan, Z. Zhang, K. Hutchings, P. Wang, W. Chen, and L. Chen. Field test of wake steering at an offshore wind farm. *Wind Energy Science*, 2:229–239, 2017. doi:10.5194/wes-2017-4. → pages 51, 52, 60, 69, 85, 104, 246

[60] P. Fleming, J. Annoni, M. Churchfield, L. A. Martinez-Tossas, K. Gruchalla, M. Lawson, and P. Moriarty. A simulation study demonstrating the importance of large-scale trailing vortices in wake steering. *Wind Energy Science*, 3(1):243–255, 2018. doi:10.5194/wes-3-243-2018. → pages 16, 31, 45, 57, 66, 69, 123, 137, 242

[61] P. Fleming, J. King, K. Dykes, E. Simley, J. Roadman, A. Scholbrock, P. Murphy, J. K. Lundquist, P. Moriarty, K. Fleming, J. V. Dam, C. Bay, R. Mudafort, H. Lopez, J. Skopek, M. Scott, B. Ryan, and D. Brake. Initial Results From a Field Campaign of Wake Steering Applied at a Commercial Wind Farm : Part 1. *Accepted for review in Wind Energy Science Discussions*, n/a, 2019. doi:10.5194/wes-2019-5. → pages 51, 246

[62] P. A. Fleming, P. M. Gebraad, S. Lee, J. W. van Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty. Evaluating techniques for redirecting turbine wakes using SOWFA. *Renewable Energy*, 70:211–218, 2014. ISSN 09601481. doi:10.1016/j.renene.2014.02.015. → pages 18, 242, 247

[63] P. A. Fleming, A. Ning, P. M. O. Gebraad, and K. Dykes. Wind plant system engineering through optimization of layout and yaw control. *Wind Energy*, 19(2):329–344, feb 2016. doi:10.1002/we.1836. → page 241

[64] S. T. Frandsen, H. E. Jørgensen, O. Rathmann, J. Badger, S. Ott, P. E. Rethore, S. E. Larsen, R. Barthelmie, K. Hansen, and L. E. Jensen. The making of a second-generation wind farm efficiency model complex. *Wind Energy*, 12(5):445–458, 2009. ISSN 10954244. doi:10.1002/we.351. → page 42

[65] P. Gebraad, J. J. Thomas, A. Ning, P. Fleming, and K. Dykes. Maximization of the annual energy production of wind power plants by optimization of layout and yaw-based wake control. *Wind Energy*, 20(1): 97–107, jan 2017. doi:10.1002/we.1993. → page 242

[66] P. M. Gebraad. *Data-driven Wind Plant Control*. PhD Thesis. Delft University of Technology, 2014. → pages 68, 238, 242, 245, 247

[67] P. M. Gebraad and J. W. Van Wingerden. A control-oriented dynamic model for wakes in wind plants. *Journal of Physics: Conference Series*, 524(1), 2014. ISSN 17426596. doi:10.1088/1742-6596/524/1/012186. → pages 27, 29, 105, 231

[68] P. M. Gebraad, F. W. Teeuwisse, J. W. Van Wingerden, P. A. Fleming, S. D. Ruben, J. R. Marden, and L. Y. Pao. A data-driven model for wind plant power optimization by yaw control. In *Proceedings of the 2014 American Control Conference*, pages 3128–3134, 2014. doi:10.1109/ACC.2014.6859118. → page 245

[69] P. M. Gebraad, P. A. Fleming, and J. W. Van Wingerden. Wind turbine wake estimation and control using FLORIDyn, a control-oriented dynamic wind plant model. *Proceedings of the American Control Conference*, pages 1702–1708, 2015. doi:10.1109/ACC.2015.7170978. → pages 20, 23, 49, 61, 65, 104, 244

[70] P. M. O. Gebraad and J. W. van Wingerden. Maximum power-point tracking control for wind farms. *Wind Energy*, 18(3):429–447, mar 2015. doi:10.1002/we.1706. → pages 23, 39, 50, 59, 62, 104, 105, 238

[71] P. M. O. Gebraad, F. C. van Dam, and J. W. van Wingerden. A model-free distributed approach for wind plant control. In *Proceedings of the 2013 American Control Conference*, pages 628–633, 2013. doi:10.1109/ACC.2013.6579907. → page 238

[72] P. M. O. Gebraad, F. W. Teeuwisse, J. W. van Wingerden, P. A. Fleming, S. D. Ruben, J. R. Marden, and L. Y. Pao. Wind plant power optimization through yaw control using a parametric model for wake effects - A CFD simulation study. *Wind Energy*, 19(1):95–114, 2016. doi:10.1002/we.1822. → pages 20, 22, 26, 27, 31, 44, 45, 48, 60, 61, 69, 72, 75, 77, 104, 105, 123, 127, 137, 231, 245

[73] N. Gionfra, H. Siguerdidjane, G. Sandou, and D. Faille. Hierarchical control of a wind farm for wake interaction minimization. *International Federation of Automatic Control*, 49(27):330–335, 2016. doi:10.1016/j.ifacol.2016.10.713. → pages 37, 237

[74] T. Göçmen, G. Giebel, P. E. Sørensen, and N. K. Poulsen. *Possible Power Estimation of Down-Regulated Offshore Wind Power Plants*. PhD Thesis. Technical University of Denmark, 2016. ISBN 9788793278530. → page 239

[75] T. Göçmen, P. V. D. Laan, P. E. Réthoré, A. P. Diaz, G. C. Larsen, and S. Ott. Wind turbine wake models developed at the technical university of Denmark: A review. *Renewable and Sustainable Energy Reviews*, 60: 752–769, 2016. ISSN 18790690. doi:10.1016/j.rser.2016.01.113. → pages 25, 105

[76] J. P. Goit and J. Meyers. Optimal control of energy extraction in wind-farm boundary layers. *Journal of Fluid Mechanics*, 768:5–50, 2015. ISSN 14697645. doi:10.1017/jfm.2015.70. → pages 42, 55, 240

[77] J. P. Goit, W. Munters, and J. Meyers. Optimal coordinated control of power extraction in LES of a wind farm with entrance effects. *Energies*, 9 (29), 2016. ISSN 19961073. doi:10.3390/en9010029. → pages 21, 42, 240

[78] M. Hall and A. Goupee. Validation of a lumped-mass mooring line model with DeepCwind semisubmersible model test data. *Ocean Engineering*, 104:590–603, 2015. doi:10.1016/j.oceaneng.2015.05.035. → page 122

[79] M. Hall, B. Buckham, and C. Crawford. Evaluating the importance of mooring line model fidelity in floating offshore wind turbine simulations. *Wind Energy*, 17(12):1835–1853, 2014. doi:10.1002/we.1669. → pages xx, 122, 126, 128, 129

[80] C. Han. *Position Control of a Floating Offshore Wind Turbine System Using Aerodynamic Force*. MASc Thesis. The University of British Columbia, 2017. → page 17

[81] C. Han, J. R. Homer, and R. Nagamune. Movable range and position control of an offshore wind turbine with a semi-submersible floating platform. In *Proceedings of the 2017 American Control Conference*, pages 1389–1394, 2017. doi:10.23919/ACC.2017.7963147. → pages 17, 53, 69, 70

[82] E. Hau. *Wind Turbines: Fundamentals, Technologies, Application, Economics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 3rd ed edition, 2013. ISBN 978-3-642-27150-2. doi:10.1007/978-3-642-27151-9. → page 22

[83] F. Heer, P. M. Esfahani, M. Kamgarpour, and J. Lygeros. Model based power optimisation of wind farms. In *Proceedings of the 2014 European Control Conference*, pages 1145–1150, 2014. doi:10.1109/ECC.2014.6862365. → pages 37, 239

[84] J. Herp, U. V. Poulsen, and M. Greiner. Wind farm power optimization including flow variability. *Renewable Energy*, 81:173–181, 2015. ISSN 18790682. doi:10.1016/j.renene.2015.03.034. → page 233

[85] T. Horvat, V. Spudic, and M. Baotic. Quasi-stationary optimal control for wind farm with closely spaced turbines. *Proceedings of the 2012 International Convention on Information and Communication Technology, Electronics, and Microelectronics*, pages 829–834, 2012. → page 234

[86] M. Howland, S. K. Lele, and J. O. Dabiri. Wind farm power optimization through wake steering. *ArXiv : Submit / 2588307 [Physics . Flu-Dyn]*, n/a, 2019. doi:10.13140/RG.2.2.16116.40322. → pages 48, 57, 66, 243

[87] International Renewable Energy Agency staff. Statistics time series, 2018. URL http://resourceirena.irena.org/gateway/dashboard/?topic=4{&}subTopic=16. → pages xv, 8, 9

[88] International Renewable Energy Agency staff. Global trends in renewable energy costs, 2018. URL http://resourceirena.irena.org/gateway/dashboard/?topic=3{&}subTopic=1066. → pages xv, 10, 11

[89] International Renewable Energy Agency staff. Employment time series, 2018. URL http://resourceirena.irena.org/gateway/dashboard/?topic=7{&}subTopic=53. → page 8

[90] International Renewable Energy Agency staff. Investment trends, 2018. URL http://resourceirena.irena.org/gateway/dashboard/?topic=6{&}subTopic=11. → pages xv, 9, 10

[91] N. O. Jensen. A Note on Wind Generator Interaction. Technical report, Risø National Laboratory. Report number: Risø-M-2411, 1983. → pages 25, 75, 104

[92] D. Jia and B. H. Krogh. Dsitributed model predictive control. In *Proceedings of the American Control Conference*, pages 2767–2772, 2001. → page 150

[93] Á. Jiménez, A. Crespo, and E. Migoya. Application of a LES technique to characterize the wake deflection of a wind turbine in yaw. *Wind Energy*, 13 (6):559–572, 2009. doi:10.1002/we.380. → pages 16, 26, 69, 75, 119, 181

[94] K. E. Johnson and G. Fritsch. Assessment of extremum seeking control for wind farm energy production. *Wind Engineering*, 36(6):701–715, 2012. ISSN 0309-524X. doi:10.1260/0309-524X.36.6.701. → pages 40, 59, 104, 105, 238

[95] K. E. Johnson and N. Thomas. Wind farm control: Addressing the aerodynamic interaction among wind turbines. *Proceedings of the*

American Control Conference, pages 2104–2109, 2009. doi:10.1109/ACC.2009.5160152. → pages 16, 69, 105, 238

[96] J. Jonkman, S. Butterfield, W. Musial, and G. Scott. Definition of a 5-MW Reference Wind Turbine for Offshore System Development. Technical report, National Renewable Energy Laboratory. Report number: NREL/TP-500-38060, 2009. → pages xiii, xx, xxii, 85, 86, 126, 127, 131, 185, 186

[97] J. M. Jonkman. Dynamics modeling and loads analysis of an offshore floating wind turbine. Technical report, National Renewable Energy Laboratory, 2007. → page 80

[98] I. Katić, J. Højstrup, and N. Jensen. A simple model for cluster efficiency. *Proceedings of the European Wind Energy Association Conference and Exhibition*, pages 407–410, 1986. → pages 20, 26, 70, 75, 104, 120, 181, 231

[99] J. Kazda, M. Zendehbad, S. Jafari, N. Chokani, and R. S. Abhari. Mitigating adverse wake effects in a wind farm using non-optimum operational conditions. *Journal of Wind Engineering and Industrial Aerodynamics*, 154:76–83, 2016. ISSN 0167-6105. doi:10.1016/j.jweia.2016.04.004. → pages 35, 43, 56, 235

[100] J. Kazda, M. Mirzaei, and N. A. Cutululis. On the architecture of wind turbine control required for induction-based optimal wind farm control. In *Proceedings of the 2018 American Control Conference*, pages 3074–3079, 2018. → pages 37, 239

[101] A. C. Kheirabadi and R. Nagamune. Modeling and power optimization of floating offshore wind farms with yaw and induction-based turbine repositioning. *Proceedings of the American Control Conference*, pages 5458–5463, 2019. → pages 29, 53, 70, 247

[102] A. C. Kheirabadi and R. Nagamune. A quantitative review of wind farm control with the objective of wind farm power maximization. *Journal of Wind Engineering & Industrial Aerodynamics*, 192(May):45–73, 2019. doi:10.1016/j.jweia.2019.06.015. → pages 4, 104, 105, 175

[103] A. C. Kheirabadi and R. Nagamune. Real-time relocation of floating offshore wind turbine platforms for wind farm efficiency maximization: An assessment of feasibility and steady-state potential. *Ocean Engineering*,

208(May), 2020. ISSN 00298018. doi:10.1016/j.oceaneng.2020.107445. → pages 5, 107, 135, 176, 180, 197

[104] H. Kim, K. Kim, I. Paek, and N. Yoo. Development of a time-domain simulation tool for offshore wind farms. *Journal of Power Electronics*, 15 (4):1047–1053, 2015. ISSN 15982092. doi:10.6113/JPE.2015.15.4.1047. → page 239

[105] H. Kim, K. Kim, and I. Paek. Power regulation of upstream wind turbines for power increase in a wind farm. *International Journal of Precision Engineering and Manufacturing*, 17(5):665–670, 2016. ISSN 20054602. doi:10.1007/s12541-016-0081-1. → page 234

[106] H. Kim, K. Kim, and I. Paek. Model based open-loop wind farm control using active power for power increase and load reduction. *Applied Sciences*, 7(10):1068, 2017. ISSN 2076-3417. doi:10.3390/app7101068. → pages 37, 59, 239

[107] T. Knudsen, T. Bak, and M. Svenstrup. Survey of wind farm control-power and fatigue optimization. *Wind Energy*, 18(8):1333–1351, 2015. doi:10.1002/we.1760. → pages 13, 104, 202

[108] P. N. Köhler, M. A. Müller, and F. Allgöwer. A distributed economic MPC framework for cooperative control under conflicting objectives. *Automatica*, 96:368–379, 2018. ISSN 00051098. doi:10.1016/j.automatica.2018.07.001. → pages 140, 142, 178, 189

[109] Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How. Distributed robust receding horizon control for multivehicle guidance. *IEEE Transactions on Control Systems Technology*, 15(4):627–641, 2007. ISSN 10636536. doi:10.1109/TCST.2007.899152. → pages 140, 141, 177, 193

[110] G. C. Larsen, H. A. Madsen, K. Thomsen, and T. J. Larsen. Wake meandering: A pragmatic approach. *Wind Energy*, 11(4):377–395, 2008. ISSN 10954244. doi:10.1002/we.267. → page 42

[111] J. Lee and D. Angeli. Cooperative distributed model predictive control for linear plants subject to convex economic objectives. In *Proceedings of the IEEE Conference on Decision and Control*, number 6, pages 3434–3439. IEEE, 2011. ISBN 9781612848006. doi:10.1109/CDC.2011.6160694. → pages 140, 178

[112] J. Lee and D. Angeli. Distributed cooperative nonlinear economic MPC. In *Proceedings of the 20th International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, 2012. → pages 140, 178

[113] J. Lee, E. Son, B. Hwang, and S. Lee. Blade pitch angle control for aerodynamic performance optimization of a wind farm. *Renewable Energy*, 54:124–130, 2013. ISSN 09601481. doi:10.1016/j.renene.2012.08.048. → page 234

[114] Y. Li and Z. Wu. Stabilization of floating offshore wind turbines by artificial muscle based active mooring line force control. In *Proceedings of the 2016 American Control Conference*, pages 2277–2282, 2016. doi:10.1109/ACC.2016.7525257. → page 17

[115] P. B. S. Lissaman. Energy effectiveness of arbitrary arrays of wind turbines. *Journal of Energy*, 3(6):323–328, 1979. → pages 10, 11, 25, 34, 49, 68, 97

[116] J. Liu, X. Chen, D. M. De La Peña, and P. D. Christofides. Iterative distributed model predictive control of nonlinear systems: Handling asynchronous, delayed measurements. *IEEE Transactions on Automatic Control*, 57(2):528–534, 2012. ISSN 00189286. doi:10.1109/TAC.2011.2164729. → pages 140, 177, 188

[117] P. Liu, A. Kurt, and U. Ozguner. Distributed Model Predictive Control for Cooperative and Flexible Vehicle Platooning. *IEEE Transactions on Control Systems Technology*, 27(3):1115–1128, 2019. ISSN 1558-0865. doi:10.1109/TCST.2018.2808911. → pages 141, 177

[118] L. Machielse, S. Barth, E. Bot, H. Hendriks, and G. Schepers. Evaluation of "Heat and Flux" Farm Control. Technical report, Energy Research Centre of the Netherlands, 2007. → pages 36, 236

[119] D. Madjidian and A. Rantzer. A stationary turbine interaction model for control of wind farms. In *Proceedings of the 18th World Congress, The International Federation of Automatic Control*, pages 4921–4926, 2011. doi:10.3182/20110828-6-IT-1002.00267. → page 234

[120] J. M. Maestre, D. Muñoz De La Peña, E. F. Camacho, and T. Alamo. Distributed model predictive control based on agent negotiation. *Journal of Process Control*, 21(5):685–697, 2011. doi:10.1016/j.jprocont.2010.12.006. → pages 141, 177

[121] J. Manwell, J. McGowen, and A. Rogers. *Wind Energy Explained: Theory, Design, and Application*. John Wiley & Sons, Ltd, 2nd edition, 2009. → pages 9, 11, 25, 75, 77, 111

[122] J. R. Marden, S. D. Ruben, and L. Y. Pao. A model-free approach to wind farm control using game theoretic methods. *IEEE Transactions on Control Systems Technology*, 21(4):1207–1214, 2013. ISSN 10636536. doi:10.1109/TCST.2013.2257780. → pages 21, 23, 38, 39, 48, 61, 62, 69, 237

[123] L. A. Martínez-Tossas, M. J. Churchfield, and S. Leonardi. Large eddy simulations of the flow past wind turbines: actuator line and disk modeling. *Wind Energy*, 18(6):1047–1060, jun 2015. ISSN 10954244. doi:10.1002/we.1747. → page 30

[124] L. A. Martínez-Tossas, J. Annoni, P. A. Fleming, and M. J. Churchfield. The aerodynamics of the curled wake: a simplified model in view of flow control. *Wind Energy Science*, 4(1):127–138, 2019. doi:10.5194/wes-4-127-2019. → pages 27, 75

[125] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. doi:10.1016/S0005-1098(99)00214-9. → pages 139, 177

[126] D. Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, 2014. doi:10.1016/j.automatica.2014.10.128. → pages 139, 177

[127] S. McTavish, D. Feszty, and F. Nitzsche. Evaluating Reynolds number effects in small-scale wind turbine experiments. *Journal of Wind Engineering and Industrial Aerodynamics*, 120:81–90, 2013. ISSN 01676105. doi:10.1016/j.jweia.2013.07.006. → page 35

[128] D. Medici. *Experimental Studies of Wind Turbine Wakes - Power Optimisation and Meandering*. PhD Thesis. KTH Royal Institute of Technology, 2005. → pages 75, 241

[129] J. Meyers, W. Munters, and J. Goit. A framework for optimization of turbulent wind-farm boundary layers and application to optimal control of wind-farm energy extraction. In *Proceedings of the 2016 American Control Conference*, pages 519–524, 2016. → page 240

[130] W. Miao, C. Li, J. Yang, and X. Xie. Numerical investigation of the yawed wake and its effects on the downstream wind turbine. *Journal of Renewable and Sustainable Energy*, 8(3), 2016. doi:10.1063/1.4953791. → pages 46, 242

[131] M. Mirzaei, T. Göçmen, G. Giebel, P. E. Sørensen, and N. K. Poulsen. Turbine control strategies for wind farm power Optimization. In *Proceedings of the 2015 American Control Conference*, pages 1709–1714, 2015. → pages 69, 234

[132] T. P. Miyanawala and R. K. Jaiman. A low-dimensional learning model via convolutional neural networks for unsteady wake-body interaction. *Accepted for review in the Journal of Fluid Mechanics*, n/a, 2018. doi:arXiv:1807.09591v1. → pages 20, 65

[133] G. Mosetti, C. Poloni, and D. Diviacco. Optimization of wind turbine positioning in large wind farms by means of a Genetic algorithm. J Wind Eng Ind Aerody 51:105–116. *Journal of Wind Engineering and Industrial Aerodynamics*, 51(51):105–116, 1994. → page 12

[134] M. A. Muller and F. Allgower. Economic and Distributed Model Predictive Control : Recent Developments in Optimization-Based Control. *SICE Journal of Control, Measurement, and System Integration*, 10(2):39–52, 2017. → pages 139, 177

[135] M. A. Müller, M. Reble, and F. Allgöwer. Cooperative control of dynamically decoupled systems via distributed model predictive control. *International Journal of Robust and Nonlinear Control*, 22:1376–1397, 2012. → pages 141, 177

[136] W. Munters and J. Meyers. Effect of wind turbine response time on optimal dynamic induction control of wind farms. *Journal of Physics: Conference Series*, 753, 2016. ISSN 17426596. doi:10.1088/1742-6596/753/5/052007. → page 240

[137] W. Munters and J. Meyers. Optimal dynamic induction and yaw control of wind farms: Effects of turbine spacing and layout. *Journal of Physics: Conference Series*, 1037, jun 2018. ISSN 1742-6588. doi:10.1088/1742-6596/1037/3/032015. → pages 50, 56, 245

[138] W. Munters and J. Meyers. Towards practical dynamic induction control of wind farms : analysis of optimally controlled wind-farm boundary layers

and sinusoidal induction control of first-row turbines. *Wind Energy Science*, 3:409–425, 2018. → pages 42, 43, 55, 56, 60, 65, 66, 240

[139] W. Munters and J. Meyers. Dynamic strategies for yaw and induction control of wind farms based on large-eddy simulation and optimization. *Energies*, 11(177), 2018. ISSN 1996-1073. doi:10.3390/en11010177. → pages 50, 51, 59, 60, 65, 66, 245

[140] R. R. Negenborn and J. M. Maestre. Distributed model predictive control: An overview and roadmap of future research opportunities. *IEEE Control Systems Magazine*, 34(4):87–97, 2014. ISSN 1066033X. doi:10.1109/MCS.2014.2320397. → page 139

[141] K. Nilsson, S. Ivanell, K. S. Hansen, R. Mikkelsen, J. N. Sørensen, S.-P. Breton, and D. Henningson. Large-eddy simulations of the Lillgrund wind farm. *Wind Energy*, 18(3):449–467, 2015. doi:10.1002/we.1707. → pages 2, 10, 34, 68, 69, 175, 235

[142] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Science + Business Media, New York, NY, 2nd edition, 2006. → page 85

[143] J. G. J. Olivier, G. Janssens-Maenhout, M. Muntean, and J. A. H. W. Peters. Trends in Global CO2 Emissions: 2016 Report. Technical report, PBL Netherlands Environmental Assessment Agency, 2016. → page 8

[144] J. Pannek. Parallelizing a state exchange strategy for noncooperative distributed NMPC. *Systems and Control Letters*, 62(1):29–36, 2013. ISSN 01676911. doi:10.1016/j.sysconle.2012.10.015. → pages 141, 177

[145] J. Park and K. H. Law. Cooperative wind turbine control for maximizing wind farm power using sequential convex programming. *Energy Conversion and Management*, 101:295–316, 2015. doi:10.1016/j.enconman.2015.05.031. → page 244

[146] J. Park and K. H. Law. A data-driven, cooperative wind farm control to maximize the total power production. *Applied Energy*, 165:151–165, 2016. ISSN 03062619. doi:10.1016/j.apenergy.2015.11.064. → pages 49, 62, 236, 244

[147] J. Park, S. Kwon, and K. H. Law. Wind farm power maximization based on a cooperative static game approach. In *Proceedings of the 2013 SPIE Smart Structures and Materials + Nondestructive Evaluation and Health Monitoring*, 2013. doi:10.1117/12.2009618. → page 244

[148] J. Park, S.-D. Kwon, and K. Law. A data-driven, cooperative approach for wind farm control: A wind tunnel experimentation. *Energies*, 10(852), 2017. ISSN 1996-1073. doi:10.3390/en10070852. → pages 50, 60, 61, 65, 69, 245

[149] J. Quick, J. Annoni, R. King, K. Dykes, and P. Fleming. Optimization Under Uncertainty for Wake Steering Strategies. Technical report, National Renewable Energy Agency, 2017. → page 241

[150] S. Raach, D. Schlipf, F. Beyer, and P. W. Cheng. Wake redirecting using feedback control to improve the power output of wind farms. In *Proceedings of the 2016 American Control Conference*, pages 1387–1392, 2016. → page 244

[151] A. Richards and J. P. How. Robust distributed model predictive control. *International Journal of Control*, 80(9):1517–1531, 2007. → pages 140, 141, 177, 193

[152] A. Robertson, J. Jonkman, and M. Masciola. Definition of the Semisubmersible Floating System for Phase II of OC4. Technical report, National Renewable Energy Laboratory. Report number: NREL/TP-5000-60601, 2014. → pages xiii, xx, xxii, 85, 86, 107, 126, 127, 131, 180, 185, 186

[153] S. F. Rodrigues, R. Teixeira Pinto, M. Soleimanzadeh, P. A. Bosman, and P. Bauer. Wake losses optimization of offshore wind farms with moveable floating wind turbines. *Energy Conversion and Management*, 89:933–941, 2015. doi:10.1016/j.enconman.2014.11.005. → pages 17, 52, 70, 247

[154] S. G. Rubin. A review of marching procedures for parabolized Navier-Stokes equations. In T. Cebeci, editor, *Numerical and Physical Aspects of Aerodynamic Flows*, chapter 11, pages 171–185. Springer Science + Business Media, New York, 1st edition, 1982. → page 31

[155] A. Sanfelice Bazanella, L. Campestrini, and D. Eckhard. *Data-driven Controller Design: The H2 Approach*. Springer Netherlands, 1st edition, 2012. ISBN 978-94-007-2300-9. doi:10.1007/978-94-007-2300-9. → page 22

[156] C. Santoni, U. Ciri, M. Rotea, and S. Leonardi. Development of a high fidelity CFD code for wind farm control. In *Proceedings of the 2015 American Control Conference*, pages 1715–1720, 2015. → pages 34, 235

[157] J. G. Schepers and S. P. Van Der Pijl. Improved modelling of wake aerodynamics and assessment of new farm control strategies. *Journal of Physics: Conference Series*, 75, 2007. ISSN 17426588. doi:10.1088/1742-6596/75/1/012039. → pages 36, 43, 69, 236

[158] J. Schreiber, S. Cacciola, F. Campagnolo, V. Petrović, D. Mourembles, and C. L. Bottasso. Wind shear estimation and wake detection by rotor loads - First wind tunnel verification. *Journal of Physics: Conference Series*, 753 (3), 2016. doi:10.1088/1742-6596/753/3/032027. → pages 22, 65

[159] J. Serrano González, M. B. Payán, and J. R. Santos. Optimum wind turbines operation for minimizing wake effect losses in offshore wind farms. In *Proceedings of the 2013 International Conference on Environment and Electrical Engineering*, 2013. doi:10.1109/EEEIC-2.2013.6737905. → page 233

[160] J. Serrano González, M. Burgos Payán, J. Riquelme Santos, and Á. G. González Rodríguez. Maximizing the overall production of wind farms by setting the individual operating point of wind turbines. *Renewable Energy*, 80:219–229, 2015. ISSN 18790682. doi:10.1016/j.renene.2015.02.009. → pages 69, 233

[161] R. Shakoor, M. Y. Hassan, A. Raheem, and Y. K. Wu. Wake effect modeling: A review of wind farm layout optimization using Jensen's model. *Renewable and Sustainable Energy Reviews*, 58:1048–1059, 2016. doi:10.1016/j.rser.2015.12.229. → pages 10, 12, 68, 104, 124

[162] C. R. Shapiro, P. Bauweraerts, J. Meyers, C. Meneveau, and D. F. Gayme. Model-based receding horizon control of wind farms for secondary frequency regulation. *Wind Energy*, 20(7):1261–1275, 2017. ISSN 10991824. doi:10.1002/we.2093. → pages 105, 106

[163] C. R. Shapiro, D. F. Gayme, and C. Meneveau. Modelling yawed wind turbine wakes: a lifting line approach. *Journal of Fluid Mechanics*, 841: 1–12, 2018. ISSN 0022-1120. doi:10.1017/jfm.2018.75. → pages 48, 106, 243

[164] J. Shu, B. H. Zhang, and Z. Q. Bo. A wind farm coordinated controller for power optimization. In *Proceedings of the 2011 IEEE Power and Energy Society General Meeting*, 2011. doi:10.1109/PES.2011.6039552. → pages 42, 59, 238

[165] A. J. Sørensen. A survey of dynamic positioning control systems. *Annual Reviews in Control*, 35(1):123–136, 2011. ISSN 13675788. doi:10.1016/j.arcontrol.2011.03.008. → page 17

[166] N. N. Sørensen. *General Purpose Flow Solver Applied to Flow Over Hills*. PhD Thesis. Technical University of Denmark, 1995. → page 30

[167] E. staff. Electricity production, 2018. URL https://yearbook.enerdata.net/electricity/world-electricity-production-statistics.html. → page 8

[168] S. staff. STAR-CCM+, 2018. URL https://mdx.plm.automation.siemens.com/star-ccm-plus. → page 30

[169] M. Steinbuch, W. D. Boer, and O. Bosgra. Optimal control of wind power plants. *Journal of Wind Engineering and Industrial Aerodynamics*, 27: 237–246, 1988. ISSN 01676105. doi:10.1016/0167-6105(88)90039-6. → pages 14, 234

[170] B. T. Stewart, S. J. Wright, and J. B. Rawlings. Cooperative distributed model predictive control for nonlinear systems. *Journal of Process Control*, 21(5):698–704, 2011. ISSN 09591524. doi:10.1016/j.jprocont.2010.11.004. → pages 141, 177

[171] J. Tian, C. Su, M. Soltani, and Z. Chen. Active power dispatch method for a wind farm central controller considering wake effect. In *Proceedings of the 2014 Industrial Electronics Conference*, pages 5450–5456, 2014. doi:10.1109/IECON.2014.7049333. → pages 69, 233

[172] J. Tian, D. Zhou, C. Su, F. Blaabjerg, and Z. Chen. Optimal control to increase energy production of wind farm considering wake effect and lifetime estimation. *Applied Sciences*, 7(65), 2017. doi:10.3390/app7010065. → page 234

[173] U.S Department of Energy staff. U.S. Energy and Employment Report. Technical report, U.S. Department of Energy, 2017. → page 9

[174] U.S. Energy Information Administration staff. Assumptions to the Annual Energy Outlook 2018: Electricity Market Module. Technical report, U.S. Energy Information Administration, 2018. → page 10

[175] M. Vali, V. Petrović, B. Sjoerd, J.-W. van Wingerden, and M. Kühn. Adjoint-based model predictive control of wind farms: Beyond the quasi steady-state power maximization. *International Federation of Automatic*

Control, 50(1):4510–4515, 2017. doi:10.1016/j.ifacol.2017.08.382. → pages 41, 59, 64, 239

[176] D. van der Hoek, S. Kanev, J. Allin, D. Bieniek, and N. Mittelmeier. Effects of axial induction control on wind farm energy production - A field test. *Renewable Energy*, 140:994–1003, 2019. doi:10.1016/j.renene.2019.03.117. → pages 43, 44, 56, 240

[177] M. T. van Dijk, J.-W. van Wingerden, T. Ashuri, Y. Li, and M. A. Rotea. Yaw-misalignment and its impact on wind turbine loads and wind farm power output. *Journal of Physics: Conference Series*, 753, 2016. ISSN 1742-6588. doi:10.1088/1742-6596/753/6/062013. → page 241

[178] M. T. van Dijk, J. W. van Wingerden, T. Ashuri, and Y. Li. Wind farm multi-objective wake redirection for optimizing power production and loads. *Energy*, 121:561–569, 2017. ISSN 03605442. doi:10.1016/j.energy.2017.01.051. → page 241

[179] L. J. Vermeer, J. N. Sørensen, and A. Crespo. Wind turbine wake aerodynamics. *Progress in Aerospace Sciences*, 39(6-7):467–510, 2003. ISSN 03760421. doi:10.1016/S0376-0421(03)00078-2. → pages 25, 105

[180] P. Vermeulen and P. Builtjes. Mathematical Modelling of Wake Interaction in Wind Turbine Arrays. Technical report, Netherlands Organization for Applied Scientific Research, Division of Technology for Society, 1981. → page 234

[181] J. Wagenaar, L. Machielse, and J. Schepers. Controlling Wind in ECN's Scaled Wind Farm. Technical report, Energy Research Centre of the Netherlands, 2012. → pages 16, 47, 243

[182] C. Wang, J. Wang, F. Campagnolo, D. B. Carraón, and C. L. Bottasso. Validation of large-eddy simulation of scaled waked wind turbines in different yaw misalignment conditions. *Journal of Physics: Conference Series*, 1037, jun 2018. ISSN 1742-6588. doi:10.1088/1742-6596/1037/6/062007. → pages 30, 231

[183] L. Wang, A. Tan, and Y. Gu. A novel control strategy approach to optimally design a wind farm layout. *Renewable Energy*, 95:10–21, 2016. ISSN 18790682. doi:10.1016/j.renene.2016.03.104. → page 233

[184] R. Wang, I. R. Manchester, and J. Bao. Distributed economic MPC with separable control contraction metrics. *IEEE Control Systems Letters*, 1(1):

104–109, 2017. ISSN 24751456. doi:10.1109/LCSYS.2017.2708660. → pages 140, 178

[185] I. J. Wolf, H. Scheu, and W. Marquardt. A hierarchical distributed economic NMPC architecture based on neighboring-extremal updates. In *Proceedings of the American Control Conference*, pages 4155–4160. IEEE, 2012. ISBN 9781457710957. doi:10.1109/acc.2012.6315353. → pages 140, 178

[186] C. Yan. *Wind Turbine Wakes: From Numerical Modeling to Machine Learning*. PhD Thesis. University of Delaware, 2018. → pages 20, 22, 65

[187] Z. Yang, Y. Li, and J. E. Seem. Optimizing energy capture of cascaded wind turbine array with nested-loop extremum seeking control. *Journal of Dynamic Systems, Measurement, and Control*, 137(12), 2015. doi:10.1115/1.4031593. → pages 40, 41, 238

[188] S. Zhong and X. Wang. Decentralized model-free wind farm control via discrete adaptive filtering methods. *IEEE Transactions on Smart Grid*, 9 (4):2529–2540, 2018. ISSN 19493053. doi:10.1109/TSG.2016.2614434. → pages 38, 39, 59, 61, 237

Appendix A

Wind Farm Control Review Tables

Table A.1: Summary of wake models and simulation tools used within wind farm control studies that have been reviewed in the current chapter.

Model	Type	Fidelity	Time-dependency	Wake phenomena	Relative error
Park	Parametric	Low	Steady	Momentum deficit	0.1 to 15.9 %[1]
FLORIS	Parametric	Low	Steady	Momentum deficit Wake redirection	0.1 to 5.3 %[2]
FLORIDyn	Parametric	Low	Quasi-dynamic	Momentum deficit Wake redirection	7.9 to 11.7 %[3]
FOWFSim	Parametric	Low	Steady	Momentum deficit Wake redirection Floating turbine motion	-
WFSim	Turbulent 2D NS	Medium	Dynamic	Momentum deficit Wake redirection	82.1 % VAF[4]
Ainslie	Turbulent axisymmetric NS	Medium	Steady	Momentum deficit	0.5 to 8.4 %[5]
FarmFlow	Turbulent 3D parabolized NS	Medium	Steady Quasi-dynamic	Momentum deficit	0.2 to 26.9 %[6]
SOWFA	3D LES + Actuator line theory	High	Dynamic	Momentum deficit Wake redirection	0.2 to 14 %[7]
SP-Wind	3D LES + Actuator disc theory	High	Dynamic	Momentum deficit Wake redirection	-

[1] Relative error results were obtained by digitizing *Figure 3* and calculating the percent deviation of the Park model results relative to the TNO wind tunnel results [98].

[2] Relative error results were obtained from *Table II* [72].

[3] RMSE values were obtained from *Figure 5a* [67]

[4] Variance accounted for (VAF) result was obtained from *Table 3* [23, 25]

[5] Relative error results were obtained by digitizing *Figure 2* [4]

[6] Relative error results were obtained by digitizing *Figures 2* to *4* and comparing UPM Code results with TNO experimental results [45].

[7] Relative error results were obtained from *Tables 1* and *2* [182].

Table A.1 continued from previous page

Model	Type	Fidelity	Time-dependency	Wake phenomena	Relative error
UTD-WF	3D LES + Actuator disc/line theory	High	Dynamic	Momentum deficit Wake redirection	-
EllipSys3D	3D RANS	High	Dynamic	Momentum deficit Wake redirection	-
STAR-CCM+	3D RANS/LES	High	Dynamic	Momentum deficit Wake redirection	-

Table A.2: Summary of studies that have assessed the steady-state potential of power de-rating.

Ref.	Optimization/analysis method	Inputs	Optimization model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[20]	Centralized particle swarm optimization	λ, β	Steady Park model	Steady Park model	Const. speed	10.57 %[8]
[21]	Centralized dynamic programming	a	Steady Park model	Steady Park model	Const. speed	8.33 %[9]
[47]	Manual optimization	λ	Steady Park model	Steady Park model	Const. speed	12.80 %[10]
[159]	Centralized genetic algorithm	λ, β	Steady Park model	Steady Park model	Const. speed	10.94 %[11]
[160]	Centralized genetic algorithm	λ, β	Steady Park model	Steady Park model	Const. speed	7.55 %[12]
[171]	Centralized particle swarm optimization	λ, β	Steady Park model	Steady Park model	Const. speed	26.77 %[13]
[84]	Centralized sequential optimization	a	Steady Park model	Steady Park model	Const. speed	5.40 %[14]
[183]	Distributed genetic algorithm	a	Steady Park model	Steady Park model	Const. speed	1.52 %[15]
[14]	Centralized sequential quadratic programming	a	Steady Park model	Steady Park model	Const. speed	-[16]

[8] Relative wind farm efficiency gain corresponds to *simulation case 1*, and was obtained from discussions in the section entitled *Case Study* [20].

[9] Relative wind farm efficiency gain was obtained from discussions in *Section III* [21].

[10] Relative wind farm efficiency gain was obtained by digitizing *Figure 18* at a wind direction of 90 deg [47].

[11] Relative wind farm efficiency gain was obtained from discussions in *Section IV* [159].

[12] Relative wind farm efficiency gain was obtained from discussions in *Section 4* [160].

[13] Relative wind farm efficiency gain was obtained from discussions in *Section VII* [171].

[14] Relative wind farm efficiency gain was obtained from *Table 1* [84].

[15] Relative wind farm efficiency gain was obtained from *Table 5* [183].

[16] The wind farm power output corresponding to an uncontrolled wind farm is not listed; therefore wind farm efficiency gains cannot be calculated [14].

Table A.2 continued from previous page

Ref.	Optimization/analysis method	Inputs	Optimization model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[22]	Centralized particle swarm optimization	a	Steady Park model	Steady Park model	Const. speed	9.96 %[17]
[172]	Manual optimization	λ, β	Steady Park model	Steady Park model	Const. speed	5.41 %[18]
[131]	Centralized genetic algorithm	P	Steady Park model	Steady Park model	Const. speed	13.34 %[19]
[52]	Manual optimization	C_P	Modified steady park model	Modified steady park model	Const. speed Turbulence	2.6 %[20]
[85]	Centralized sequential quadratic programming	ω	Steady empirical model	Steady empirical model	Const. speed	2.85 %[21]
[42]	Centralized gradient descent optimization	P	Steady parametric wake model [119]	Steady parametric wake model [119]	Const. speed	3.23 %[22]
[169]	Manual optimization	λ	Steady Milly model [180]	Steady Milly model [180]	Const. speed	4 %[23]
[113]	Centralized genetic algorithm	β	Steady eddy viscosity model	Steady eddy viscosity model	Const. speed Turbulence	4.50 %[24]
[105]	Numerical simulation	P	No optimization	Dynamic Ainslie model	Const. speed Turbulence	4.10 %[25]

[17] Relative wind farm efficiency gain was obtained from discussions in *Section 5* [22].

[18] Relative wind farm efficiency gain was obtained from *Figure 8* [172].

[19] Relative wind farm efficiency gain was obtained from *Table I* and corresponds to the gain obtained from using the constant rotor speed control method relative to constant reference control at 5 MW [131].

[20] Relative wind farm efficiency gain was obtained from *Figure 12a*.

[21] Relative wind farm efficiency gain was obtained from *Table I* [85].

[22] Relative wind farm efficiency gain was obtained from *Tables 4.1 and 4.2* [42].

[23] Relative wind farm efficiency gain was obtained from discussions under the heading entitled *Maximizing energy capture* in *Section B* [169].

[24] Relative wind farm efficiency gain was obtained from discussions in *Section V* [113].

[25] Relative wind farm efficiency gain was obtained from discussions in *Section 3.1* [105].

Table A.2 continued from previous page

Ref.	Optimization/analysis method	Inputs	Optimization model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[141]	Numerical simulation	β	No optimization	EllipSys3D (dynamic CFD)	Const. speed Turbulence Wind shear	$-7.70\,\%$[26]
[9]	Centralized grid search	β	Steady Park model	Steady FLORIS model SOWFA[27]	Const. speed Turbulence Wind shear	$24.82\,\%$[28] $-9.35\,\%$[29]
[48]	Numerical simulation	β	No optimization	Dynamic LES	Const. speed Turbulence Wind shear	$2.80\,\%$[30]
[99]	Numerical simulation	λ, β	No optimization	MULTI3 (Dynamic k-ω RANS model)	Const. speed Turbulence Wind shear	$2.5\,\%$[31] $9.7\,\%$[32]
[156]	Centralized dynamic programming	λ	Steady Park model	UTD-WF (Dynamic LES)	Const. speed Turbulence Wind shear	$5.7\,\%$[33] $0.2\,\%$[34]

[26] Relative wind farm efficiency gain was obtained from *Table III* [141].

[27] The listed simulated wind conditions correspond to the dynamic SOWFA CFD tool. Constant wind speeds were simulated when using the FLORIS wake model.

[28] Relative wind farm efficiency gain corresponds to the steady FLORIS wake model, and was obtained by digitizing *Figure 8*[9].

[29] Relative wind farm efficiency gain corresponds to the dynamic SOWFA CFD tool, and was obtained by digitizing *Figure 8*[9].

[30] Relative wind farm efficiency gain was obtained from *Figure 8* [48].

[31] Relative wind farm efficiency gain corresponds to only WTG 4 and 11, and was obtained from *Figure 13b* and discussions in *Section 3.2.2* [99].

[32] Relative wind farm efficiency gain corresponds to all four wind turbines, and was obtained from *Figure 13b* and discussions in *Section 3.2.2* [99].

[33] Relative wind farm efficiency gain was obtained from *Table III* and corresponds to evaluation using the Jensen wake model [156].

[34] Relative wind farm efficiency gain was obtained from *Table III* and corresponds to evaluation using UTD-WF [156].

Table A.2 continued from previous page

Ref.	Optimization/analysis method	Inputs	Optimization model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[43][35]	Wind tunnel experimentation	β	No optimization	Scaled wind farm	-	4.60 %[36]
[31]	Wind tunnel experimentation	P	No optimization	Scaled wind farm	-	0.90 %[37]
[16]	Wind tunnel experimentation	λ, β	No optimization	Scaled wind farm	-	0 %[38]
[118]	Wind tunnel experimentation	β	No optimization	Scaled wind farm	-	3.46 %[39]
[2]	Wind tunnel experimentation	λ, β	No wind farm optimization	Scaled wind farm	-	3.26 %[40] 3.81 %[41]
[157]	Field experimentation + Noise reduced wind turbine operation	β	No optimization	ECN's EWTW wind farm	-	Inconclusive[42]
[26]	Field experimentation	β	No optimization	ECN's EWTW wind farm	-	Inconclusive[43]

[35] Additional information is available in a companion study [146].

[36] Relative wind farm efficiency gain was obtained from discussions in *Section 5.2* [43].

[37] Relative wind farm efficiency gain was obtained from *Figure 7* [31].

[38] The zero gain in wind farm efficiency is evident from *Figures 4a* and *4b* [16]; namely, the combinations of turbine operating parameters that maximize wind farm performance are those that maximize individual turbine performance.

[39] Relative wind farm efficiency gain corresponds to a wind speed of 9 m/s, and was obtained by digitizing *Figure 5.3* [118].

[40] Relative wind farm efficiency gain corresponds to a scenario where the upstream turbine operates at a tip-speed ratio λ that was higher than its optimal value, and was obtained from by digitizing *Figure 10b* [2].

[41] Relative wind farm efficiency gain corresponds to the gain in wind farm power output observed when the upstream turbine's collective blade pitch angle β was varied from 0 deg to 2 deg at a spacing of six rotor diameters, and was obtained by digitizing *Figure 10a* [2].

[42] There is no clear discrepancy between the wind farm power output results in *Figure 10*, which correspond to wind farm operation under different control strategies [157]. It is mentioned by the authors that the concept of axial induction control is validated however, since little change in the wind farm power output indicates that the loss in power output from the upstream turbine is compensated by a gain in power output from the remaining turbines.

[43] There is no discernible rise in the wind farm power coefficient in *Figure 4.1.f* [26]. A wind power output increase is observed at a wind speed of 8 m/s; however the upstream turbine also experiences a power output increase at this operating point. Since the upstream turbine is undisturbed, this data point is deemed erroneous. This point is confirmed by the authors in *Section 4.1*.

Table A.3: Summary of studies that have developed wind farm controllers based on the concept of power de-rating.

Ref.	Control method	Inputs	Control model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[1]	Centralized artificial bee optimization + Maximum power-point tracking wind turbine control	λ	Steady Park model	Steady Park model	Const. speed Var. direction	4.03 %[44]
[73]	Centralized sequential quadratic programming optimization + Feedback linearization and model predictive wind turbine control	a	Steady Park model	Steady Park model	Var. speed	4 %[45]
[122]	Centralized/distributed game-theoretic control	a	Model-free	Steady Park model	Const. speed	34.05 %[46]
[15]	Centralized/distributed gradient estimation-based control	a	Model-free	Steady Park model	Const. speed	-[47]
[188]	Distributed discrete adaptive filtering optimization	a	Model-free	Steady Park model	Const. speed Var. direction Turbulence[48]	3.87 %[49]

[44] Relative wind farm efficiency gain was obtained by digitizing *Figure 12g* and corresponds to a wind direction of 180 deg [1].

[45] Relative wind farm efficiency gain was obtained from discussions in *Section 5*[73].

[46] Relative wind farm efficiency gain was calculated based on results presented in *Figure 5b* [122]. This figure presents the wind farm power output normalized relative to the optimal wind farm power output. The normalized power output was converted to the relative gain in wind farm efficiency via simple algebraic manipulation.

[47] Power production from the suboptimal wind farm is not presented; therefore the efficiency gain cannot be calculated [15].

[48] The authors mention that a random variable is added to the power output of each turbine at each iteration to simulate environmental disturbances [188]. No statistical information is provided regarding this random signal.

[49] Relative wind farm efficiency gain corresponds to Algorithm 2, and was obtained by digitizing *Figure 13* [188].

Table A.3 continued from previous page

Ref.	Optimization/analysis method	Inputs	Control model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[94][50]	Centralized extremum seeking control	a	Model-free	Dynamic Park model	Var. speed Turbulence	$3.8\,\%$[51] $-13.2\,\%$[52]
[70][53]	Distributed maximum power-point tracking	a	Model-free	Dynamic Park model	Var. speed	$4\,\%$[54]
[3]	Centralized multi-resolution simultaneous perturbation stochastic approx.	a	Model-free	Dynamic Park model	Var. speed Var. direction	$32\,\%$[55]
[187]	Centralized nested-loop extremum seeking control	T_{gen}	Model-free	Dynamic SimWindFarm	Const. speed Turbulence	$1.30\,\%$[56]
[164]	Centralized nonlinear model predictive control + Maximum power-point tracking wind turbine control	ω	Dynamic mosaic-tile model	Dynamic mosaic-tile model	Var. speed	$1.96\,\%$[57]

[50] Additional information is available in a companion publication [95].

[51] Relative wind farm efficiency gain corresponds to a low turbulence intensity setting of 2 %, and was obtained from discussions in *Section 5* [94].

[52] Relative wind farm efficiency gain corresponds to a high turbulence intensity setting of 8 %, and was obtained from discussions in *Section 5* [94].

[53] Additional information is available in companion publications [66, 71].

[54] Relative wind farm efficiency gain was obtained from *Figure 8* and from discussions in *Section 5.4* [70].

[55] Relative wind farm efficiency gain was obtained from *Figure 10* [3].

[56] Relative wind farm energy production gain was obtained from discussions in the section entitled *Simulation study: Simulation for turbulent wind* [187].

[57] Relative wind farm efficiency gain corresponds to a turbine spacing of 560 m (i.e. seven rotor diameters), and was obtained from *Figure 9* [164].

Table A.3 continued from previous page

Ref.	Optimization/analysis method	Inputs	Control model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[83]	Centralized brute-force optimization + Model predictive wind turbine control	λ, β	Steady Park model	Dynamic SimWindFarm	Var. speed Turbulence	0.41 %[58] 1.42 %[59]
[106]	Centralized Nelder-Mead simplex optimization + Demanded power-point tracking wind turbine control	P	Steady Ainslie model [104]	Dynamic Ainslie model [104]	Const. speed Turbulence	2.92 %[60] −2.88 %[61]
[100]	DTU wind farm controller + Maximum power-point tracking wind turbine control	P	PossPOW wake model [74]	PossPOW wake model [74]	Const. speed Turbulence	1 %[62]
[175]	Centralized adjoint-based model predictive control	a	Dynamic WFSim	Dynamic WFSim	Const. speed Var. direction	23.59 %[63] −3.6 %[64]

[58] Relative energy production gain corresponds to a 1 × 3 wind-aligned row wind farm modeled using FAST, and was obtained by manipulating the $E_{\text{tot,greedy}}/E^*_{\text{tot}}$ ratio described in *Section V.A* [83]. This ratio was manipulated to determine the energy production increase resulting from the proposed MPC controller, operating at optimal conditions dictated by the wind farm optimization algorithm, relative to a traditional greedy controller.

[59] Relative energy production gain corresponds to a 1 × 10 wind-aligned row wind farm modeled using Aeolus SimWindFarm, and was obtained by manipulating the $E_{\text{tot,greedy}}/E^*_{\text{tot}}$ ratio described in *Section V.B* [83]. This ratio was manipulated to determine the energy production increase resulting from the proposed MPC controller, operating at optimal conditions dictated by the wind farm optimization algorithm, relative to a traditional greedy controller.

[60] Relative wind farm efficiency gain corresponds to a turbulence intensity of 7.5 %, a mean wind speed of 8 m/s, a controller sampling frequency of 0.1 Hz, and a turbine spacing of seven rotor diameters, and was obtained from *Figure 10* [106].

[61] Relative wind farm efficiency gain corresponds to a turbulence intensity of 21 %, a mean wind speed of 10.5 m/s, a controller sampling frequency of 0.01 Hz, and a turbine spacing of seven rotor diameters, and was obtained from *Figure 11* [106].

[62] Relative wind farm efficiency gain was obtained from discussions in *Section III.B* [100].

[63] Relative wind farm efficiency gain was obtained by digitizing *Figure 4* [175]. The value corresponds to the maximum relative gain in wind farm power output over the course of the simulation.

[64] Relative wind farm efficiency gain was obtained by digitizing *Figure 4* [175]. The value corresponds to the maximum relative loss in wind farm

Table A.3 continued from previous page

Ref.	**Optimization/analysis method**	**Inputs**	**Control model**	**Evaluation model**	**Simulated wind conditions**	**Relative efficiency gain**
[41]	Centralized nested extremum-seeking control	T_{gen}	Model-free	UTD-WF	Const. speed Turbulence	$7.8\,\%$[65]
[76]	Centralized model predictive control	C_T	Dynamic LES	Dynamic LES	Const. speed Turbulence Wind shear	$15.80\,\%$[66]
[138][67]	Centralized model predictive control	C_T	Dynamic LES	Dynamic LES	Const. speed Turbulence Wind shear	$21.19\,\%$[68]
[77][69]	Centralized model predictive control	C_T	Dynamic LES	Dynamic LES	Const. speed Turbulence Wind shear	$7\,\%$[70]
[176]	Lookup table for optimal blade pitch angle settings	β	FarmFlow	Goole Fields wind farm	-	-15.9 to $11.1\,\%$[71]

power output over the course of the simulation.

[65] Relative wind farm efficiency gain was obtained by from *Table 3* [41].

[66] Relative wind farm efficiency gain corresponds to a simulation with no penalization of turbulent dissipation, and was obtained from *Table 3* [76].

[67] Additional information is available in a companion publication [136].

[68] Relative wind farm efficiency gain corresponds to the least restrictive C3t0 controller setting, and was obtained by digitizing *Figure 3a* [138].

[69] This study expands on previous work by considering wind farm entrance effects [76]. Additional information is available in a companion study [129].

[70] Relative wind farm efficiency gain was obtained from discussions in *Section 3.1* [77].

[71] Range of relative wind farm efficiency gains was obtained by digitizing *Figure 12*, and corresponds to different wind speeds while the wind was aligned with the row of turbines [176].

Table A.4: Summary of studies that have assessed the steady-state potential of yaw-based wake redirection.

Ref.	Optimization/analysis method	Inputs	Optimization model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[128][72]	Manual optimization	γ	Interpolated wind tunnel test data	Interpolated wind tunnel test data	-	8.9 %[73]
[11]	Distributed alternating direction method of multipliers algorithm	γ	Steady wake model	Steady wake model	Const. speed	17.7 %[74]
[149]	Centralized constrained optimization by using a linear approximation	γ	Steady FLORIS model	Steady FLORIS model	Const. speed	3.29 %[75]
[63]	Centralized sequential quadratic programming	γ	Steady FLORIS model	Steady FLORIS model	Const. speed	7.7 %[76]
[178]	Centralized quasi-Newton algorithm	γ	Modified steady FLORIS model	Modified steady FLORIS model	Const. speed	2.85 %[77]
[177]	Centralized game-theoretic algorithm	γ	Modified steady FLORIS model	Modified steady FLORIS model	Const. speed	5.49 %[78]

[72] Additional information is available in a companion publication [46].

[73] Relative wind farm efficiency gain corresponds to a turbine spacing of three rotor diameters, and was obtained from *Table 2* in *Paper 2* [128].

[74] Relative wind farm efficiency gain corresponds to the proposed distributed algorithm, and was obtained from *Table II* [11].

[75] Relative wind farm efficiency gain was obtained from *Table 1* and corresponds to the stochastic average power production increase obtained from using the OUU solution relative to no yaw offsets [149].

[76] Relative wind farm efficiency gain corresponds to the original wind farm layout with yaw optimization, and was obtained from discussions in *Section 5.2* [63].

[77] Relative wind farm efficiency gain corresponds to the single-objective optimization of power case ($\lambda = 1$), and was obtained from *Table 1* [178].

[78] Relative wind farm efficiency gain was obtained from *Table 1* and corresponds to the single-objective optimization of power case ($\gamma = 1$) for a wind direction of 5 deg [177].

Table A.4 continued from previous page

Ref.	**Optimization/analysis method**	**Inputs**	**Optimization model**	**Evaluation model**	**Simulated wind conditions**	**Relative efficiency gain**
[65]	Centralized gradient-based sequential quadratic programming	x, y, γ	Modified steady FLORIS model	Modified steady FLORIS model	Const. speed	5.26 %[79]
[130]	Numerical simulation	γ	No wind farm optimization	STAR-CCM+ (dynamic CFD)	Const. speed Turbulence Wind shear	2.11 %[80]
[39]	Numerical simulation	γ	No wind farm optimization	Dynamic LES	Const. speed Turbulence Winds shear	13.77 %[81]
[58][82]	Numerical simulation	γ	No wind farm optimization	SOWFA (dynamic CFD)	Const. speed Turbulence Wind shear	4.6 %[83]
[60]	Numerical simulation	γ	No wind farm optimization	SOWFA (dynamic CFD)	Const. speed Turbulence Wind shear	12.5 %[84]
[2]	Wind tunnel experimentation	γ	No wind farm optimization	Scaled wind farm	-	12 %[85]

[79] Relative wind farm annual energy production (AEP) gain corresponds to the case with both layout and wake steering optimization, and was obtained from *Table II* [65].

[80] Relative wind farm efficiency gain was obtained from *Figure 7b* and discussions in *Section III.A* [130].

[81] Relative wind farm efficiency gain was obtained by digitizing *Figure 8a* and corresponds to a yaw offset of 25deg [39].

[82] Additional information is available in companion publications [62, 66].

[83] Relative wind farm efficiency gain was obtained from *Figure 4* [58].

[84] Relative wind farm efficiency gain was obtained from *Figure 4* [58].

[85] Relative wind farm efficiency gain was obtained from discussions in *Section 3.2.4* and *Figure 10c*, and corresponds to the highest wind farm power output gain observed in response to varying the nacelle yaw angle γ of the upstream turbine [2].

Table A.4 continued from previous page

Ref.	Optimization/analysis method	Inputs	Optimization model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[31]	Wind tunnel experimentation	γ	No optimization	Scaled wind farm	-	25.2 % %[86] 7.0 %[87]
[19]	Wind tunnel experimentation	γ	No optimization	Scaled wind farm	-	17 %[88]
[181]	Field experimentation	γ	No wind farm optimization	ECN's scaled wind farm	-	Inconclusive[89]
[86]	Field experimentation	γ	Lifting line model [163]	1×6 wind farm in Alberta, CA	-	7 to 47 %[90]

[86] Relative wind farm efficiency gain corresponds to a lateral downstream wind turbine displacement of -0.45 rotor diameters, and was obtained from *Figure 11* [31].

[87] Relative wind farm efficiency gain corresponds to no lateral downstream wind turbine displacement, and was obtained from *Figure 11* [31].

[88] Relative wind farm efficiency gain was obtained from *Figure 5*, and corresponds to the right-most column of colored dots that represent turbine yaw angles [19].

[89] There is no discernible pattern that relates the yaw angles of individual turbines to the wind farm power output; this point is discussed in *Section 3.3* and is evident in *Figure 6* [181].

[90] Range of relative wind farm efficiency gains was obtained from *Table 1*, and corresponds to different wind speeds and direction deviations from perfect wind-alignment [86].

Table A.5: Summary of studies that have developed wind farm controllers based on the concept of yaw-based wake redirection.

Ref.	**Control method**	**Inputs**	**Control model**	**Evaluation model**	**Simulated wind conditions**	**Relative efficiency gain**
[147]	Centralized cooperative game-theoretic control	a, γ	Steady wake model	Steady wake model	Const. speed	25.51 %[91]
[145]	Centralized cooperative game-theoretic control	a, γ	Steady wake model	Steady wake model	Const. speed	7.14 %[92]
[146]	Centralized/distributed cooperative game-theoretic control	a, γ	Gaussian process learning	Steady wake model	Const. speed	24.5 %[93]
[69]	Nonlinear model predictive control using extensive grid search	γ	FLORIDyn	FLORIDyn	Const. speed	0.19 %[94]
[150]	Predefined optimal yaw angles + PI wind turbine wake steering controller + Gain scheduled PI wind turbine power controller	γ	No wind farm controller	Dynamic modified SimWindFarm	Const. speed Turbulence	4.5 %[95]
[27]	Lookup table of optimal set-points + Turbine yaw controller	a, γ	Steady wake model (similar to FLORIS model)	Dynamic wake model (similar to SimWindFarm)	Const. speed Var. direction Turbulence	-[96]

[91] Relative wind farm efficiency gain corresponds to the highest percent gain in wind farm power output across different wind directions, and was obtained by digitizing *Figure 7* [147].

[92] Relative wind farm efficiency gain was obtained from discussions in *Section 6.3* [145].

[93] Relative wind farm efficiency gain corresponds to a distributed optimization scheme, and was obtained by digitizing *Figure 13c* and calculating the final rise in mean power efficiency relative to its initial value [146].

[94] Relative wind farm energy production gain was obtained from discussions in *Section IV* [69].

[95] Relative wind farm efficiency gain was obtained from discussions in *Section VI.B* [150].

[96] It was not possible to obtain relative efficiency gain data from this study; however, relative energy production gain over three hours of simulation

Table A.5 continued from previous page

Ref.	**Control method**	**Inputs**	**Control model**	**Evaluation model**	**Simulated wind conditions**	**Relative efficiency gain**
[72][97]	Centralized game-theoretic adaptive control	γ	Steady FLORIS model	SOWFA (dynamic CFD)	Const. speed Turbulence Wind shear	13.03 %[98]
[139]	Centralized model predictive control	C_T, $\dot{\gamma}$	Dynamic LES	Dynamic LES	Const. speed Turbulence Wind shear	34.48 %[99]
[137]	Centralized model predictive control	C_T, $\dot{\gamma}$	Dynamic LES	Dynamic LES	Const. speed Turbulence Wind shear	46 %[100]
[148]	Centralized cooperative game-theoretic control	T_g, β, γ	Gaussian process learning	Scaled wind farm	-	33.22 %[101]
[32]	Distributed extremum seeking control + PI wind turbine wake steering controller + Gain scheduled PI wind turbine power controller	γ	Model-free	Scaled wind farm	-	20 %[102]

was 2.06 % [27].

[97] Additional information is available in companion publications [66, 68].

[98] Relative wind farm efficiency gain was obtained from *Table II* [72].

[99] Relative wind farm efficiency gain was achieved using combined yaw-based wake redirection and power de-rating wind farm control methods, and corresponds to turbulent inflow wind. The result was obtained from *Figure 12b* [139].

[100] Relative wind farm efficiency gain corresponds to the gridded layout with a wide wind turbine spacing of six rotor diameters, and was obtained from *Table 1* [137].

[101] Relative wind farm efficiency gain was obtained from *Figure 14* [148].

[102] Relative wind farm efficiency gain was obtained by digitizing the right plot in *Figure 5* and averaging all data points past 200 sec [32].

Table A.5 continued from previous page

Ref.	Control method	Inputs	Control model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[59]	Lookup table for optimal yaw settings + Turbine yaw offset controller	γ	Steady FLORIS model	Longyuan Rudong Chaojiandai wind farm	-	$5.56\,\%$[103] -12.16 to $19.96\,\%$[104]
[61]	Lookup table for optimal yaw settings + Turbine yaw offset controller	γ	Steady FLORIS model	1×2 commercial wind farm	-	-7.9 to $10.2\,\%$[105]

[103] Relative wind farm efficiency gain corresponds to a scenario in which the wind is aligned with turbines C1 and D1, and was obtained by digitizing data from *Figure 6c* [59]. This figure was selected for analysis since the turbine spacing between turbines C1 and D1 was seven rotor diameters.

[104] Range of relative wind farm efficiency gains corresponds to the entire range of wind directions presented in *Figure 6c* [59].

[105] Range of relative wind farm efficiencies was obtained by digitizing *Figure 14*, and corresponds to different wind directions.

Table A.6: Summary of studies that have assessed the steady-state potential of turbine repositioning.

Ref.	Optimization/analysis method	Inputs	Optimization model	Evaluation model	Simulated wind conditions	Relative efficiency gain
[153]	Centralized covariance matrix adaption evolutionary strategy	x, y	Steady Park model	FamFlow (steady CFD)	Const. speed	−11.49 to 18.11 %[106] −4.04 to 4.77 %[107]
[58][108]	Numerical simulation	y	No optimization	SOWFA (dynamic CFD)	Const. speed Turbulence Wind shear	41.0 %[109]
[101]	Centralized sequential quadratic programming	a, γ	FOWFSim	FOWFSim	Const. speed	16.5 to 53.0 %[110]

[106] Range of relative wind farm efficiency gains corresponds to turbine relocation within a suboptimal gridded wind farm layout, and was obtained by digitizing *Figure 12b* [153].

[107] Range of relative wind farm efficiency gains corresponds to turbine relocation within an optimized wind farm layout, and was obtained by digitizing *Figure 12b* [153].

[108] Additional information is available in companion publications [62, 66].

[109] Relative wind farm efficiency gain corresponds to lateral position control, and was obtained from *Figure 4* [58].

[110] Range of relative wind farm efficiency gains corresponds to different mooring line lengths, and was obtained from *Figure 3* and discussions in *Section IV.a* [101].