

# Modeling for Wind Farm Control

A THESIS  
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL  
OF THE UNIVERSITY OF MINNESOTA  
BY

Jennifer Annoni

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY

Peter J. Seiler, advisor

May 2016

© Jennifer Annoni 2016

## Acknowledgments

This has been an incredible journey in my life and I could not have done this alone. I want to take this time to thank everyone who has been a part of this endeavor. First and foremost, I would like to thank my advisor, Pete Seiler. Without you, none of this would have happened or been possible. You believed in me from the start and I am forever grateful for the time you spent advising me on everything from controls, to wind energy, to running, and beyond. I hope that one day I am able to give back even a small fraction of what you have given me.

I'd also like to thank my committee members, Mihailo Jovanovic, Joseph Nichols, and Ellen Longmire. I would like to thank you for guiding and collaborating with me and for making me a well-rounded research scientist. I'd like to thank my labmates of 309 for our discussions that ranged from science to football. I'd also like to thank the people at the National Renewable Energy Laboratory especially Katie, Paul, and Pieter. My visits to Colorado have been very influential on my research and career decisions. Ryan, thank you for your support, good discussions (especially about adjoints), and adventures over the past year. I would like to give a shout out to my volleyball team (Al, Bri, Liz, Kim, and Beth). You guys have kept me sane, especially this last year.

Lastly, I would like to thank my family for their support including helping me move to and from Colorado, bringing me cookies when you knew I had to work late nights, and keeping things in perspective. To my sister Beth, I would like to personally thank you for everything over the last four years including but not limited to all of our science meetings at the tea garden, making me laugh, and listening to all of my presentations such that you could probably have given them yourself.

It has been a wild ride. Here's to changing the world and Go Packers!

# Dedication

*To My Family*

## **Abstract**

The focus of this thesis is to improve the economic viability of wind energy and help integrate wind into the electric system. Wind energy plays a key role in meeting the renewable energy demands in the United States. Currently, wind farms experience a significant loss of power production due to the interactions between wind turbines when their individual performance is maximized. The main technical goal of this research is to develop techniques to obtain simplified models that will be used to properly coordinate wind turbines for more efficient operation of wind farms.

# Contents

|  |           |
|--|-----------|
| <b>List of Figures</b>                               | <b>ix</b> |
| <b>Chapter 1 Introduction</b>                        | <b>1</b>  |
| <b>Chapter 2 Wind Farm Background</b>                | <b>4</b>  |
| 2.1 Introduction . . . . .                           | 4         |
| 2.2 Single Turbine Control . . . . .                 | 4         |
| 2.3 Turbine Modeling . . . . .                       | 8         |
| 2.4 Aerodynamics of Wind Turbines . . . . .          | 10        |
| 2.5 Wind Farm Control . . . . .                      | 12        |
| 2.5.1 Wake Characteristics . . . . .                 | 12        |
| 2.5.2 Two-Turbine Coordination . . . . .             | 14        |
| 2.6 Wake Models . . . . .                            | 16        |
| 2.6.1 Static Park Model . . . . .                    | 17        |
| 2.6.2 Dynamic Wake Meandering Model . . . . .        | 18        |
| 2.6.3 Actuator Disk Model . . . . .                  | 19        |
| 2.6.4 Simulator fOr Wind Farm Applications . . . . . | 20        |
| 2.7 Comparison of Wake Models . . . . .              | 21        |
| 2.7.1 Maximizing Power with Each Model . . . . .     | 23        |

|                  |  |           |
|------------------|--|-----------|
| 2.7.2            | Evaluation of Axial Induction Control using SOWFA . . . . .        | 24        |
| 2.8              | Conclusions . . . . .  | 28        |
| <b>Chapter 3</b> | <b>Data-Driven Modeling using Wind Tunnel Experiments</b>          | <b>30</b> |
| 3.1              | Introduction . . . . .   | 30        |
| 3.2              | Dynamic Park Model . . . . .                                       | 31        |
| 3.2.1            | Turbine Dynamics . . . . .   | 31        |
| 3.2.2            | Wake Transport . . . . .   | 33        |
| 3.2.3            | Linearized Model . . . . .   | 33        |
| 3.3              | Experimental Setup . . . . .                                       | 34        |
| 3.3.1            | Wind Tunnel . . . . .  | 34        |
| 3.3.2            | Turbine Model . . . . .  | 35        |
| 3.3.3            | Three Turbine Setup . . . . .                                      | 35        |
| 3.4              | Results . . . . .  | 38        |
| 3.4.1            | Wake Characteristics - PIV . . . . .                               | 38        |
| 3.4.2            | Propagation Dynamics - Voltage Measurements . . . . .              | 41        |
| 3.4.3            | Example Model Application: Wind Farm Control . . . . .             | 45        |
| 3.5              | Conclusions . . . . .  | 47        |
| <b>Chapter 4</b> | <b>Reduced-Order Modeling</b>                                      | <b>48</b> |
| 4.1              | Introduction . . . . .   | 48        |
| 4.2              | Criteria . . . . .   | 49        |
| 4.3              | Balanced Truncation . . . . .                                      | 50        |
| 4.4              | System Identification . . . . .                                    | 51        |
| 4.4.1            | Numerical Algorithm for Subspace State Space System Identification | 51        |
| 4.4.2            | Eigensystem Realization Algorithm . . . . .                        | 52        |

|                  |   |           |
|------------------|---|-----------|
| 4.4.3            | Linear Parameter Varying Subspace Identification . . . . .      | 54        |
| 4.5              | Proper Orthogonal Decomposition . . . . .                       | 54        |
| 4.6              | Balanced Proper Orthogonal Decomposition . . . . .              | 56        |
| 4.7              | Dynamic Mode Decomposition . . . . .                            | 57        |
| 4.8              | Input-Output Reduced-Order Modeling . . . . .                   | 59        |
| 4.9              | Impact of Process Noise in Identifying Systems . . . . .        | 62        |
| 4.10             | Analyzing the DMD modes . . . . .                               | 64        |
| 4.10.1           | Optimal Amplitudes of DMD modes . . . . .                       | 65        |
| 4.10.2           | Sparsity-Promoting DMD for Systems with Inputs . . . . .        | 66        |
| 4.10.3           | Alternating Direction Method of Multipliers Algorithm . . . . . | 67        |
| 4.11             | Example: Linearized Channel Flow . . . . .                      | 68        |
| 4.11.1           | DMD vs. DMD with Exogenous Inputs . . . . .                     | 68        |
| 4.11.2           | Sparsity-Promoting DMD for Systems with Inputs . . . . .        | 71        |
| 4.12             | Conclusions . . . . .   | 73        |
| <b>Chapter 5</b> | <b>Parameter Varying Reduced-Order Modeling</b>                 | <b>74</b> |
| 5.1              | Introduction . . . . .  | 74        |
| 5.2              | Linearization for Parameter Varying Systems . . . . .           | 75        |
| 5.3              | State Consistency Issue . . . . .                               | 76        |
| 5.4              | Reduced-Order Parameter Varying Linearizations . . . . .        | 77        |
| 5.5              | Example: Nonlinear Mass-Spring-Damper . . . . .                 | 80        |
| 5.5.1            | Model Formulation: 1-Block . . . . .                            | 80        |
| 5.5.2            | Model Formulation: M-Blocks . . . . .                           | 82        |
| 5.5.3            | IOROM Method: Single Operating Point . . . . .                  | 83        |
| 5.5.4            | Parameter Varying IOROM Method . . . . .                        | 84        |

|                  |   |            |
|------------------|---|------------|
| 5.6              | Conclusion . . . . .  | 86         |
| 5.7              | Appendix . . . . .  | 86         |
| <b>Chapter 6</b> | <b>Wind Farm Application: Actuator Disk Model</b>               | <b>88</b>  |
| 6.1              | Introduction . . . . .  | 88         |
| 6.2              | Nonlinear Governing Equations . . . . .                         | 88         |
| 6.3              | Linearized Equations . . . . .                                  | 91         |
| 6.3.1            | Linear Outputs . . . . .  | 91         |
| 6.3.2            | Nonlinear Outputs . . . . .                                     | 92         |
| 6.4              | Example: Two Turbine Array . . . . .                            | 92         |
| 6.4.1            | Setup . . . . .   | 92         |
| 6.4.2            | IOROM Method: Single Operating Point . . . . .                  | 93         |
| 6.4.2.1          | Model-Based Controller Design Example . . . . .                 | 96         |
| 6.4.3            | IOROM Parameter Varying Method . . . . .                        | 98         |
| 6.5              | Conclusions . . . . .   | 100        |
| <b>Chapter 7</b> | <b>Extension to Large Eddy Simulations and Wind Tunnel Ex-</b>  | <b>102</b> |
|                  | <b>periments</b>  |            |
| 7.1              | Introduction . . . . .  | 102        |
| 7.2              | Input-Output Reduced-Order Model with a Kalman Filter . . . . . | 102        |
| 7.3              | Large Eddy Simulations . . . . .                                | 104        |
| 7.3.1            | Two-Turbine Setup . . . . .                                     | 105        |
| 7.3.2            | IOROM with SOWFA . . . . .                                      | 105        |
| 7.3.3            | Results . . . . .   | 108        |
| 7.4              | Wind Tunnel Experiments . . . . .                               | 112        |
| 7.4.1            | Setup . . . . .   | 113        |

|                  |   |            |
|------------------|---|------------|
| 7.4.2            | IOROM with the Wind Tunnel . . . . .            | 115        |
| 7.4.3            | Results . . . . .                               | 115        |
| 7.5              | Conclusions . . . . .                           | 117        |
| 7.6              | Appendix:Tall-Skinny QR Factorization . . . . . | 117        |
| <b>Chapter 8</b> | <b>Conclusions</b>                              | <b>121</b> |
|                  | <b>Bibliography</b>                             | <b>124</b> |

# List of Figures

|      |  |    |
|------|--|----|
| 2.1  | 2.5 MW Clipper research turbine located at UMore Park, Rosemount, MN.  | 5  |
| 2.2  | Three-dimensional (left) and two-dimensional (right) normalized $C_P$ curve of the 2.5 MW Clipper Turbine at UMore Park, Minnesota. . . . .                                    | 6  |
| 2.3  | Regions of wind turbine operation. The blue curve represents the power in the wind and the green curve represents the power captured by the turbine.                           | 6  |
| 2.4  | Block diagram of a standard generator torque controller. . . . .   | 7  |
| 2.5  | Block diagram of a standard blade pitch controller. . . . .  | 7  |
| 2.6  | Streamtube Control Volume of a Turbine. . . . .  | 9  |
| 2.7  | $C_P$ vs. $C_T$ of an ideal turbine. . . . .   | 10 |
| 2.8  | Blade element momentum theory. The effective wind speed impacting the turbine blades is a function of the freestream velocity, axial induction, and blade pitch angle. . . . . | 11 |
| 2.9  | Blade element momentum theory. The lift and drag forces on the turbine blades are shown in this diagram. . . . .   | 11 |
| 2.10 | Aerodynamic interactions in the Horns Rev wind farm. Photo credits: Vattenfall Wind Power, Denmark. . . . .  | 13 |
| 2.11 | Distinction between different regions in the wake. Note that the red lines in the dotted lines indicate the magnitude of the velocity in that region. . . .                    | 13 |
| 2.12 | Two-turbine setup for evaluating axial induction control using various wake models. . . . .  | 15 |
| 2.13 | Setup for deriving the Park model. . . . .   | 17 |

|      |  |    |
|------|--|----|
| 2.14 | One turbine (left) and two turbine (right) mean streamwise velocity computed using SOWFA. . . . .  | 20 |
| 2.15 | (Left) Streamwise velocity component downstream of Turbine 1. (Right) Comparison of the streamwise velocity of the upstream turbine wake using the Park model and SOWFA. . . . .   | 22 |
| 2.16 | Comparison of power (top), power variance (middle), and tower loads (bottom) generated from each wake model. . . . .   | 23 |
| 2.17 | Comparison of potential percentage of power gained using axial induction control with different wake models. . . . .   | 25 |
| 2.18 | SOWFA simulation results using the axial induction control concepts on a two-turbine setup, using pitch (left) or generator torque (right) offsets on the front turbine to affect the power on the downstream turbine. . . . .   | 26 |
| 2.19 | Kinetic power added to the flow behind the turbine by introducing a 2° pitch offset. The black circle visualizes the location of the second turbine rotor. . . . .   | 27 |
| 2.20 | Energy balance based on the results shown in Figure 2.20: power lost on the upstream turbine by introducing a 2° pitch offset, $\Delta P_{T1}$ (red line), compared to flow power added in a virtual downstream rotor plane, $\Delta P_{\text{wind}}$ (black bar), and the maximum power that can be recovered by the downstream turbine, $C_{P,\text{max}}\Delta P_{\text{wind}}$ (gray bar). . . . . | 28 |
| 3.1  | Block diagram of the dynamic Park model. . . . .   | 31 |
| 3.2  | Two turbine setup in the SAFL wind tunnel (left) and the 2.5 MW Clipper turbine at UMore Park (right). Photo credit: Kevin Howard. . . . .   | 36 |
| 3.3  | Three turbine setup aligned. Turbine 1 is the most upstream turbine, Turbine 2 is the middle turbine, and Turbine 3 is the most downstream turbine. . . . .  | 37 |
| 3.4  | (Left) Rated mean streamwise velocity behind Turbine 1. (Right) Derated mean streamwise velocity behind Turbine 1. . . . .   | 39 |
| 3.5  | (Left) Wake expansion behind Turbine 1 in the rated case. Note, the white dots mark the outside of the wake. (Right) Wake expansion in the derated case. . . . .   | 39 |
| 3.6  | Time averaged spanwise velocity profile from PIV experiments compared to the spanwise profile obtained from the Park model . . . . .   | 40 |

|      |   |    |
|------|---|----|
| 3.7  | Centerline velocity profile from Turbine 1 to Turbine 3. The vertical black line indicates where the second turbine is located. . . . .   | 40 |
| 3.8  | Turbulent kinetic energy behind Turbine 1 under two operating conditions: rated (left) and derated (right). . . . .   | 41 |
| 3.9  | Voltage time series at 0.07 Hz (first column), 0.2 Hz (second column), and 2 Hz (third column). The voltages have been normalized by the voltage at Turbine 1. Note that the signals for Turbine 2 (middle row) and Turbine 3 (bottom row) have been scaled by a factor of 10 or 100 for visualization purposes. . . . .  | 42 |
| 3.10 | Input/output dynamics in the three turbine setup compared to the dynamic Park model. . . . .  | 43 |
| 3.11 | PI control implemented using the dynamic Park model. . . . .  | 45 |
| 3.12 | Results of implementing a PI controller with the dynamic Park model. This is compared to the results from the wind tunnel with no control. Note $\tau_{g1}$ is in terms of voltage applied. . . . .   | 46 |
| 4.1  | Depiction of the channel flow problem. . . . .  | 69 |
| 4.2  | (Left) Eigenvalues obtained with standard DMD and the IOROM approach. (Right) Eigenvalues in the presence of noise with variance $10^{-11}$ . . . . .   | 69 |
| 4.3  | This figure shows the sensitivity of the eigenvalues, $\delta\lambda$ , using the DMD and IOROM approaches for the channel flow problem. The values plotted are the upper bound. . . . .  | 70 |
| 4.4  | (Left) Average eigenvalues identified by DMD and the IOROM approach after 10 simulations. (Right) Average eigenvalues for 100 simulations. . . . .  | 71 |
| 4.5  | (Left) This figure shows the average maximum error between entries in the full-order $A$ matrix and the $A$ matrices identified by the DMD (blue) and the IOROM approach (red). (Right) This figure shows the average maximum variance between entries in the full-order $A$ matrix and the $A$ matrices identified by the DMD (blue) and the IOROM approach (red). . . . . | 71 |
| 4.6  | (Left) Performance loss with respect to the number of DMD modes selected. (Right) Eigenvalues that are omitted when using the sparsity promoting approach. This is shown for $N_z = 31$ . . . . .   | 72 |

|     |  |    |
|-----|--|----|
| 4.7 | (Left) Non-zero (Nz) $x_i$ vectors as $\gamma$ increases. The number of non-zero vectors correspond to the number of DMD modes retained. (Right) Performance loss as $\gamma$ increases. . . . .   | 73 |
| 5.1 | Nonlinear mass-spring-damper (Left) and system constants (Right). . . . .  | 81 |
| 5.2 | System response vs. time (left) and vs. parameter (right) . . . . .  | 82 |
| 5.3 | IOROM Results: POD/Fit Relative Errors (left) and Bode Plot (right). Note that the last block, $x(100)$ , also corresponds to the output of the system. . . . .  | 84 |
| 5.4 | IOROM Results: Output vs. Time (left) and States vs. Time (right) . . . . .  | 85 |
| 5.5 | LPV IOROM Results: Output vs. Time (left) and States vs. Time (right). Again, note that the last block, $x(100)$ , also corresponds to the output of the system. . . . .   | 86 |
| 6.1 | (Left) Mean streamwise velocity computed using the actuator disk model for a $2 \times 3$ wind farm with 4D spacing in the y direction and 4D spacing in the x direction. (Right) Mean spanwise velocity computed using the actuator disk model for a $2 \times 3$ wind farm with 4D spacing in the y direction and 4D spacing in the x direction. . . . . | 90 |
| 6.2 | Two-turbine setup (left) and the corresponding baseflow (right) for the low Re flow specified. It should be noted that the flow is depicted as if you are looking at the two-turbine array from above. . . . .   | 93 |
| 6.3 | Model fitting error computed for the actuator disk example . . . . .   | 94 |
| 6.4 | IOROM Results of the actuator disk example. The top plot shows the flow field reconstructed using 20 modes. The middle plot shows the full-order nonlinear flow. The bottom plot shows the output (indicated by the triangle in the top and middle plot) of the reduced-order model compared to the full-order model. . . . .                              | 95 |
| 6.5 | This plot shows the output of the reduced-order models constructed using ERA and the IOROM approach proposed in this thesis. The output for this particular example is the velocity fluctuations at the measurement point 3D downstream of the second turbine. . . . .   | 96 |
| 6.6 | Time domain output of the actuator disk flow with and without the LQG controller designed using the IOROM. . . . .   | 97 |

|     |  |     |
|-----|--|-----|
| 6.7 | Time domain signal of the Reynolds number, input, and output. The output of the reduced-order LPV model is compared to the full-order nonlinear simulation. . . . .  | 99  |
| 6.8 | Flow Visualization at 5 different snapshots in the flow for the LPV formulation. These snapshots correspond to the same example in Figure 6.7. Note: the white lines at the turbine locations indicate the different thrust values on the turbines. As noted previously, the formulation of the actuator disk example introduces an asymmetric forcing on the turbine to induce wake meandering. . . . . | 101 |
| 7.1 | (Left) Setup for the two-turbine array in SOWFA. (Right) Forced input used for the two-turbine array. . . . .  | 106 |
| 7.2 | Frequency content of the flow velocity field at 4D downstream of the upstream turbine. Specifically, this shows the -20dB bandwidth of the fast Fourier transform of the velocity signal for sample points at 4D downstream. Note that the velocity signal was sampled at 1Hz. The pure yellow may indicate that the frequency at these points exceeds 0.5Hz. . . . .                                    | 107 |
| 7.3 | POD modes 1, 2, 10, 20, 50, and 100 of the SOWFA simulation. . . . .   | 108 |
| 7.4 | (Left) Percent energy of each mode in the SOWFA simulation. (Right) Model error computed based on the number of modes used to identify a model of the system. . . . .  | 109 |
| 7.5 | Flow reconstructed using the reduced-order model (top) and compared to SOWFA (second plot). The third and fourth plot show the inputs at the upstream turbine. Lastly, the bottom plot shows the output of the reduced-order model compared to the output of SOWFA. . . . .  | 110 |
| 7.6 | The model was validated using a different data set. The corresponding results are shown here. Flow reconstructed using the reduced-order model (top) and compared to SOWFA (second plot). The third and fourth plot show the inputs at the upstream turbine. Lastly, the bottom plot shows the output of the reduced-order model compared to the output of SOWFA. . . . .                                | 111 |

|      |   |     |
|------|---|-----|
| 7.7  | This figure shows an example of a IOROM that was overfit to the original data. The model used in this example had 300 states. When trying to use this IOROM (along with a Kalman filter) with a different data set, the input-output relationship is retained (bottom plot) but the reduced-order states are no longer able to represent the full-order spatial dynamics of the simulation. | 112 |
| 7.8  | (Left) PIV setup in the wind tunnel. (Right) Forced input and corresponding output (streamwise velocity 3D downstream).   | 114 |
| 7.9  | POD modes of the PIV wind tunnel experiments.   | 116 |
| 7.10 | Energy of each POD mode from the wind tunnel data.  | 117 |
| 7.11 | Flow reconstructed using the reduced-order model (top) and compared to the PIV results (second plot). The third plot shows the input voltage at the upstream turbine and the bottom plot shows the wind speed output of the reduced-order model compared to the PIV results.  | 118 |
| 7.12 | The model was validated using a different data set. The corresponding results are shown on the right. Flow reconstructed using the reduced-order model (top) and compared to the PIV results (second plot). The third plot shows the input voltage at the upstream turbine and the bottom plot shows the wind speed output of the reduced-order model compared to the PIV results.          | 119 |

## Nomenclature

### Abbreviations and Acronyms

|              |  |
|--------------|--|
| <i>CFD</i>   | Computational Fluid Dynamics                                       |
| <i>RANS</i>  | Reynolds Averaged Navier Stokes                                    |
| <i>DMD</i>   | Dynamic Mode Decomposition   |
| <i>POD</i>   | Proper Orthogonal Decomposition                                    |
| <i>DMDc</i>  | Dynamic Mode Decomposition with Control                            |
| <i>TSR</i>   | Tip-Speed Ratio  |
| <i>FAST</i>  | Fatigue, Aerodynamics, Structures, and Turbulence                  |
| <i>NREL</i>  | National Renewable Energy Laboratory                               |
| <i>DWM</i>   | Dynamic Wake Meandering  |
| <i>SOWFA</i> | Simulator fOr Wind Farm Applications                               |
| <i>VWiS</i>  | Virtual Wind Simulator   |
| <i>MSI</i>   | Minnesota Supercomputing Institute                                 |
| <i>SAFL</i>  | Saint Anthony Falls Laboratory                                     |
| <i>PIV</i>   | Particle Image Velocimetry   |
| <i>TKE</i>   | Turbulent Kinetic Energy   |
| <i>PI</i>    | Proportional-Integral  |
| <i>PID</i>   | Proportional-Integral-Derivative                                   |
| <i>NASID</i> | Numerical Algorithm for Subspace State Space System Identification |
| <i>ERA</i>   | Eigensystem Realization Algorithm                                  |
| <i>SVD</i>   | Singular Value Decomposition                                       |
| <i>LPV</i>   | Linear Parameter Varying   |
| <i>BPOD</i>  | Balanced Proper Orthogonal Decomposition                           |
| <i>IOROM</i> | Input-Output Reduced-Order Model                                   |
| <i>PL</i>    | Performance Loss   |
| <i>LiDAR</i> | Light Detection and Ranging  |
| <i>LQG</i>   | Linear Quadratic Gaussian  |
| <i>LQR</i>   | Linear Quadratic Regulator   |
| <i>PRBS</i>  | Pseudo-Random Binary Sequence                                      |

### List of Symbols

#### Wind Energy

|          |                               |
|----------|-------------------------------|
| $\tau_g$ | generator torque, Nm          |
| $\beta$  | blade pitch angle, deg or rad |
| $\omega$ | rotor speed, rad/s            |

|                      |  |
|----------------------|--|
| $U$                  | wind speed perpendicular to the rotor plane, m/s |
| $P$                  | power, W   |
| $\rho$               | air density, kg/m <sup>3</sup>                   |
| $A$                  | rotor area, m <sup>2</sup>                       |
| $C_P$                | power coefficient, [-]                           |
| $\lambda$            | tip-speed ratio, [-]                             |
| $R$                  | radius, m  |
| $U_\infty$           | freestream velocity, m/s                         |
| $\dot{\omega}$       | angular acceleration, rad/s <sup>2</sup>         |
| $J$                  | rotor inertia, kg/m <sup>2</sup>                 |
| $\tau_{\text{aero}}$ | aerodynamic torque, Nm                           |
| $N$                  | gearbox ratio, [-]                               |
| $T$                  | thrust, N  |
| $a$                  | axial induction factor, [-]                      |
| $C_T$                | thrust coefficient, [-]                          |
| $U_T$                | tangential velocity, m/s                         |
| $W$                  | effective wind speed, m/s                        |
| $\alpha$             | angle of attack, deg or rad                      |
| $L_T$                | lift, N  |
| $D_T$                | drag, N  |
| $\sigma_r$           | blade solidity, [-]                              |
| $c$                  | chord length, m                                  |
| $B$                  | number of blades, [-]                            |
| $C_l$                | lift coefficient, [-]                            |
| $C_d$                | drag coefficient, [-]                            |
| $\lambda_r$          | local tip-speed ratio, [-]                       |
| $D$                  | diameter, m                                      |
| $\kappa$             | wake constant, [-]                               |
| $b$                  | wake half width, m                               |
| $u_c$                | center wake velocity, m/s                        |
| $\gamma$             | generator torque constant, [-]                   |
| $\tau_d$             | time delay, s                                    |
| $t$                  | time, s  |
| $\nu$                | kinematic viscosity, m/s <sup>2</sup>            |
| $\nu_T$              | eddy viscosity, m/s <sup>2</sup>                 |
| $L$                  | length of domain, m                              |
| $Re$                 | Reynolds number, [-]                             |

## Controls/Fluid Dynamics

|            |                                    |
|------------|------------------------------------|
| $u$        | input                              |
| $A$        | dynamic state matrix               |
| $B$        | input state matrix                 |
| $C$        | output state matrix                |
| $D$        | feedforward state matrix           |
| $y$        | output                             |
| $\omega$   | frequency                          |
| $x$        | states                             |
| $n_x$      | number of states                   |
| $n_u$      | number of inputs                   |
| $n_y$      | number of outputs                  |
| $n_s$      | number of snapshots                |
| $W_c$      | controllability Gramian            |
| $W_o$      | observability Gramian              |
| $\sigma$   | singular value                     |
| $\xi$      | oblique projection                 |
| $r$        | reduced-order                      |
| $v$        | adjoint variable                   |
| $\dagger$  | pseudo-inverse                     |
| $F$        | reduced-order dynamic state matrix |
| $G$        | reduced-order input state matrix   |
| $H$        | reduced-order output state matrix  |
| $z$        | reduced-order state                |
| $\varphi$  | POD modes                          |
| $\phi$     | DMD modes                          |
| $\lambda$  | lagrangian                         |
| $J$        | cost function                      |
| $\gamma$   | regularization parameter           |
| $X_1, X_0$ | states snapshot matrices           |
| $U_0$      | input snapshot matrix              |
| $Y_0$      | output snapshot matrix             |
| $\rho$     | time-varying parameter             |
| $Q$        | orthonormal subspace               |
| $u$        | streamwise velocity component      |
| $v$        | spanwise velocity component        |
| $w$        | vertical velocity component        |
| $P$        | pressure                           |

|          |                                     |
|----------|-------------------------------------|
| $x$      | x coordinate                        |
| $y$      | y coordinate                        |
| $z$      | z coordinate                        |
| $r$      | r coordinate                        |
| $dx$     | grid spacing in the x direction     |
| $dy$     | grid spacing in the y direction     |
| $N_x$    | number of points in the x direction |
| $N_y$    | number of points in the y direction |
| $\psi$   | streamfunction                      |
| $\Omega$ | vorticity                           |

# Chapter 1

## Introduction

In the United States, many states have a renewable portfolio standard or goal. For example, Minnesota has a renewable portfolio standard target of 25% renewable energy by 2025 [1]. Wind energy will be a significant factor in achieving this goal. Wind farm control can be used to increase wind energy efficiency by maximizing power in wind farms that are already installed. It can also be used to mitigate structural loads to maximize the lifetime of turbines and better integrate wind energy into the energy market.

Currently, turbines are controlled individually to maximize their own performance. Many studies have shown that operating all turbines in a wind farm at their optimal operating point leads to suboptimal performance of the overall wind farm [2–7]. Therefore, there is the potential to increase total power and reduce structural loads by properly coordinating the individual turbines in a wind farm [8]. An improved understanding of the aerodynamic interactions between turbines can aid in the design of enhanced control strategies that coordinate all turbines in a farm. Designing wind farm control strategies requires a model of the wind farm that has a low computational cost, but retains the dominant dynamics. A variety of wake models exist in literature that are useful for studying wind farm control. The simplest model is the Park model [9]. The Park model is static and provides a quick preliminary description of the wake interactions in a wind farm. Other approaches model the wake using the Reynolds Averaged Navier-Stokes (RANS) with a mixing length model (e.g. the eddy viscosity model [10]). Several high-fidelity computational fluid dynamics (CFD) models, e.g. large eddy simulations (LES), have been developed as well [11, 12]. These high-fidelity models are more accurate tools and can be used for evaluating wind farm controllers; however, they are computationally expensive. These low- and high-fidelity models have been used to evaluate wind farm control strategies. The analysis provides conflicting results based on the wake model chosen for control design. For example, control

strategies designed using simple static models may report significant improvements in wind farm performance, but an analysis of such control strategies using high-fidelity simulations can result in minimal to no improvements in wind farm performance. An example of a comparison of control predictions between a high-fidelity and simplified model is given in [13], where constant offsets of pitch and torque are used to change wake deficits. It is shown that extensions to the Park model are needed to match the results of high-fidelity models. Lastly, it is important to note that the variability of wind provides a challenge to the output of wind farms. The ability of the wake models to provide prediction and control capabilities is essential for improving the role of wind energy on the electric grid. Chapter 2 goes into detail on various wake models that are used in the wind farm controls literature.

Improving models for wind farm control requires a better understanding of the aerodynamic interactions in a wind farm. Although many optimization studies have been performed using static models and constant offsets of the operating point of the wind turbines, dynamic wake modeling and control approaches have been proposed recently. Previously proposed approaches use high-order first principle modeling by implementing the spatially filtered Navier-Stokes equations, e.g. [14], to arrive at a dynamic wake model. This thesis focuses on techniques to construct a reduced-order wake model for wind farm control from data generated by simulations or experiments.

Chapter 3 focuses on experiments conducted in a wind tunnel that help highlight the limitations of the static Park model and presents a dynamic Park model derived from experimental data. The Park model is compared to experiments that were done in the atmospheric boundary layer wind tunnel at the Saint Anthony Falls Laboratory, at the University of Minnesota, using a three turbine setup with model turbines, aligned in the wind direction. The results from the comparison are presented with some suggestions on how to improve the Park model to account for the flow dynamics.

Other techniques have been developed by the fluid dynamics and controls communities that are relevant for reduced-order wake modeling. Several of these methods are summarized in Chapter 4. These methods range from analytical reduced-order modeling, such as balanced truncation [15], to data-driven reduced-order modeling such as system identification [16]. These techniques identify a low-dimensional system to describe the dynamics of a high-dimensional system. Subspace techniques applied to balanced truncation can also perform model reduction on nonlinear systems [17]. In the fluid dynamics literature, proper orthogonal decomposition (POD) is a standard method where the state is projected onto a low-dimensional subspace of POD modes constructed using data from the high-order system [18–20]. Dynamic mode decomposition (DMD) is another data-driven approach that fits time-domain data with linear dynamics on a reduced-order subspace [21–23]. This approach has ties to the Koopman operator [24]. Both POD and DMD are, in their basic form,

for autonomous (unforced) systems. An extension of DMD was developed by [25] known as DMD with control (DMDc) to construct reduced-order models with control inputs. Specifically, this thesis develops a method that can handle many states (states  $> 10^5$ ), can handle inputs and outputs, does not require an adjoint, generates a model where the states have physical meaning, and can be used for parameter varying systems.

The main contribution of this thesis is an extension of DMDc used to construct reduced-order linear parameter varying (LPV) models that approximate a high-order nonlinear model, described in Chapter 5. The nonlinear system is assumed to have a parameterized collection of equilibrium operating points. For the wind farm example addressed in this thesis, the freestream wind speed parameterizes the equilibrium condition in the wind farm. The proposed approach involves two steps. First, POD and direct subspace identification are combined to construct an input-output reduced-order model (IOROM) [26]. Specifically, this extension of DMDc uses direct N4SID on a low-dimensional subspace, generated using POD, to construct a reduced-order linear model at one operating condition. Second, the reduced-order models constructed at fixed operating conditions are “stitched” together using a parameter varying linearization. The key technical issue is that the states of the reduced-order model must have a consistent meaning across all operating conditions. The approach used in the chapter handles the issue by constructing a single reduced-order subspace that is used at all operating conditions. This approach and the LPV linearization method are demonstrated on a medium-fidelity wind farm control example in Chapter 6. Finally, this approach can be extended to higher-fidelity models and simulations as is illustrated in Chapter 7.

## Chapter 2

# Wind Farm Background

### 2.1 Introduction

This chapter provides the background information for the wind farm control problem. In particular, this chapter begins with single turbine control and single turbine modeling in Section 2.2 and 2.3. The modeling of single turbine aerodynamics is critical to providing realistic insights into wind farm fluid dynamics. Specifically, the aerodynamics of individual wind turbines are reviewed in Section 2.4. Section 2.5 relates the single turbine modeling and aerodynamics to the wind farm control problem.

An improved understanding of the aerodynamic interactions within a wind farm is needed to develop dynamic models that can be used for wind farm control. A variety of wake models are currently used in literature ranging from low-fidelity quasi-static models to high-fidelity computational fluid dynamic models (Section 2.6). These tools are compared, in Section 2.7, by evaluating the power, loads, and flow characteristics for the coordinated two-turbine array. The results presented in this chapter highlight the advantages and disadvantages of existing wake models for design and analysis of coordinated wind farm controllers.

### 2.2 Single Turbine Control

This section reviews the operation and control of a single turbine. The overall goal of a utility-scale turbine is to maximize individual power production and/or mitigate structural loads, depending on the wind speed. The University of Minnesota has a 2.5 MW Clipper Research turbine, Figure 2.1, which is referenced throughout this thesis. Utility-scale turbines typically have several inputs that can be controlled to increase the captured power and reduce structural loads. These inputs include generator torque,  $\tau_g$ , and blade pitch angle,  $\beta$  to control the rotor speed,  $\omega$ , of the turbine at varying wind speeds,  $U$ . In general,



Figure 2.1: 2.5 MW Clipper research turbine located at UMore Park, Rosemount, MN.

the generator torque is varied at low wind speeds to maximize power captured [27]. At high wind speeds, the blade pitch angle is used to maintain the rated rotor speed and torque. This will aid in mitigating mechanical and electrical loads. The power captured by a single turbine can be expressed by

$$P = \frac{1}{2} \rho A U^3 C_P(\beta, \lambda), \quad (2.1)$$

where  $\rho$  [kg/m<sup>3</sup>] is the air density,  $A$  [m<sup>2</sup>] is the area swept by the rotor,  $U$  [m/s] is the wind speed perpendicular to the rotor plane,  $\lambda$  is the tip-speed ratio (TSR), and  $C_P$  is the power coefficient. The tip-speed ratio is defined as  $\frac{\omega R}{U_\infty}$ , where  $R$  [m] is the radius, and  $U_\infty$  [m/s] is the freestream velocity. The power coefficient is the fraction of available power in the wind captured by the turbine.  $C_P$  is a function of  $\beta$  [rad] and nondimensional  $\lambda$ . Figure 2.2 is the normalized  $C_P$  curve of the 2.5MW Clipper Turbine. The peak efficiency has been normalized for proprietary reasons.

The standard turbine controller, with  $\tau_g$  and  $\beta$  as inputs, can be split into 3 regions based on the wind speed, see Figure 2.3. In Region 1, the turbine does not produce any power because the wind speed is not sufficient to operate. Once the wind reaches the cut-in speed, the turbine enters Region 2. In this region, the turbine keeps the blades fixed at the optimal blade pitch angle,  $\beta_*$ , and maximizes the generator torque,  $\tau_g$ , to maximize the power of the turbine. Lastly, in Region 3, the generator torque reaches its rated torque and the turbine is producing its rated power. In this region, the power is held constant by fixing the generator torque,  $\tau_{g,rated}$  and adjusting the blade pitch angle. The blades are pitched

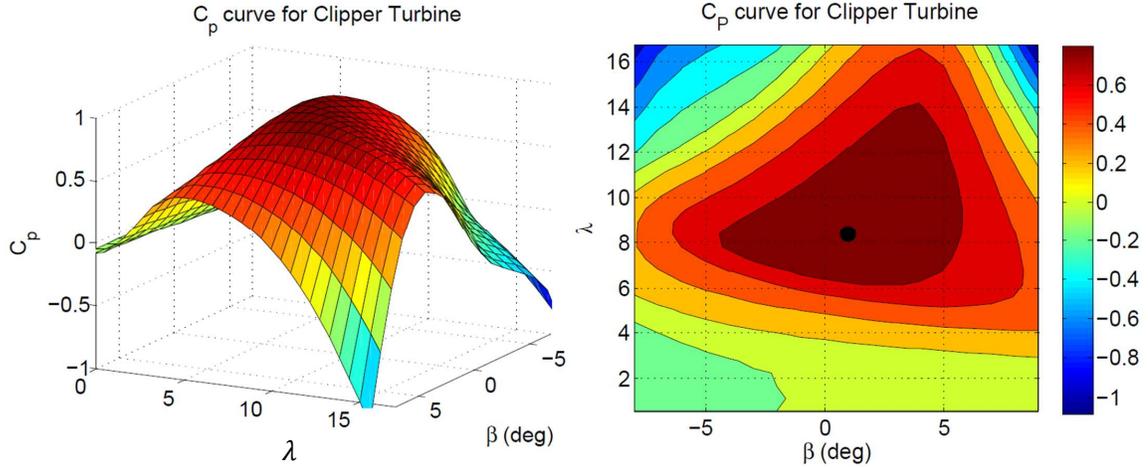


Figure 2.2: Three-dimensional (left) and two-dimensional (right) normalized  $C_P$  curve of the 2.5 MW Clipper Turbine at UMore Park, Minnesota.

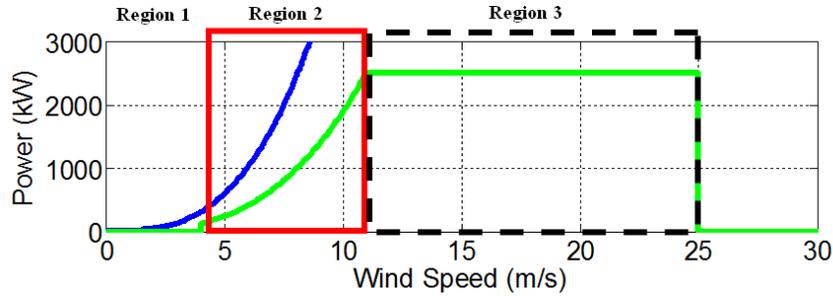


Figure 2.3: Regions of wind turbine operation. The blue curve represents the power in the wind and the green curve represents the power captured by the turbine.

to maintain the rated rotor speed and generator torque. This helps to minimize structural loads. The remainder of this section will address the specifics of the controller operations in each region.

In Region 2, the controller typically used is a generator torque controller. To develop this controller, the dynamics of the turbine are often modeled as a single degree-of-freedom rotational system:

$$\dot{\omega} = \frac{1}{J} (\tau_{\text{aero}} - \tau_g), \quad (2.2)$$

where  $\dot{\omega}$  [rad/s<sup>2</sup>] is the angular acceleration,  $J$  [kg/m<sup>2</sup>] is the rotor inertia,  $\tau_{\text{aero}}$  [Nm] is the aerodynamic torque. The power captured by a turbine can be expressed in terms of rotor speed by

$$P = \tau_{\text{aero}} \omega. \quad (2.3)$$

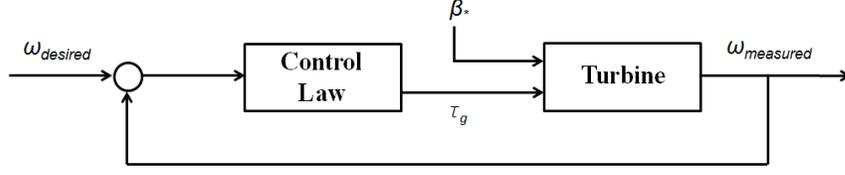


Figure 2.4: Block diagram of a standard generator torque controller.

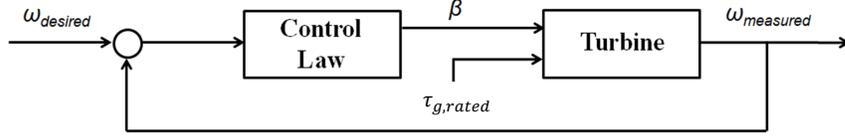


Figure 2.5: Block diagram of a standard blade pitch controller.

Using this relationship, the aerodynamic torque can be rewritten as

$$\tau_{\text{aero}} = \frac{P}{\omega} = \frac{\rho A U^3 C_P(\beta, \lambda)}{2\omega}. \quad (2.4)$$

As mentioned previously, the objective of a generator torque controller is to maximize power. This is done by maintaining an optimal blade pitch angle,  $\beta_*$ , and TSR,  $\lambda_*$ . The blade pitch angle is held fixed at  $\beta_*$ , and the generator torque is controlled to achieve  $\lambda_*$  in varying wind conditions. The block diagram of this controller is shown in Figure 2.4. The generator torque can be computed using the standard control law for wind turbines

$$\tau_g = K_g \omega^2, \quad (2.5)$$

where  $K_g = \frac{C_{P_*} \rho A R^5}{2\lambda_*^3 N}$  and  $N$  is the gearbox ratio. If  $K_g$  is chosen properly, the power from the turbine will converge to the optimal power in steady winds. In turbulent winds, the turbine will cycle around the peak  $\lambda_*$ . Substituting  $\tau_{\text{aero}}$  and  $\tau_g$  into (2.2), it can be shown that the turbine will converge toward the desired operating point,  $C_{P_*}(\beta_*, \lambda_*)$ .

In Region 3, the turbine controller holds the generator torque constant, at  $\tau_{g,\text{rated}}$  and pitches the blades to keep the rotor speed constant at its rated speed and helps minimize structural loads (see Figure 2.5). It is common to use a proportional-integral or proportional-integral-derivative controller for blade pitch control [27, 28]

$$\beta(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \dot{e}(t) \quad (2.6)$$

$$e(t) = \omega_{\text{rated}}(t) - \omega(t), \quad (2.7)$$

where  $K_p$ ,  $K_i$ , and  $K_d$  are constants that can be chosen based on the desired performance of the turbine in this region and  $e$  is the difference between the rated rotor speed,  $\omega_{\text{rated}}$ , and the actual rotor speed,  $\omega$ . The blade pitch controller computes the blade pitch that will minimize  $e$ . Additional details and references on single turbine operation can be found in [27–29].

A single turbine can be simulated using the Fatigue, Aerodynamics, Structures, and Turbulence (FAST) model developed by the National Renewable Energy Laboratory (NREL) [30]. FAST is a nonlinear simulation package that models the dominant structural modes for a wind turbine, e.g., tower and blade bending modes. In addition, the aerodynamic forces on the blade are modeled using blade element theory. FAST can determine the power production and loading characteristics experienced by a single turbine for a given wind profile. However, it does not include the capability to model the effect of the turbine on the airflow including downstream wakes.

### 2.3 Turbine Modeling

Oftentimes, when modeling a wind farm, a turbine is represented as a porous actuator disk having constant, radial, or variable loading. The advantage to using the actuator disk is that the blades of the turbine do not have to be modeled, which reduces the overall computation time. The maximum theoretical performance of a single turbine has been derived by Fredrick Lanchester, Albert Betz, and Nikolai Joukowsky in the early 1900s using an idealized actuator disk, i.e., independent of turbine design [31].

Consider a streamtube with an initial streamwise velocity,  $U_1$ , and a velocity behind the turbine,  $U_2$  (Figure 2.6). The turbine is represented by an actuator disk,  $S$ . If a turbine extracted 100% of the power out of the wind, the velocity behind the turbine would be 0 m/s, effectively blocking any wind from flowing through the turbine. Therefore, the velocity behind the turbine has to be nonzero for wind to flow through, but less than the initial velocity. The streamtube expands due to conservation of mass.

Using conservation of mass and momentum, it has been shown that the maximum power coefficient that a turbine can achieve is 0.593 or 59.3% of the power in the wind can be extracted by the turbine. See [29] for the full derivation. In practice, utility-scale turbines usually have a power coefficient around 0.4 to 0.5.

When using an actuator disk to represent a turbine, the axial induction factor,  $a$ , provides an input to describe the turbine operation. The induction factor is a measure of how much the wind slows down due to the action of the turbine. In particular, the induction factor for a single turbine is defined as  $a := 1 - \frac{U_1}{U_\infty}$ , where  $U_1$  denotes the average horizontal

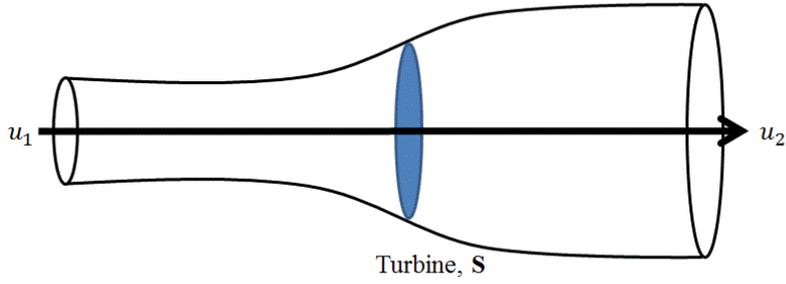


Figure 2.6: Streamtube Control Volume of a Turbine.

flow velocity across the rotor plane and  $U_\infty$  denotes the freestream velocity. The induction factor can be related to the power and thrust coefficient of a turbine. The axial thrust force (perpendicular to the rotor plane) is given by  $T = \frac{1}{2}\rho AU^2 C_T$  where  $C_T$  is the thrust coefficient. The power of the turbine is defined in (2.1). Specifically,  $C_T$  and  $C_P$  can be written as a function of the axial induction factor,  $a$  [29]

$$C_T(a) = 4a(1 - a), \quad (2.8)$$

$$C_P(a) = 4a(1 - a)^2, \quad (2.9)$$

As mentioned above, the maximum power coefficient is  $C_{P,\max} = 0.593$  achieved at an induction factor of  $a = \frac{1}{3}$ , see Figure 2.7. In practice, the induction factor can be related to the blade pitch angle and the generator torque, which are standard inputs to a utility-scale turbine [28,29]. The connection between the axial induction factor and the standard inputs to a utility-scale turbine, i.e., blade pitch angle and generator torque will be made in the next section.

The actuator line model is a second, more complex, turbine model [32]. This model takes finite sections of the rotating blade and calculates the airfoil lift and drag forces as they act on the flow (see Section 2.4 for more details). The lift and drag forces depend on the blade airfoil geometry and flow conditions. Nondimensional lift and drag are typically stored in look-up tables as a function of angle of attack between airflow and blade chord. This model can take considerably more computing time. Both the actuator disk and actuator line models can be used when modeling wind farms in computational fluid dynamic wake models.

This thesis addresses both the actuator disk and the actuator line representations of turbines when modeling wind farms. The next section will explore the connection between the

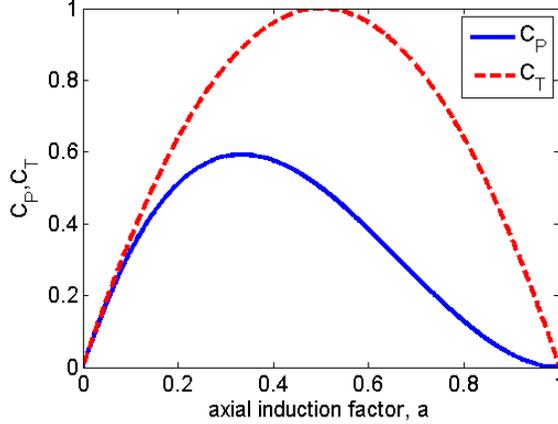


Figure 2.7:  $C_P$  vs.  $C_T$  of an ideal turbine.

parameters of the actuator disk model, i.e., the axial induction factor, and the parameters of the actuator line model, i.e., blade pitch angle and tip-speed ratio. Specifically, this indicates that a model (or controller) for wind farms that has turbines modeled as actuator disks can be compared to models (or controllers) for wind farms that have turbines modeled as actuator lines.

## 2.4 Aerodynamics of Wind Turbines

This section briefly describes the aerodynamic forces acting on a wind turbine and the resulting wind turbine characteristics. Further information regarding the aerodynamics of wind turbines can be found in [29].

Horizontal axis wind turbines are designed such that the rotor is perpendicular to the freestream wind,  $U_\infty$ . The velocity at the rotor,  $U$  is defined as  $U = U_\infty(1 - a)$  where  $a$  is the axial induction factor. The presence of the rotor decelerates the freestream flow as shown in Figure 2.6. The rotating blades contribute an additional tangential velocity component,  $U_T$ ,

$$U_T = \omega R. \quad (2.10)$$

The effective local wind speed is then computed as

$$W = \sqrt{(U_\infty(1 - a))^2 + U_T^2}. \quad (2.11)$$

The angle between the tangential velocity,  $U_T$  and  $U_\infty(1 - a)$  is  $\phi$  which is a combination of the angle of attack,  $\alpha$ , and blade pitch angle,  $\beta$ , seen in Figure 2.8. The angle of attack,  $\alpha$  can then be expressed as

$$\alpha = \phi - \beta. \quad (2.12)$$



The local lift force can be computed as:  $\delta L_T = \frac{1}{2}\rho W^2 c C_l \delta r$  where  $c$  is the chord length,  $C_l$  is the lift coefficient, and  $r$  is the local radius. Similarly, the local drag force can be computed as:  $\delta D_T = \frac{1}{2}\rho W^2 c C_d \delta r$  where  $C_d$  is the drag coefficient. The total thrust and total power can be computed by integrating the thrust and power over that entire length of the blade for each blade. By using the power and thrust equations, an equation for the axial induction factor can be derived in terms of the blade pitch angle,  $\beta$ , and the tip-speed ratio,  $\lambda$

$$\frac{a}{1-a} = \frac{\sigma_r}{4 \sin^2(\beta + \alpha)} (C_l \cos(\beta + \alpha) + C_d \sin(\beta + \alpha)), \quad (2.15)$$

where  $\sigma_r$  is the blade solidity that can be defined as:  $\frac{Bc\omega}{2\pi\lambda_r U_\infty}$  where  $B$  is the number of blades and  $\lambda_r$  is the local tip-speed ratio. The lift coefficient and the drag coefficient can be defined in terms of the angle of attack using thin airfoil theory [33]. Additional details on the relationship between the axial induction factor and the blade pitch angle and tip-speed ratio can be found in [29]. This section demonstrates that the axial induction factor used in the actuator disk turbine model described in Section 2.3 can be compared to parameters in the actuator line model or on a utility-scale turbine (i.e., blade pitch angle and tip-speed ratio). Lastly, it is important to understand the impact of the inputs of a turbine on the power and the thrust generated by an individual turbine as it has effects on the overall wake dynamics within a wind farm.

## 2.5 Wind Farm Control

As wind turbines are placed together in a wind farm, the aerodynamic interactions between wind turbines become important. This can be seen in Figure 2.10. This is a picture of the Horns Rev wind farm off the coast of Denmark. Specifically, this picture indicates that the wakes of the upstream turbines interfere with the performance of turbines operating downstream. This section addresses these aerodynamic interactions and a potential strategy for improving wind farm performance, i.e., maximizing power and mitigating structural loads within a wind farm.

### 2.5.1 Wake Characteristics

Individual turbine control has a significant impact on the flow dynamics in the wake. The wind turbine operation creates a trailing wake that is commonly divided into two regions [34]: the near wake and the far wake (see Figure 2.11). The near wake is roughly defined as the region directly downstream of the turbine where characteristics of the flow field are determined by the turbine geometry, i.e., the rotor and hub height, the blades, and the nacelle. The flow is driven by the incoming freestream velocity, a non-zero pressure gradient, and significant turbulence associated with the tip and trailing vortices enhanced



Figure 2.10: Aerodynamic interactions in the Horns Rev wind farm. Photo credits: Vattenfall Wind Power, Denmark.

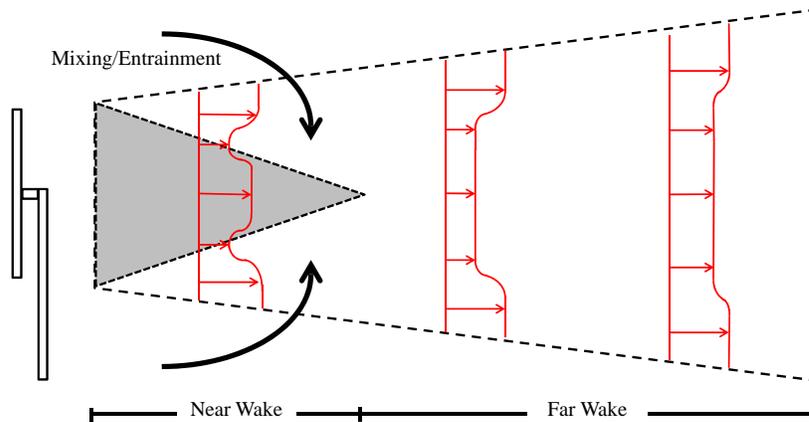


Figure 2.11: Distinction between different regions in the wake. Note that the red lines in the dotted lines indicate the magnitude of the velocity in that region.

by the shear layer produced by the separation of the flow at the blade edges. The near wake typically extends about 3 to 5 diameters downstream of the turbine. In the far wake, the pressure gradient becomes less significant. The wake is less dependent on the turbine geometry and more on topographic and thermal stability effects, in addition to any large-scale in-homogeneity of the incoming flow. In flat terrain, this region is approximately axisymmetric and self-similar making it easier to model [32]. The transition between the near and the far wake may be dependent on turbine controls. For more details, see [35].

One important flow characteristic in the far wake is the turbulence intensity. The turbulence generated in the wake can have a significant impact on the flow within a wind farm as it dictates much of the evolution of the wake as it propagates downstream. The turbulence in the wake is driven by the interaction between the ambient turbulence advected by the mean wind and the turbulence generated by the turbine. As the wind passes through the leading turbines, the flow separates from the blades and around the tower/nacelle and generates

a complex three-dimensional flow structure with elevated turbulent kinetic energy. The flow field also experiences a velocity deficit due to the thrust generated due to the power extraction by the turbine. This momentum deficit defines the wake, and a shear layer separates the wake from the freestream velocity. This shear layer contains vortices that entrain high momentum fluid, i.e., transports fluid momentum from outside the wake to inside the wake, causing the wake to expand and recover downstream of the turbine. On average the turbulence intensity increases as it passes through the next turbine downstream and the rate of entrainment increases the amount of turbulence and results in a faster recovery rate [36].

Another important wake characteristic is wake meandering. This is characterized as cyclical motions of the wake, particularly notable in the vertical and spanwise directions, which is caused by the interactions of large-scale turbulent structures in the wake [37, 38]. Wake meandering can have a significant impact on the structural loads of downstream turbines. However, wake meandering can help recover the power loss that is expected of turbine downstream in a wind farm by increasing mixing.

### 2.5.2 Two-Turbine Coordination

There is potential to increasing the performance of wind farms by better understanding the aerodynamics in a wind farm. It has been suggested by many studies [2–8] that operating turbines at their optimal operating point leads to suboptimal performance of the turbine. Control strategies for wind farm control include wake redirection [39, 40] and axial induction control [13]. Both aim at increasing total power and reducing structural loads in a wind farm. This thesis focuses on axial induction control.

Specifically, this section will formulate the axial induction control problem for a two-turbine array, shown in Figure 2.12. Let  $P_1$  and  $P_2$  denote the power from the upstream turbine and the downstream turbine, respectively. As described in Section 2.2, the power generated by the first turbine depends on the inflow wind speed as well as the blade pitch,  $\beta_1$ , and TSR,  $\lambda_1$ , for the turbine. The inflow speed for the first turbine is approximately equal to the free-stream velocity, i.e.  $u = U_\infty$ , hence the power generated by the upstream turbine can be expressed as  $P_1(\beta_1, \lambda_1, U_\infty)$ . The operation of the upstream turbine disturbs the flow and this impacts the operation of the downstream turbine. In this way, the downstream turbine depends on the blade pitch and TSR of the upstream turbine. Thus the averaged power generated by the downstream turbine has a functional form of  $P_2(\beta_1, \lambda_1, \beta_2, \lambda_2, U_\infty)$ . The precise relationship describing the aerodynamic coupling between the turbines depends on the model used for the near/far wake. The total power generated by the two-turbine

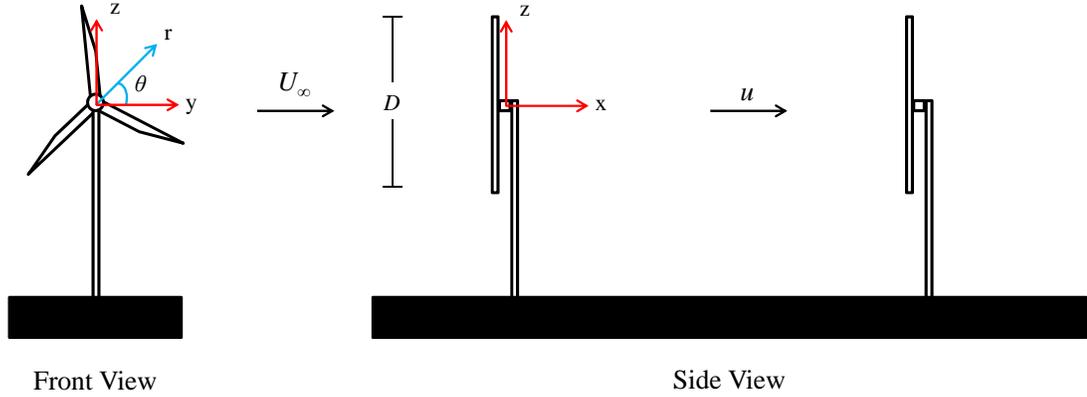


Figure 2.12: Two-turbine setup for evaluating axial induction control using various wake models.

array is thus given by

$$P_{\text{tot}}(\beta, \lambda, U_\infty) = P_1(\beta_1, \lambda_1, U_\infty) + P_2(\beta, \lambda, U_\infty), \quad (2.16)$$

where the vectors  $\beta := [\beta_1, \beta_2]^T$  and  $\lambda := [\lambda_1, \lambda_2]^T$  are defined to simplify the notation. The main objective of axial induction control is to maximize the total average power output

$$\max_{\beta, \lambda} P_{\text{tot}}(\beta, \lambda, U_\infty). \quad (2.17)$$

This problem formulation assumes a constant free-stream velocity,  $U_\infty$ , which is a steady-state formulation. A low-level generator torque control law can be used to regulate the turbine to the optimal TSR. A more realistic formulation treats the free-stream velocity as unsteady and turbulent. In this case, the objective is to maximize the average power generated by the two-turbine array. Moreover, the unsteady flow causes significant structural loads on the tower and blades of both turbines. Thus the formulation can be extended to include constraints on the loads. Alternatively, additional terms can be included in the objective function to trade off the power capture and loads.

The power maximization problem in (2.17) is difficult to solve as it involves complicated models of the turbine operation and wake interactions. As a result, previous work on turbine coordination [2–4] has focused on simplified models for the turbine operation. In particular, the turbines are modeled by actuator disks with a control input of the axial induction factor,  $a$ . As mentioned in Section 2.3, the power and thrust coefficient are functions of  $a$  and hence impacts the velocity deficit. Suboptimal performance of the front turbines is achieved by decreasing the power captured by the front turbine(s). This allows

the downstream turbine(s) to capture more power.

It should be noted that increasing or decreasing the induction factor from the optimal has different effects on the flow dynamics in the wake. For example, a decrease in the induction factor is equivalent to moving left of the peak on the  $C_P$  curve in Figure 2.7, and corresponds to a decrease in  $C_T$ . Decreasing the thrust generated by the turbine leads to less turbulent kinetic energy in the wake directly behind the upstream turbine. Alternatively, increasing the induction factor from the optimal, results in an increase in the thrust coefficient. In general, this will increase the level of turbulence in the wake of the upstream turbine, which will induce more mixing, and the velocity in the wake will recover to freestream at a faster rate. The downstream turbine may see a higher incoming velocity resulting in more available power, due to an increase in thrust. This increase in velocity is dependent on the local conditions, such as ambient turbulence and atmospheric stability. It is important to note that it is also possible, depending on the local conditions, that there is a decrease in velocity at the downstream turbine due to the increase in thrust at the upstream turbine [13]. More details on the impact of turbine operation on the flow within a wind farm will be provided in Chapter 3.

The precise relationship between the downstream wake and the induction factor of the upstream turbine,  $a_1$ , depends on the wake model. Thus the power generated by a two-turbine array can potentially be increased by the proper choice of the induction factors  $a := [a_1, a_2]^T$ . The power maximization problem formulated for this simplified turbine (actuator disk) model is given by

$$\max_a P_{tot}(a, U_\infty). \quad (2.18)$$

The connections between the simplified and more realistic power maximization problems (2.17) and (2.18) are described further in Section 2.7.1. The power computed for a wind farm is dependent on the model chosen to represent the fluid dynamics within a wind farm. The next section details the existing wake models and the differences that arise when computing this power maximization problem.

## 2.6 Wake Models

Various wake models exist that range in fidelity and computational intensity. Each model can help strengthen the understanding of wakes in a wind farm. The majority of this thesis focuses on improving wind farm modeling for control design and analysis.

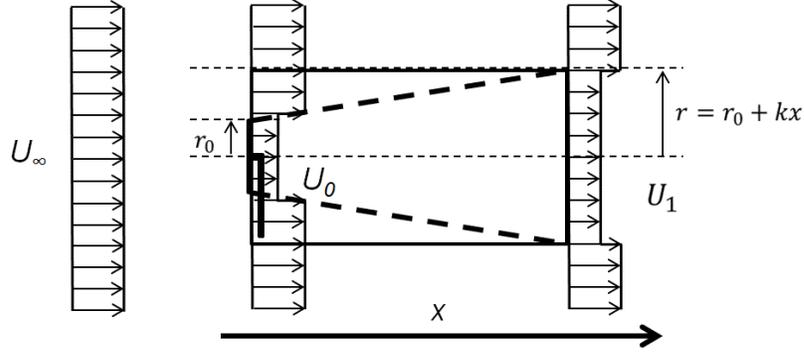


Figure 2.13: Setup for deriving the Park model.

### 2.6.1 Static Park Model

The static wake model typically used in the literature for wind farm control is the Park model [9]. The Park model has a low computational cost due to its simplicity. It is based on the assumptions that there is a steady inflow, linear wake expansion, and the velocity in the wake is uniform at a cross section downstream. The turbine is modeled as an actuator disk with uniform axial loading in a steady uniform flow.

Consider the example of a turbine operating in a freestream velocity,  $U_\infty$ . The diameter of the turbine rotor is denoted by  $D$  and the turbine is assumed to be operating at an induction factor,  $a$ . A cylindrical coordinate system is placed at the rotor hub of the first turbine with the streamwise and radial distances denoted by  $x$  and  $r$ , respectively. The velocity profile at a location  $(x, r)$  as

$$u(x, r, a) = U_\infty(1 - \delta u(x, r, a)), \quad (2.19)$$

where the velocity deficit,  $\delta u$  is given by

$$\delta u = \begin{cases} 2a \left( \frac{D}{D+2\kappa x} \right)^2, & \text{if } r \leq \frac{D+2\kappa x}{2}. \\ 0, & \text{else} \end{cases} \quad (2.20)$$

In this model, the velocity,  $u$ , is defined in the axial ( $x$ ) direction and the remaining velocity components are neglected. The wake is parameterized by a tuneable nondimensional wake decay constant,  $\kappa$  [41, 42]. Typical values of  $\kappa$  range from 0.01 to 0.5 depending on the ambient turbulence, topographical effects, and turbine operation. For example, if the ambient turbulence is high, then the wakes within the wind farm will recover faster due to the mixing of the wake resulting in higher  $\kappa$  values.

The Park model can be used to compute the power production and velocity deficit of a

turbine array. This is useful in determining operating conditions of a wind farm to maximize power. However, the assumptions of the Park model are based on a steady inflow acting on an actuator disk with uniform axial loading. Despite its limitations, the Park model can be computed in seconds and can provide some insight of turbine interaction that can be used to understand the results obtained from higher-fidelity models.

### 2.6.2 Dynamic Wake Meandering Model

The next model considered in this analysis is the medium-fidelity dynamic wake meandering (DWM) model [43]. The University of Massachusetts and NREL developed an implementation of the DWM model that was originally developed at the Technical University of Denmark [44]. It couples FAST with models for the wake deficit, turbulence, and (stochastic) meandering. The foundation of the wake deficit model used in the DWM model is the eddy viscosity model [10]. The wake deficit model numerically solves simplified Navier-Stokes equations based on the thin boundary-layer approximation and assumes a zero pressure gradient. Again, let  $x$  and  $r$  denote the streamwise and radial distance from the turbine rotor hub (Figure 2.12). In this model, the velocity components,  $u$  and  $v$ , are defined in the axial ( $x$ ) and radial ( $r$ ) directions. The velocity components  $u$  and  $v$  satisfy the following partial differential equation

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial r} = -\frac{1}{r} \frac{\partial(\overline{ru'v'})}{\partial r}. \quad (2.21)$$

The right-hand side of (2.21) is modeled using mixing length theory [34] that can be described in terms of turbulent viscosity,  $\nu_T$

$$-\overline{u'v'} = -\nu_T \frac{\partial u}{\partial r}, \quad (2.22)$$

where  $u'$  and  $v'$  denote the fluctuation velocity components in the axial and radial directions and  $\overline{u'v'}$  is a temporal average that represents a turbulent momentum flux that acts like a stress, also known as a Reynolds stress. The turbulent viscosity,  $\nu_T = k_2 b(U_\infty - u_c)$ , describes the shear stresses and eddy viscosity in the wake, where  $b$  is the wake half width,  $u_c$  is the center wake velocity, and  $k_2$  is an empirical constant of the flow field typically set to 0.009 [10].

The DWM model uses Taylor's hypothesis when modeling turbulence. This hypothesis assumes that the turbulence has no effect on the wake advection, i.e. wake transport, from upstream to downstream. A consequence of this hypothesis is that the wake advection is only a function of the mean wind speed. The DWM model is interfaced with a FAST turbine model as follows. The first turbine is simulated in FAST with a three-dimensional

input wind field generated using TurbSim [45]. The DWM model is then used to calculate the downstream wake based on the FAST simulation results for the upstream turbine. The downstream wake is then linearly superimposed on the wind field to generate the velocity conditions for the downstream turbine, i.e. the downstream turbine. Finally, a FAST simulation is performed for the downstream turbine using this wake superimposed wind profile.

The advantage of the DWM model over the Park model is that it gives a more realistic representation of the far wake at a low computational cost. The DWM model can be used to compute the power production, velocity deficit, and structural loads of a turbine array. In addition, the DWM model can run in minutes on a desktop computer. The disadvantage of the DWM model is that it is not suitable for feedback control design because it calculates the wakes of a wind turbine array one at a time, i.e. it does not provide a continuous flow. This complicates the use of this model for dynamic wind farm control.

### 2.6.3 Actuator Disk Model

Another wake model often used in literature is the actuator disk model that solves the unsteady, axisymmetric Navier-Stokes equations by using the streamfunction ( $\psi$ ) - vorticity ( $\Omega$ ) formulation assuming the flow is incompressible and inviscid [46, 47]. Note that there are two types of actuator disk models. The first type is described in Section 2.3 which specifically refers to the turbine. Here, the actuator disk model in this section refers to the flow field computed using the Navier-Stokes equations where the turbines in the wind farm are represented as actuator disks. The remainder of this thesis will use the actuator disk model to refer to the flow field calculations. Let  $(u, v)$  denote the axial and radial velocity components and  $(x, r)$  denote the downstream and radial distances. Vorticity can be defined as  $\Omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial r}$  and the streamfunction can be defined in terms of the axial and radial velocity components:  $\frac{\partial \psi}{\partial x} = rv$  and  $\frac{\partial \psi}{\partial r} = -ru$ . Under some additional technical assumptions, the Navier-Stokes equations are reformulated to the following governing equations

$$\frac{\partial \Omega}{\partial t} + \frac{\partial(u\Omega)}{\partial x} + \frac{\partial v\Omega}{\partial r} = -\frac{1}{\rho} \frac{\partial f_x}{\partial r} \quad (2.23)$$

$$\frac{\partial^2 \psi}{\partial x^2} - \frac{1}{r} \frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial r^2} = r\Omega, \quad (2.24)$$

where  $f_x$  is the volume force of the actuator disk on the flow in the axial direction. In (2.23), the vorticity transport and Poisson equation for the streamfunction are defined. The velocity components  $(u, v)$  can be computed from  $(\psi, \Omega)$ . The turbines are modeled as actuator disks with a specified volume force acting on the flow. For example, an elliptical

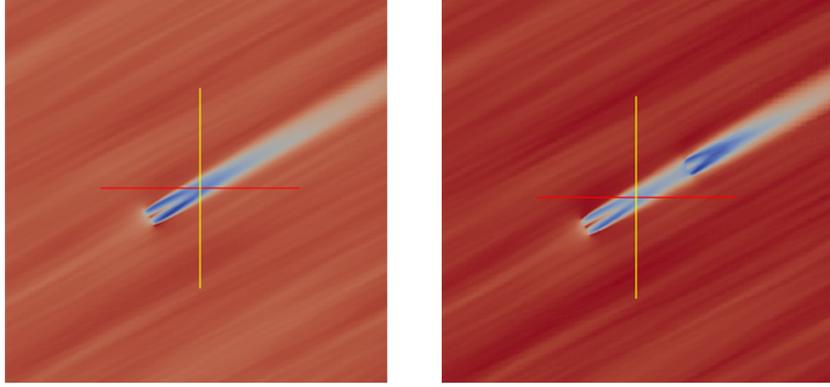


Figure 2.14: One turbine (left) and two turbine (right) mean streamwise velocity computed using SOWFA.

force distribution is used in this analysis and is given by

$$f_x(r) = \frac{3}{2}\rho U_\infty^2 C_T \sqrt{1 - \left(\frac{r}{R}\right)^2}, \quad (2.25)$$

where  $C_T$  is the thrust coefficient and  $R$  is the radius of the turbine. The equations are solved using standard CFD methods [48].

There are limitations to this specific setup for the actuator disk model. Specifically, the actuator disk model ignores viscous effects that are necessary for the velocity in the wake to recover far downstream. However, the power production and velocity deficit of a turbine array can be computed within minutes on a desktop computer.

#### 2.6.4 Simulator fOr Wind Farm Applications

The last model investigated in this analysis is the Simulator fOr Wind Farm Applications (SOWFA). SOWFA is a high-fidelity large-eddy simulation tool that was developed at the National Renewable Energy Laboratory (NREL) for wind farm studies, see Figure 2.14. SOWFA is a CFD solver based on OpenFOAM (OpenCFD Ltd., Bracknell, UK) coupled with NREL's FAST wind turbine simulator [11, 30, 49]. SOWFA has been used in previous wind farm control studies (e.g. [39, 40, 50]).

SOWFA uses an actuator line model coupled with FAST to study turbines in the atmospheric boundary layer. Specifically, SOWFA solves the three-dimensional incompressible Navier-Stokes equations and transport of potential temperature equations, which take into account the thermal buoyancy and earth rotation (Coriolis) effects in the atmosphere.

SOWFA calculates the unsteady flow field to compute the time-varying power, velocity

deficits and loads at each turbine in a wind plant. This level of computation, with high-fidelity accuracy, takes on the order of days to run on a supercomputer using a few hundred to a few thousand processors, depending on the size of the wind plant. The simulations run for this study were performed on NREL’s high-performance computer Peregrine [51] as well as the supercomputer at the Minnesota Supercomputing Institute (MSI) at the University of Minnesota.

Studies have been performed to validate SOWFA. For example, SOWFA has been compared with the 48-turbine Lillgrund wind farm field data and shows good agreement through the first five turbines in a row aligned with the wind direction [52]. In addition, SOWFA has been tested to verify that it captures the inertial range in the turbulent energy spectra and the log-layer in the mean flow, both of which characterize a realistic atmospheric boundary layer [49]. Further validation studies are being conducted.

It should be noted that there are multiple large eddy simulation codes that have been developed such as the Virtual Wind Simulator (VWiS) developed at the Saint Anthony Falls Laboratory [12]. The analysis of wake models was done in collaboration with NREL and thus uses SOWFA for comparing wake models [53].

## 2.7 Comparison of Wake Models

The wake characteristic of each model presented in this chapter were compared by simulating a two-turbine setup (Figure 2.12) over 1000 s. The simulated turbines, based on the NREL 5MW reference turbine [54], have a 126 m diameter and a hub height of 90 m. The wind speed in all simulations has a mean of 8 m/s with 6% turbulence intensity.

Figure 2.15 compares the spatially-averaged streamwise velocity profile of the two-turbine setup for each model. SOWFA is the highest fidelity model considered in this analysis and has been validated against wind farm data. The Park and DWM model results match SOWFA in the far wake at distances greater than approximately  $3D$  downstream. In addition, the Park model wake decay constant,  $\kappa = 0.45$ , was tuned to obtain a best fit agreement with SOWFA in the far wake. The velocity deficit compared from the actuator disk model agrees with SOWFA at distances up to  $3D$  downstream. The actuator disk model is invalid in the far wake because it assumes inviscid flow. It is important to note that the Park, DWM, and actuator disk model use an averaged actuator disk to represent the turbine. Tip vortices in the wake are not resolved and nothing definitive can be said about their accuracy in the near wake. SOWFA implements an actuator line turbine model and is thought to give the closest representation of the near wake of the models presented in this chapter. The wake models presented in the previous section will be compared to SOWFA.

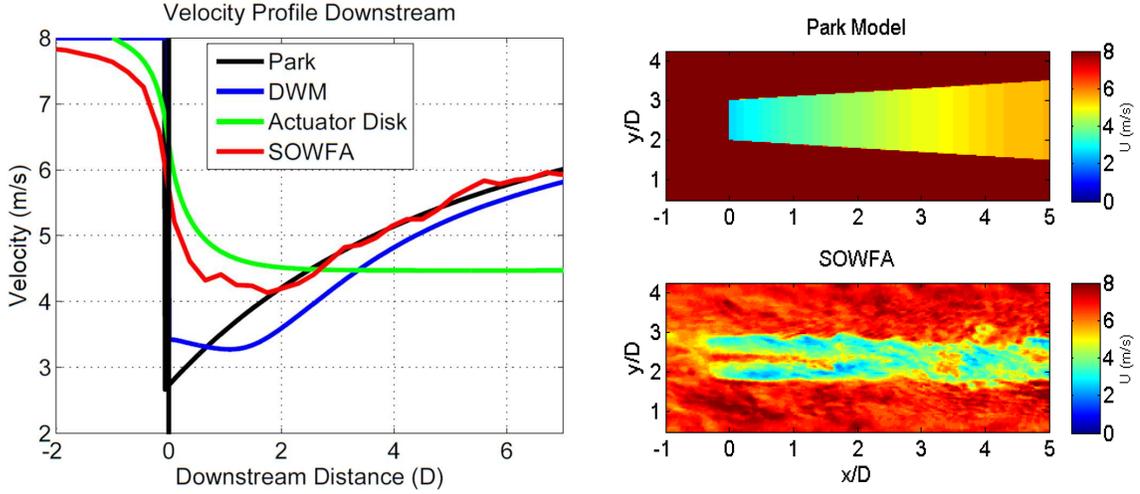


Figure 2.15: (Left) Streamwise velocity component downstream of Turbine 1. (Right) Comparison of the streamwise velocity of the upstream turbine wake using the Park model and SOWFA.

Figure 2.16 compares power production, power variance, and average tower fore-aft bending moments on the downstream turbine for each model. The results are presented for simulations with the turbine spacing varying between  $x = 2D$  and  $x = 7D$ . The average power results are consistent with the velocity profiles shown in Figure 2.15. At distances up to  $3D$  downstream, the actuator disk model results align with SOWFA. However, the power of the actuator disk model remains constant far downstream because there is no wake recovery in the model. The DWM model follows SOWFA at turbine spacings of  $4D$  and greater. These results correspond to cases where the downstream turbine is in the far wake of the upstream turbine. The Park model follows the same trend, but overestimates the power of the two-turbine array.

The middle subplot of Figure 2.16 shows the power variance, and these results reflect the deviation in power over time. The Park model assumes steady flow and does not have a time component in the model. Thus, the variance for this model is identically equal to zero for all turbine spacings. The DWM and actuator disk models show an increase in power variance as the turbine spacing increases. This does not match the qualitative trend of the SOWFA results, which show that the power variance decreases as the turbine spacing increases. The SOWFA results are more reasonable because turbulence decreases as the turbine spacing increases.

Finally, the bottom subplot of Figure 2.16 shows the tower fore-aft bending moment the results from the axial ( $x$  direction) force on the downstream turbine. The DWM model

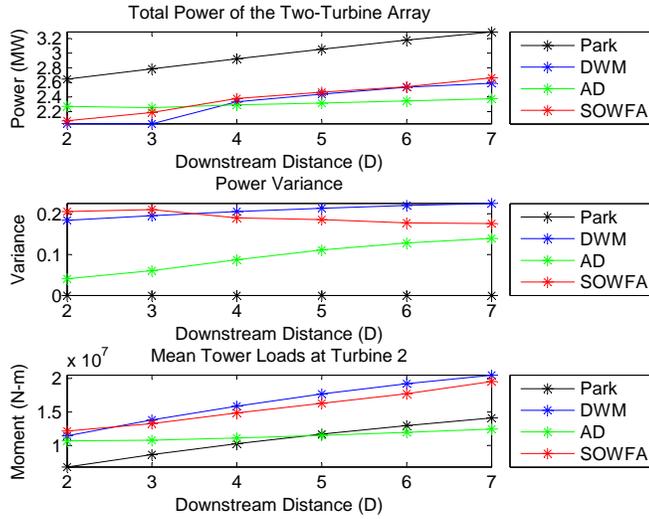


Figure 2.16: Comparison of power (top), power variance (middle), and tower loads (bottom) generated from each wake model.

and SOWFA calculate this moment directly during the simulation. The Park model and actuator disk model only model an overall axial thrust force on the turbine. The tower fore-aft moment can be approximated by multiplying the axial thrust force by the hub height of the turbine. The Park model, the DWM model, and SOWFA show an increase in the bending moment as the turbine spacing increases. The actuator disk model shows a relatively constant bending moment, which is caused by the lack of wake recovery in the model. The limitations in the Park model and the actuator disk model can be seen since both models can only provide a total axial force on a turbine. SOWFA and the DWM model are coupled with FAST, hence they provide a more accurate description of the tower loads.

### 2.7.1 Maximizing Power with Each Model

This subsection provides results for two-turbine open loop coordinated control on the Park, DWM, and actuator disk models. Section 2.5 formulated the power maximization problems with realistic (2.17) and simplified actuator disk (2.18) turbine models for a single turbine. In both cases, the maximum power from the two-turbine array is obtained by operating the rear turbine at its peak efficiency. Thus the optimization reduces to a determination of the optimal derating for the lead turbine. The Park model uses a one-dimensional representation to model the turbines with an induction factor control input. The Park model is relatively simple, hence the optimal induction factor for the lead turbine,  $a_1$ , can be determined numerically as previously demonstrated in [8].

When using the actuator disk model to characterize the flow,  $C_T^*$  is chosen for the upstream turbine that optimizes the power out of a two-turbine array. The optimal induction factor for the actuator disk model can be computed from  $C_T^*$  using (2.8).

The DWM model uses FAST to provide a more realistic turbine model and takes an input of blade pitch angle,  $\beta$ , and TSR,  $\lambda$ . The standard generator torque control law can be used to track the desired  $\lambda$ . After numerous open-loop runs at various  $(\beta, \lambda)$ , the optimal  $(\beta_*, \lambda_*)$  of the first turbine can be approximately determined. This  $(\beta_*, \lambda_*)$  can be mapped to a  $C_P^*$  value using the software package, WT\_Perf, developed at NREL [55]. This uses blade element momentum theory (described in Section 2.4) to compute the power and thrust coefficients for a turbine at specified operating points. The  $C_P^*$  can be related to the optimal induction factor using (2.9). Equations (2.8) and (2.9) are derived from simplistic models and are generally not realistic. However, they are sufficient for axial induction factors to be compared across wake models. This approach only provides a suboptimal solution to the higher-fidelity power maximization problem. SOWFA also uses FAST to model the individual turbine dynamics, hence the approach described for the DWM model could also be used to generate control inputs for SOWFA.

Figure 2.17 shows the results of axial induction control with various wake models. The Park model, with a tuned value of  $\kappa$ , shows a decrease in power gained as the turbine spacing increases. The DWM model shows that there is an increase in power, but only for cases where the turbine spacing is less than 4D. As the turbine spacing increases, there is more time for wake recovery. After a certain distance downstream of the turbine, the effects of axial induction control become less significant. The actuator disk model shows no improvement in the power of a two-turbine array when using axial induction control. This is due to the lack of wake recovery modeled using this approach.

Lastly, SOWFA also showed no improvements in the power of a two-turbine array when using axial induction control. This was investigated further in the next section to understand the discrepancies between SOWFA and the other wake models with the goal of understanding the dominant characteristics that govern the flow within a wind farm.

### 2.7.2 Evaluation of Axial Induction Control using SOWFA

A two-turbine was analyzed to understand the effects of axial induction control. More results can be found in [13]. Figure 2.18 (left) shows the simulation results of axial induction control for the two-turbine setup spaced 5D apart at 8 m/s with 6% turbulence intensity. The axial induction is modified for the front turbine by offsetting the collective blade pitch angle from the optimum setting (zero pitch offset for the NREL 5 MW turbine). The analysis shows that the turbine-level power optimal setting also yields maximum power production for the total

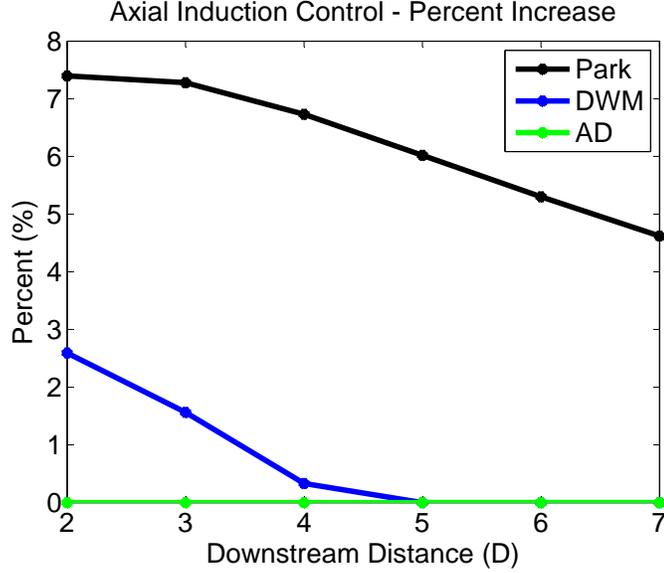


Figure 2.17: Comparison of potential percentage of power gained using axial induction control with different wake models.

wind plant. Although decreasing the axial induction of the front turbine causes an increase in power of the second turbine in the row, the power lost on the first turbine by offsetting the pitch is not regained at the second turbine. Figure 2.18 (right) shows the results of reducing the axial induction of the front turbine by modifying the generator torque. A scaling factor,  $\gamma$ , is applied to the generator torque control law [54] of the front turbine, so that the applied generator torque is  $\tau_g = \gamma K_g \omega^2$  with  $K_g = 0.0179 \text{ Nm/RPM}^2$ , resulting in a deviation from the turbine-level optimal gain,  $K_g$  for maximum power production. A reduction in TSR is needed to decrease  $C_T$  and lower the rotor axial-induction factor. This is achieved by increasing the generator torque ( $\gamma > 1$ ). Note, the  $C_T$  has a low sensitivity to the TSR, and the possible increase of the generator torque is limited because the rotor may stall when a temporary reduction of wind speed occurs. When increasing torque on the upstream turbine, a small increase in power on the downstream turbine can be observed. However, as in the pitch case, there is not enough power increase at the downstream turbine to compensate for the power production loss on the front turbine from adjusting the torque, and a decrease in total power production results.

To understand these results, an investigation was performed using the single-turbine setup. In this simulation, a two-degree pitch offset was applied to the upstream turbine and compared to the baseline case. For both simulations, the flow data was extracted at planes perpendicular to the mean wind direction at several distances downstream of the rotor.

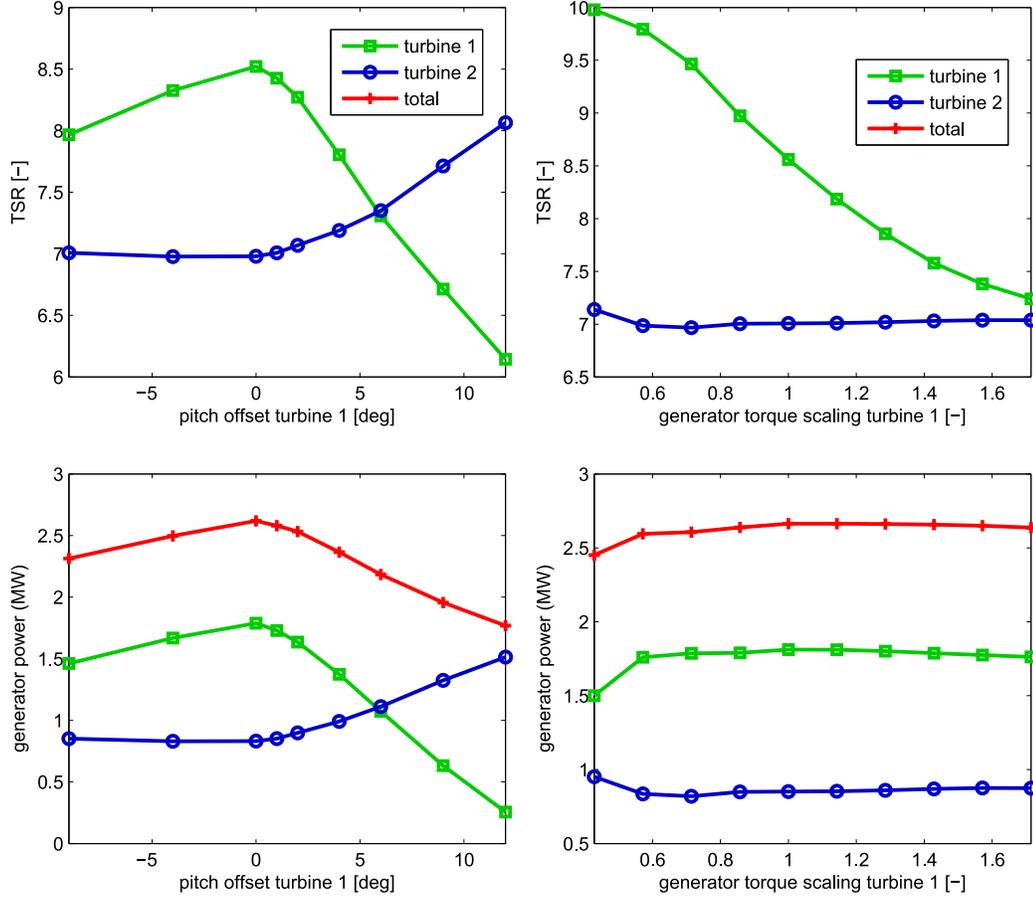


Figure 2.18: SOWFA simulation results using the axial induction control concepts on a two-turbine setup, using pitch (left) or generator torque (right) offsets on the front turbine to affect the power on the downstream turbine.

The kinetic power density of the flow through the slice is calculated as

$$P_{\text{density}} = u_{\text{axial}} \left( \frac{1}{2} \rho \vec{u} \vec{u}^T \right), \quad (2.26)$$

where  $u_{\text{axial}}$  is the axial component of the velocity of the flow through a slice, and  $\vec{u}$  is the velocity vector. By subtracting the kinetic power density of the slices for the baseline case from those of the offset case, the kinetic power added to the wake by pitching the turbine is calculated. Figure 2.19 shows the difference in the kinetic power density of the wind flowing through the cut-through slices. By visualizing the rotor plane of a “virtual” rotor of equal size placed downstream aligned in the wind direction, Figure 2.19 shows that the kinetic power conserved in the flow by using a pitch angle offset on the turbine is mostly going outside of the downstream rotor plane, because the wake expands and meanders outside of the rotor area; therefore, the pitch offset on the front turbine would cause a production loss

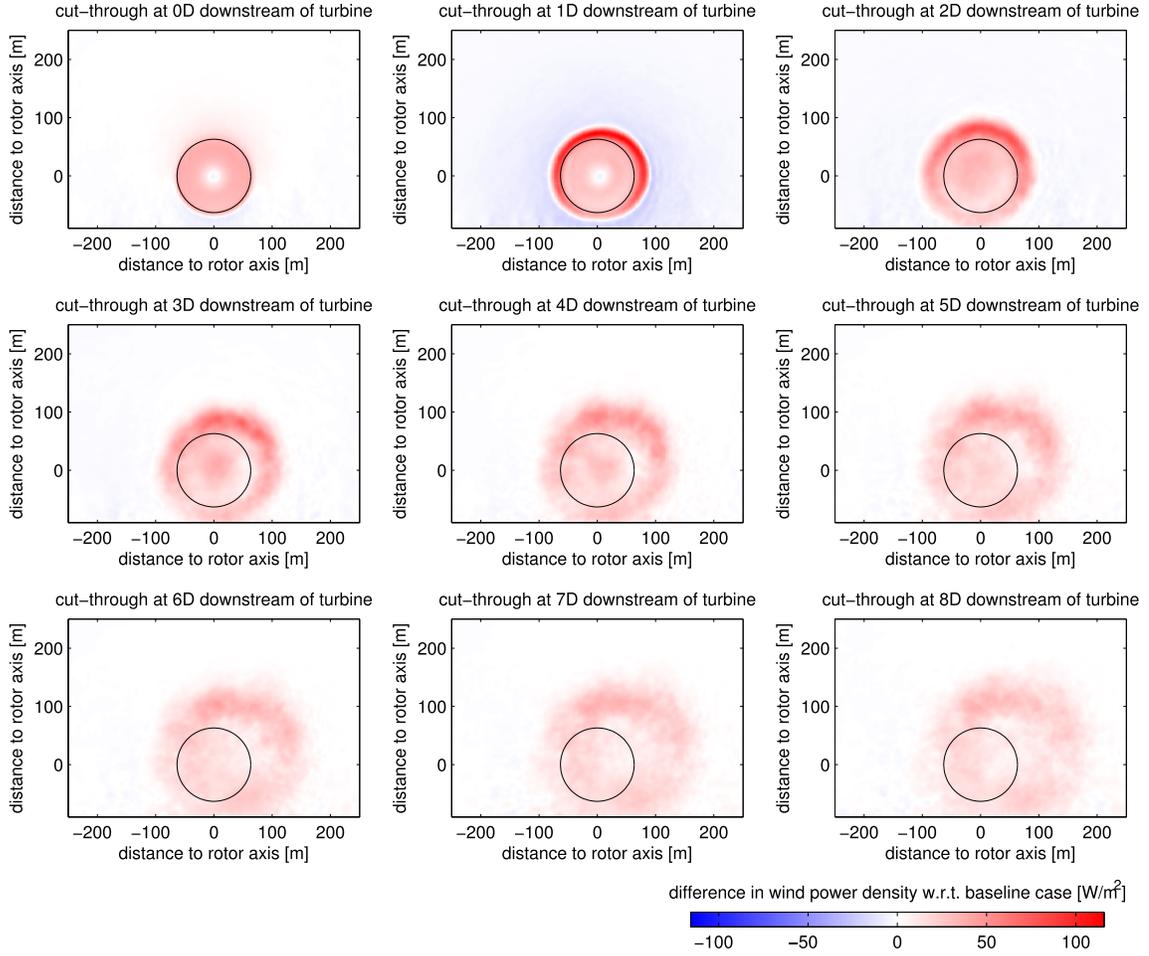


Figure 2.19: Kinetic power added to the flow behind the turbine by introducing a  $2^\circ$  pitch offset. The black circle visualizes the location of the second turbine rotor.

on the two-turbine setup, as it results in accelerating the flow surrounding the downstream rotor rather than increasing the downstream turbine’s power production. A larger portion of the energy is lost when the downstream turbine is placed farther downstream. (Note that 6D to 8D are common distances in real wind farms.) A second cause for the limited ability to improve production at the downstream turbine through pitch control offsets on the upstream turbine is that a reduction in turbine thrust force can reduce turbulence in the wake and thereby the wake recovery, which has a negative effect on the velocity at the downstream turbine.

Based on the results shown in Figure 2.19, an energy balance was made, which is shown in Figure 2.20. The balanced predicts the effect of the front turbine’s pitch offset on the power of a downstream turbine placed at a range of distances. In the balance,  $\Delta P_{\text{wind}}$  represents

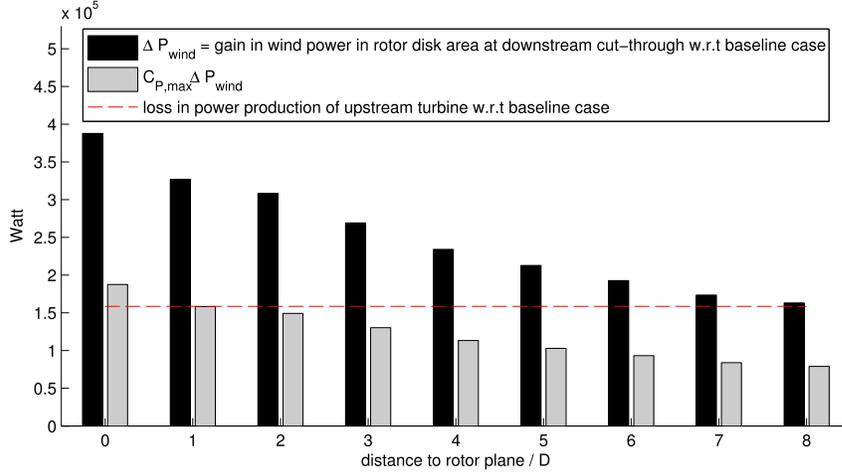


Figure 2.20: Energy balance based on the results shown in Figure 2.20: power lost on the upstream turbine by introducing a  $2^\circ$  pitch offset,  $\Delta P_{T1}$  (red line), compared to flow power added in a virtual downstream rotor plane,  $\Delta P_{\text{wind}}$  (black bar), and the maximum power that can be recovered by the downstream turbine,  $C_{P,\text{max}}\Delta P_{\text{wind}}$  (gray bar).

the total wind kinetic power increase in the area of a virtual downstream rotor. This power increase is compared to the power lost on the upstream turbine by pitching, denoted by  $\Delta P_{T1}$ . The comparison is made for a range of distances of the virtual downstream rotor. Each of the power differences is normalized to the baseline power of the upstream turbine. If we consider that the NREL 5MW turbine can operate at a maximum  $C_{P,\text{max}} = 0.48$  efficiency [54], it follows that the maximum energy gain on a downstream turbine is  $C_{P,\text{max}}\Delta P_{T1}$ . Then the balance predicts that with the simulated turbine and flow conditions, it is impossible to recover the energy lost through offsetting the pitch on the upstream turbine, because  $C_{P,\text{max}}\Delta P_{\text{wind}} < \Delta P_{T1}$ . Thus, it predicts that an increase in total power cannot be achieved with the tested pitch offset when the downstream turbine is placed at a realistic spacing (more than 1D).

## 2.8 Conclusions

There is potential for optimizing wind farm performance by coordinating turbine controllers. However, the control settings that improve wind farm performance depend on atmospheric conditions, wind farm configurations, and the turbine characteristics. High-fidelity simulations suggest that there are circumstances in which the concept of total wind plant power increase through axial induction control with pitch or torque offsets is infeasible.

In general, this chapter demonstrates that the coordinated control design requires accurate wake models with low computational cost. Low-fidelity models can provide useful insight into wake interactions, but lack the complexity to provide realistic wind farm re-

sults. Medium- and high-fidelity models are necessary for constructing an advanced control framework that can be used to optimize turbine placement and control design in a wind farm.

The results seen with SOWFA differ significantly from the results obtained using the Park model. This chapter shows one example of coordinated turbine control that highlight the differences between using low-fidelity models and high-fidelity models. There still remains the potential to improve wind farm performance by properly coordinating turbines. As illustrated in this chapter, the modeling aspect of the wind farm problem is important to developing these strategies. The remainder of this thesis investigates ways of improving the modeling of wind farms such that results from low-dimensional, computationally efficient dynamic models match the results of the higher fidelity models like SOWFA.

## Chapter 3

# Data-Driven Modeling using Wind Tunnel Experiments

### 3.1 Introduction

An improved understanding of the aerodynamic interactions between turbines can aid in the design of enhanced control strategies that coordinate all turbines in a farm. There have been some studies in the field and in wind tunnels looking at these aerodynamic interactions [56–58]. This chapter focuses on constructing a dynamic control-oriented turbine model that is derived from data of wind tunnel experiments. The experiments were conducted to better understand the wake interactions and impact on power production in a three-turbine array. The upstream turbine operating condition was modulated in time (using square waves) and the dynamic impact on the downstream turbine was recorded through the voltage output time signal. The flow dynamics observed in the experiments were used to improve a state wake model, i.e., the Park model described in Section 2.6.1, often used in literature for wind farm control. This model provides a fast, preliminary description of the wake interactions in a wind farm, but neglects some key dynamics, such as effects due to large-scale turbulent structures and wake meandering [44]. Some additional studies extended the Park model to include additional dynamics [42, 59–61].

These experiments were conducted in the atmospheric boundary layer wind tunnel at the Saint Anthony Falls Laboratory (SAFL) at the University of Minnesota using particle image velocimetry (PIV) for flow field analysis and turbine voltage modulation to capture the physical evolution in addition to the dynamics of turbine wake interactions. The Park model is compared to experiments that were done in the wind tunnel using a three-turbine setup with model turbines, aligned in the wind direction, described in Section 3.3. The

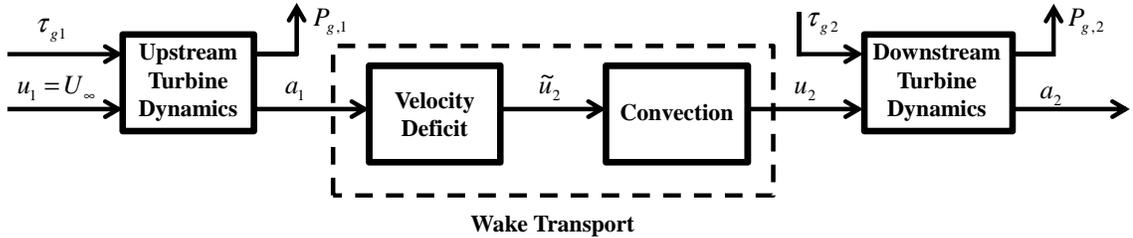


Figure 3.1: Block diagram of the dynamic Park model.

results from the comparison are presented in Section 3.4 along with some suggestions on how to improve the Park model to account for the flow dynamics. Lastly, Section 3.4.3 describes how the experimental data are used to construct a dynamic model that can be used for control of a three-turbine array.

## 3.2 Dynamic Park Model

The static Park model (Section 2.6.1) can be generalized to include turbine dynamics and the effects of wake transport. This section describes a dynamic formulation of the Park model, hereafter referred to as the dynamic Park model. The dynamic Park model includes models for the upstream and downstream turbine dynamics as well as for the wake transport. A block diagram of this dynamic Park model is shown in Figure 3.1. The following subsections describe the models used for the turbine dynamics and wake transport. Then a complete linearized model is derived for the dynamic Park model. This linearized dynamic model can serve as a design model for the wind farm controller used in the wind tunnel.

### 3.2.1 Turbine Dynamics

Section 2.6.1 discussed a static turbine model with the axial induction factor,  $a$ , as the controllable input. For this static model, the power coefficient,  $C_P(a)$ , has a cubic dependence on induction (2.9). An actual utility-scale turbine has blade pitch  $\beta$  [rad], and generator torque,  $\tau_g$  [Nm], as controllable inputs. In the experiments the blade pitch,  $\beta$ , is held fixed. Hence the model described in this section includes generator torque,  $\tau_g$  as the only controllable input. Constant blade pitch, for a utility-scale turbine, corresponds to operation in low wind speed (Region 2) conditions. If  $\beta$  is constant then the aerodynamic forces on the turbine depend only on the nondimensional tip-speed ratio (TSR) defined as  $\lambda$ . Recall that TSR can be defined as  $\lambda := \frac{\omega R}{U_\infty}$  where  $\omega$  [rad/s] is the rotor speed,  $R$  [m] is the radius, and  $U_\infty$  [m/s] is the freestream velocity. For this more realistic model, the power coefficient is given by a function  $C_P(\lambda)$ . The captured power,  $P_c$  and aerodynamic torque  $\tau_{\text{aero}}$  can then

be expressed as  $P_c = \frac{1}{2}\rho AU^3 C_P(\lambda)$  [W] and  $\tau_{\text{aero}} := \frac{P_c}{\omega}$  [Nm]. This leads to the following simplified, single degree-of-freedom rotor dynamics for the turbine

$$\dot{\omega} = \frac{1}{J} (\tau_{\text{aero}} - \tau_g), \quad (3.1)$$

where  $\dot{\omega}$  [rad/sec<sup>2</sup>] is the angular acceleration of the rotor shaft and  $J$  [kgm<sup>2</sup>] is the rotational inertia. Equation (3.1) is simply Newton's second law for the rotational dynamics of the rotor shaft. Many details of the actual turbine dynamics, e.g., tower/blade flexibility, have been neglected in this one-state model. However, it is sufficient to capture the key dynamics, such as wake meandering, at the wind farm scale.

The outputs for this simplified turbine model are the generated power  $P_g$  and the induction factor,  $a$ . The generated power is given by  $P_g = \tau_g \omega$  [W] and this represents the power converted to useful electrical power. The generated and captured powers are the same in steady state. However, they can differ in transient scenarios as some energy goes to the acceleration/deceleration of the rotor shaft. The axial induction factor,  $a$ , of a turbine is the result of the operating conditions of the turbine. Specifically,  $a$  depends only on the tip-speed ratio, i.e.,  $a(\lambda)$ . Blade element momentum theory can be used to derive expressions for  $C_P(\lambda)$  and  $a(\lambda)$  in terms of the geometric and aerodynamic properties of the turbine blades [29]. The lift to drag ratio of the turbine blades is a key parameter in these models. More details are included in Section 2.4.

The simplified nonlinear turbine dynamics consist of the rotor dynamics (3.1) along with the expressions for  $a$ ,  $C_P$ ,  $\lambda$ , and  $\tau_{\text{aero}}$ . Specifically, the model has one state ( $\omega$ ), one controllable input ( $\tau_g$ ), one disturbance input ( $U$ ), and two outputs ( $P_g$  and  $a$ ). The state derivative and outputs can be completely expressed terms of the state and inputs as follows:

$$\dot{\omega} = \frac{1}{J} \left( \frac{\rho AU^3 C_P \left( \frac{\omega R}{U} \right)}{2\omega} - \tau_g \right) := f(\omega, \tau_g, U) \quad (3.2)$$

$$\begin{bmatrix} P_g \\ a \end{bmatrix} = \begin{bmatrix} \tau_g \omega \\ a \left( \frac{\omega R}{U} \right) \end{bmatrix} := h(\omega, \tau_g, U). \quad (3.3)$$

This same dynamic model is used for both the upstream and downstream turbines. For the upstream turbine, the wind speed across the rotor plane is given by the freestream velocity, i.e.  $U = U_\infty$ . For the downstream turbine, the wind speed across the rotor plane,  $U = u_2$ , depends on the (time-dependent) velocity in the wake.

### 3.2.2 Wake Transport

The turbine dynamics described in the previous section are used to compute the time-varying axial induction for the upstream turbine,  $a_1(t)$ . In steady-state situations, the (static) Park model (2.20) can be used to compute the downstream velocity deficit due to the axial induction of the upstream turbine. In time-varying situations, the actions of the upstream turbine are not instantaneously observed at the downstream turbine. The time it takes for the wake to propagate downstream depends on the convection velocity and the distance between the turbines.

The wake transport model used here consists of the (static) Park model to determine the velocity deficit and a time delay  $\tau_d$  to account for convection delay. Assume the downstream turbine is fully operating in the wake of the upstream turbine and is located a distance  $x$  behind the upstream turbine. The time-varying velocity at the rotor plane of the downstream turbine is given by

$$\tilde{u}_2(t) = u_1(t) \left( 1 - 2a_1(t) \left( \frac{D}{D + 2\kappa x} \right)^2 \right) \quad (3.4)$$

$$u_2(t) = \tilde{u}_2(t - \tau_d). \quad (3.5)$$

These equations define the wake transport. Specifically, (3.4) accounts for the velocity deficit while (3.5) accounts for the convection delay. In other words, the velocity at the downstream turbine at time  $t - \tau_d$ . The convection delay should depend on the velocity in the wake. For simplicity, the convection delay is approximated using the freestream velocity,  $\tau_d = \frac{x}{U_\infty}$ .

### 3.2.3 Linearized Model

A nonlinear model for many turbines operating in a line can be constructed by linking together the models described previously for the turbine dynamics and wake transport. This formulation provides insight into the aerodynamic coupling that occurs within a wind farm. Section 3.4.2 compares the dynamic Park model with data obtained from wind tunnel experiments. Specifically, a linearized model for the wind farm is used as a basis for comparison. This subsection briefly describes the construction of the linearized dynamics for two turbines. The derivation can easily be extended to three or more turbines.

The linearization is constructed about an equilibrium trim condition. This is a reasonable approximation if the fluctuations about the mean baseflow are small. Specifically, assume the generator torque and wind speed inputs to the upstream turbine are held constant at  $\bar{\tau}_{g,1}$  and  $\bar{u}_1 = U_\infty$ , respectively. The upstream turbine will reach an equilibrium rotor speed,  $\bar{\omega}_1$ , generated power  $\bar{P}_{g,1}$ , and axial induction  $\bar{a}_1$ . The (static) Park model gives the

downstream velocity in steady conditions as  $\bar{u}_2 = \bar{u}_1 \left( 1 - 2\bar{a}_1 \left( \frac{D}{D+2\kappa x} \right)^2 \right)$ . Finally, if the generator torque at the downstream turbine is held constant at  $\bar{\tau}_{g,2}$  then the downstream rotor will reach an equilibrium state/output of  $(\bar{\omega}_2, \bar{P}_{g,2}, \bar{a}_2)$ . This collection of constant values (denoted by overbars) defines an equilibrium trim condition for the two-turbine array.

Deviations from the trim conditions are denoted using  $\delta$ , e.g.,  $\delta\tau_g(t) = \tau_g(t) - \bar{\tau}_g$ . The nonlinear turbine dynamics has only one state. Hence the linearized dynamics for the upstream and downstream turbines have the following transfer function form

$$\begin{bmatrix} \delta P_{g,i}(s) \\ \delta a_i(s) \end{bmatrix} = \frac{N_i}{s + p_i} \begin{bmatrix} \delta\tau_{g,i}(s) \\ \delta u_i(s) \end{bmatrix} \quad \text{with } i = 1,2, \quad (3.6)$$

where  $p_i$  are scalars and  $N_i$  are  $2 \times 2$  transfer-matrices for  $i = 1,2$  and  $s$  indicates the Laplace transform, e.g.  $\delta u_i(s)$  is the Laplace transformed quantity of the velocity at turbine  $i$ , i.e.,  $\delta u_i(t)$ . The wake transport consists of a nonlinear function and a pure time delay. Hence the linearized wake transport contains no dynamics and can be represented in the frequency domain by:

$$\delta u_2(s) = e^{-s\tau_d} D_{\text{Park}} \begin{bmatrix} \delta a_1(s) \\ \delta u_1(s) \end{bmatrix}, \quad (3.7)$$

where  $D_{\text{Park}}$  is a  $1 \times 2$  matrix gain. The linearized components can be combined to obtain a linear model for the two turbine array. For example, the wind tunnel experiments roughly corresponds to varying the upstream generator torque,  $\tau_{g1}$ , and measuring the downstream power,  $P_2$ . The transfer function from  $\delta\tau_{g,1}$  to  $\delta P_2$  is then given by:

$$\begin{aligned} G_{\delta\tau_{g,1} \rightarrow \delta P_2}(s) &= G_{\delta\tau_{g,1} \rightarrow \delta a_1}(s) G_{\delta a_1 \rightarrow \delta u_2}(s) G_{\delta u_2 \rightarrow \delta P_2}(s) \\ &= \left( \frac{(N_2)_{12}}{s + p_2} \right) \left( e^{(-s\tau_d)} (D_{\text{Park}})_{11} \right) \left( \frac{(N_2)_{21}}{s + p_1} \right). \end{aligned} \quad (3.8)$$

Note that this linearized model corresponds to a second order system with a pure time delay. The experimental results provided in Section 3.4.2 can be accurately fit with a model of this form.

### 3.3 Experimental Setup

#### 3.3.1 Wind Tunnel

The experiments used to address wind farm modeling were completed in a closed loop wind tunnel at the Saint Anthony Falls Laboratory (SAFL) on the campus of the University of Minnesota. A 150 kW fan drives the flow into the tunnel test section, which has a length of 16 m and a cross-section of 1.7 m by 1.7 m. Coarse wire mesh and a honeycomb flow

straightner condition the flow prior to entering a contraction with an area ratio of 6.6:1 upstream of the main test section. Immediately at the end of the contraction, a trip is placed to promote boundary layer growth. The model turbines were placed roughly 13 m downstream of the trip in the test section where a turbulent boundary layer thickness of  $\delta \approx 0.6$  m was measured under thermally neutral conditions. The neutral cases investigated in this chapter set the air and floor temperature equal, and were held to within  $\pm 0.2^\circ\text{C}$ . Mean and fluctuating flow statistics of the baseline turbulent boundary layer are provided in [62].

### 3.3.2 Turbine Model

The turbine models tested in the tunnel are approximately 1:750 scale models of the Clipper Liberty 2.5 MW utility-scale wind turbine, see Figure 3.2. The resulting dimensions for the model are a three-blade rotor with a diameter of 0.128 m (GWS/EP-5030x3 rotor). While the original design for this rotor was for use on model aircraft, the blades herein were oriented such that the high pressure surface faced the inflow. In addition, the blade airfoil profiles are nearly flat. The hub height is 0.104 m and remains within the lower 25 % of the boundary layer, similar to the full-scale turbine. The TSR was controlled by selecting the model turbine generator and is on the lower end of the TSR range for turbines used in the field. This set of experiments use a specific hub velocity of approximately 4.5 m/s. It is in the operating range of standard turbines and provides a detailed comparison to wind farm experiments completed at the same velocity. The free-spinning TSR for the model turbine in the undisturbed boundary layer is  $\lambda \approx 4.5$ , while typical values for utility-scale turbines range between  $\lambda \approx 3.5$  and 10.

The model turbines have a small DC generator in which a voltage output can be measured or a voltage input can be applied to control the turbine operating condition. The DC voltage input is restricted to lie within  $\pm 1.25$  V. A zero voltage input corresponds to a free spinning turbine. In this condition, the turbine operates at a high TSR. Applying a positive voltage places a torque on the motor shaft causing the turbine to operate at a lower TSR. A properly chosen voltage results in the turbine operating near its optimal TSR,  $\lambda = 3$ . Thus the DC voltage input mimics the effect of the generator torque on the utility-scale turbine.

### 3.3.3 Three Turbine Setup

#### Operating Conditions

The model turbines were operated in two different states: (i) rated; (ii) derated, i.e., higher thrust. In the rated state, a 1.25 V input was applied from a DC power supply or function generator to the DC generator on the model turbine. This input applies a torque opposing



Figure 3.2: Two turbine setup in the SAFL wind tunnel (left) and the 2.5 MW Clipper turbine at UMore Park (right). Photo credit: Kevin Howard.

the aerodynamic torque and controls the TSR. In the derated state, the turbine was allowed to operated under no load, i.e., a zero voltage input [63]. The rated case corresponds to a turbine operating near the peak of the  $C_P$  curve in Figure 2.7. The derated case corresponds to a turbine operating on the right side of the  $C_P$  curve which leads to a larger  $C_T$  and hence larger turbulence levels in the wake. In these experiments, the voltage generated by each turbine is similar to the power generated by a utility-scale turbine. For more details on turbine voltage production and analysis verification, the reader is directed to [63].

### Experimental Conditions

These experiments consisted of three turbines placed in a row with a five diameter ( $5D$ ) spacing between the turbines, see Figure 3.3. Each experiment was run at a wind speed of  $4.5 \text{ m/s}$  under a neutral boundary layer with  $1.5\%$  turbulence intensity for for  $100 \text{ s}$ . This sample time was selected as it was the limit for the maximum number of samples for the data acquisition system when sampling at  $10 \text{ kHz}$ . Wall parallel particle image velocimetry (PIV) was used to capture the varying physical characteristics in the wake created by the upstream and downstream turbines under rated and derated states with the overall goal of using the results to improve the Park model. This provides some insight as to how specific control actions of Turbine 1 affect the overall flow characteristics in the wake. The use of wall parallel PIV simplified the problem by removing the extra factor of the boundary layer from the wake development. PIV uses a pulsating laser sheet synchronized with high

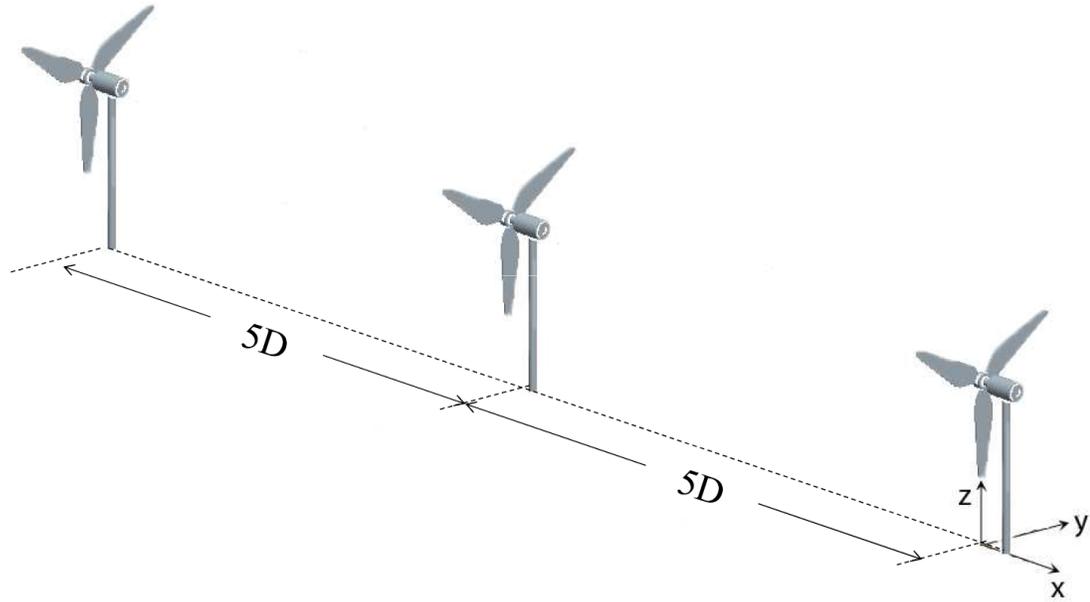


Figure 3.3: Three turbine setup aligned. Turbine 1 is the most upstream turbine, Turbine 2 is the middle turbine, and Turbine 3 is the most downstream turbine.

resolution cameras to capture the instantaneous movement of seeding particles in the flow. Olive oil droplets on the order of 5 to 10 microns in diameter are injected into the wind tunnel and tracked by taking snapshots in time and comparing the locations of individual groups of particles to obtain the change in distance between the subsequent frames. The time between snapshots is known and, therefore, the velocity vectors of the particles can be computed using two-dimensional spatial cross correlation in the interrogation windows in which the full image is subdivided (TSI PIV software). In this specific case, a fine  $32 \times 32$  pixel<sup>2</sup> interrogation window is used with a 50% overlap, providing a spatial resolution of approximately 1.8 mm in the streamwise and spanwise directions. Each run consisted of 700 snapshots over 100 s.

### Measuring the Dynamic Response

In addition to PIV, voltage tests were used to characterize the input/output behavior of the three turbine setup with a goal of constructing a dynamic model of the input/output behavior of the three-turbine array. Several experiments were run to examine the dynamic response of Turbine 2 and Turbine 3 based on the inputs to Turbine 1. Square waves with varying frequencies were applied to the upstream turbine and the voltage response of the downstream turbines was recorded. Square waves were chosen as an input for convenience

based on the function generator and power supplies available in the wind tunnel at the time of the experiments. The square waves varied in frequency from 0.02 Hz to 10 Hz with a 50% duty cycle. Specifically, the square waves varied from 0 V (derated) to 1.25 V (rated), i.e., the turbine was switching states at varying frequencies. By measuring the magnitude and phase of the power at Turbine 2 and Turbine 3, a dynamic model can be constructed that relates the input of Turbine 1 to the voltage output of Turbine 2 and Turbine 3 at different frequencies. The result is a model that can be used for design and analysis for coordinated turbine control in the wind tunnel.

Note that the frequencies in these experiments were chosen to inspect the ability of the model turbines to respond to a given input. At high frequencies, the downstream turbine response to an input at the upstream turbine is limited due to the internal electrical resistance, and primarily, the rotational inertia of the rotor. The frequencies chosen for the wind tunnel voltage experiments can be translated to relevant wind gust events for utility-scale turbines. The model turbine rotor speed was measured using a tachometer and was observed to be approximately 2400 RPM, or 40 Hz. Thus the voltage perturbation frequencies used in the experiments are between 0.0005 to 0.25 of the model rotor turbine speed. The Clipper turbine, in Figure 3.2, has a rotational frequency of around 0.2 Hz (one blade revolution computed in s) while operating in the same region as the model turbine. The voltage perturbations used in the experiments thus scale to frequencies between 0.0001 Hz and 0.05 Hz ( $= 0.2[0.0005 \text{ } 0.25]$ ) for the Clipper turbine. Qualitatively, these time scales correspond to wind variations on the order of hours down to wind gusts, which can be on the order of tens of seconds [64].

### 3.4 Results

The PIV results were used to better understand the characteristics of the flow including mean wind speed, turbulent kinetic energy, and wake meandering. Voltage tests were used to understand the dynamics of the system including the input/output behavior. A better understanding of these characteristics can be used to develop a better model of a wind farm.

#### 3.4.1 Wake Characteristics - PIV

The experimental data gathered using PIV was first used to compute the mean velocity behind Turbine 1 to understand the overall effects of turbine control on the velocity behind the turbine. Figure 3.4 shows the mean wind speeds behind Turbine 1 in the rated (left) and derated (derated) cases. This shows that when the upstream turbine is operating under higher thrust conditions (derated), the turbine blocks more of the flow, forcing the flow around the turbine, resulting in a larger velocity deficit directly behind the turbine than in

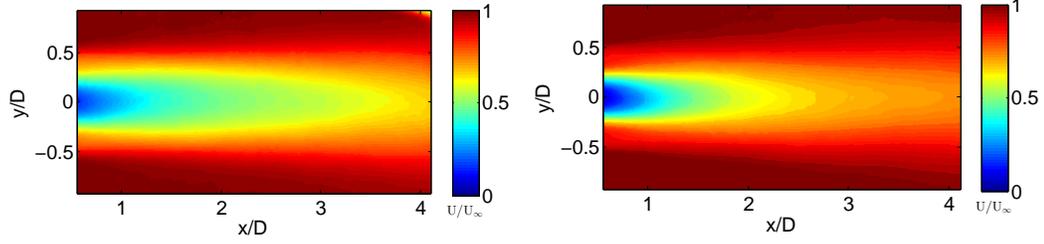


Figure 3.4: (Left) Rated mean streamwise velocity behind Turbine 1. (Right) Derated mean streamwise velocity behind Turbine 1.

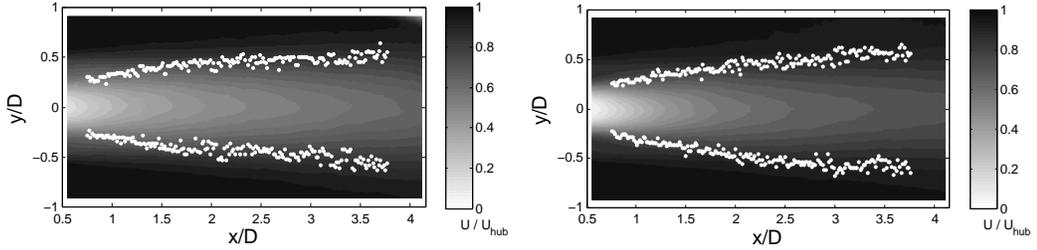


Figure 3.5: (Left) Wake expansion behind Turbine 1 in the rated case. Note, the white dots mark the outside of the wake. (Right) Wake expansion in the derated case.

the rated case. Figure 3.5 shows the growth of the turbine wake, as defined by the domain of its large scale meandering motions [35]. The wake expansion in the  $y$  direction depends on the turbine operating condition. The edges of the wake are tracked in Figure 3.5. The wake in the derated case expands from  $0.5D$  directly behind the rotor to  $0.82D$  at  $3D$  downstream. In the rated case, the wake expands from  $0.5D$  to  $0.74D$ . This means that the wake expands at an average of  $6.1^\circ$  in the derated case and  $4.5^\circ$  in the rated case. More details can be found in [35]. The flow is normalized by the freestream velocity at hub height.

This wake depiction, in Figure 3.4, varies from the Park model described in Section 2.6.1. Recall that the Park model assumed that the velocity in the wake was constant at a given cross section downstream. Figure 3.6 shows the spanwise velocity deficit at  $1D$ ,  $2D$ ,  $3D$ , and  $4D$  downstream in the wake from PIV compared to the Park model assumption. This discrepancy can have a significant impact on control. Some studies have implemented modifications to the Park model by introducing different zones of the wake that provide a more accurate description of the wake [39]. An alternative approach may be to modify the Park model to incorporate a Gaussian velocity profile that evolves as a function of the downstream distance [34, 65].

The mean velocity is also evaluated along the centerline, i.e.,  $y = 0$ , to understand the effects of turbine control on the rate of wake recovery. Figure 3.7 shows the centerline

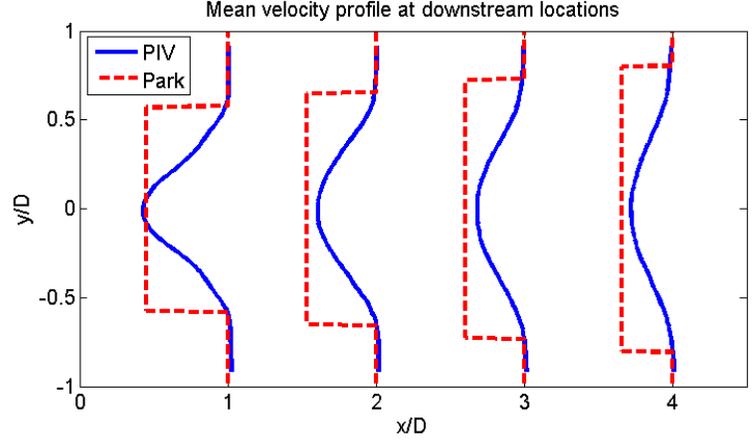


Figure 3.6: Time averaged spanwise velocity profile from PIV experiments compared to the spanwise profile obtained from the Park model

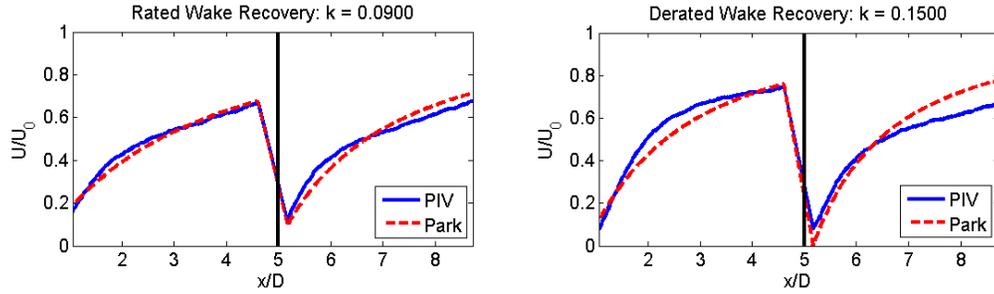


Figure 3.7: Centerline velocity profile from Turbine 1 to Turbine 3. The vertical black line indicates where the second turbine is located.

mean velocity from Turbine 1 to Turbine 3. PIV data captured behind Turbine 1 as well as PIV data captured behind Turbine 2 are combined in Figure 3.7. The vertical black line indicates where Turbine 2 is located. These results are compared to the Park model. In the derated case, a  $\kappa$  value of 0.15 best describes the centerline velocity, while a  $\kappa$  values of 0.09 better describes the rated case. Recall (2.20), a single  $\kappa$ , or wake expansion coefficient, is typically used to describe interactions in a wind farm across all operating conditions. The results in Figure 3.7 suggest that a single  $\kappa$  value may not be able to capture the impact of different operating conditions nor is it able to accurately capture the wake recovery behind Turbine 2. This discrepancy has to do with the different turbulence levels and structures present in the wake.

Figure 3.8 shows the turbulent kinetic energy (TKE) behind Turbine 1 in the rated and derated case. Specifically, when the rated case is compared to the derated case, there is noticeably more TKE in the case where the turbine is operating under derated conditions. This is due to the increase in the induction factor that was described in Section 2.3. This

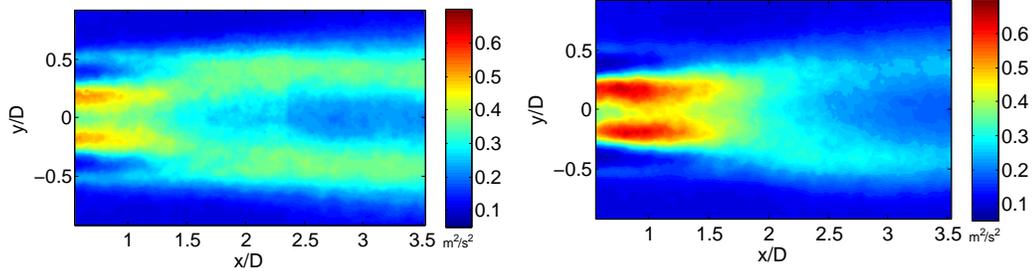


Figure 3.8: Turbulent kinetic energy behind Turbine 1 under two operating conditions: rated (left) and derated (right).

increase in turbulence leads to additional mixing in the wake, causing fluid outside the wake to be entrained into the wake. This allows the wake to recover at a faster rate. As more fluid is entrained the wake expands, as seen in Figure 3.4. This suggests that, in the Park model, a single wake expansion coefficient,  $\kappa$ , does not fully describe the various operating conditions in a wind farm. There has been a study done to investigate this effect [66]. The Park model can be modified to incorporate the change in turbulence due to varying operating conditions. For example, the wake expansion coefficient,  $\kappa$ , can be thought of as a function of the induction factor(s) of the upstream turbine(s) as is done in [13]. Specifically, as the induction factor of the upstream turbine(s) changes, the wake expansion coefficient would also change to accommodate the change in the turbulence level.

### 3.4.2 Propagation Dynamics - Voltage Measurements

A complimentary set of voltage input/output experiments was performed to understand the dynamics of the turbine array and to highlight the temporal limitations of the static Park model. Two different single-input-single-output models are considered in these experiments: the input voltage at Turbine 1 to the output voltage at Turbine 2 and the input voltage at Turbine 1 to the output voltage at Turbine 3. These models could be used to coordinate turbines in the wind tunnel.

Figure 3.9 displays a few selected voltage inputs to Turbine 1 (top row) and the voltage generated at Turbine 2 (middle row) and Turbine 3 (bottom row). The results shown are ensemble averages over the total number of periods contained in each 100s experiment. Specifically, the results shown for 0.07 Hz, 0.2 Hz, and 2 Hz correspond to ensemble average over 5, 20, and 200 periods, respectively. This is important to note because each plot corresponds to an ensemble average over a different number of periods. Notice in Figure 3.9, there is a  $180^\circ$  lag between the voltage input at Turbine 1 and the voltage output at Turbine 2 at low frequencies. When the signal on the upstream turbine is in a rated state, this means that the turbine is operating at a higher efficiency, and is extracting more energy

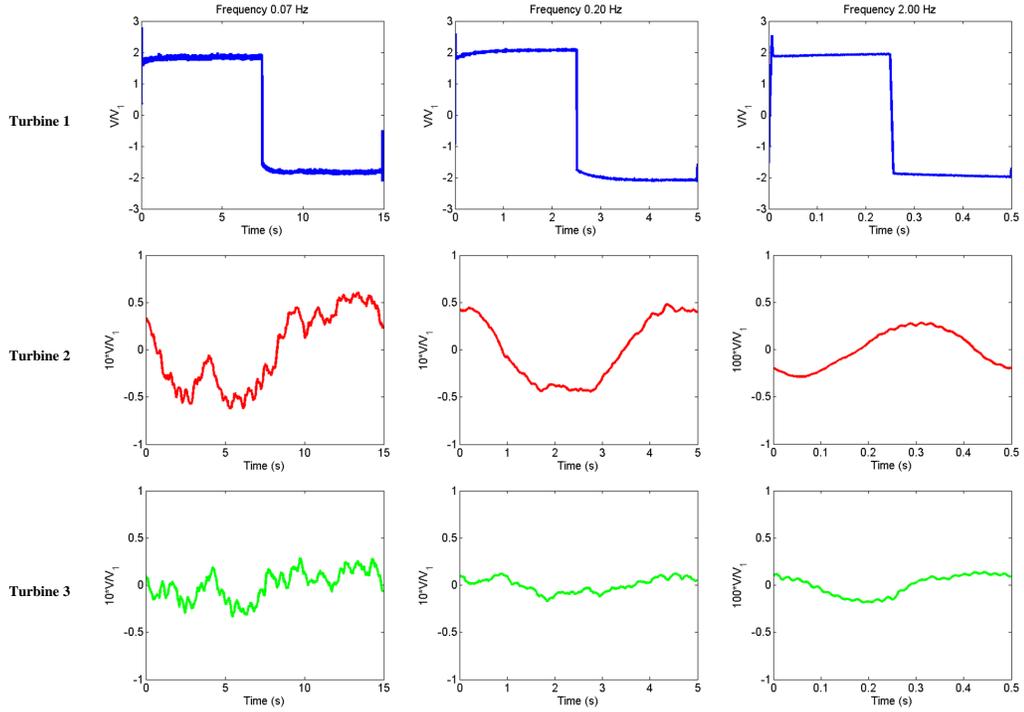


Figure 3.9: Voltage time series at 0.07 Hz (first column), 0.2 Hz (second column), and 2 Hz (third column). The voltages have been normalized by the voltage at Turbine 1. Note that the signals for Turbine 2 (middle row) and Turbine 3 (bottom row) have been scaled by a factor of 10 or 100 for visualization purposes.

out of the wind and overall wake recovery is slower than in the derated case. When the upstream turbine is derated, the wake experiences a faster recovery rate due to the increased thrust generated by the turbine on the flow. This increase in thrust increases mixing and TKE resulting in a higher wind speed at the downstream turbine. It is important to note that this was observed under a neutral boundary layer in the wind tunnel. It is possible that increasing the ambient turbulence and operating under different atmospheric stability conditions could decrease the significance of the changing TSR of the upstream turbine.

As the frequency increases, the voltage production of Turbine 2 and Turbine 3 experiences a greater phase lag from the input at Turbine 1. For example, at 2 Hz (right column of Figure 3.9), the downstream turbine voltage responses experience a greater phase lag than in the 0.07 Hz case. Turbine 2 and Turbine 3 do not respond instantaneously to the effects

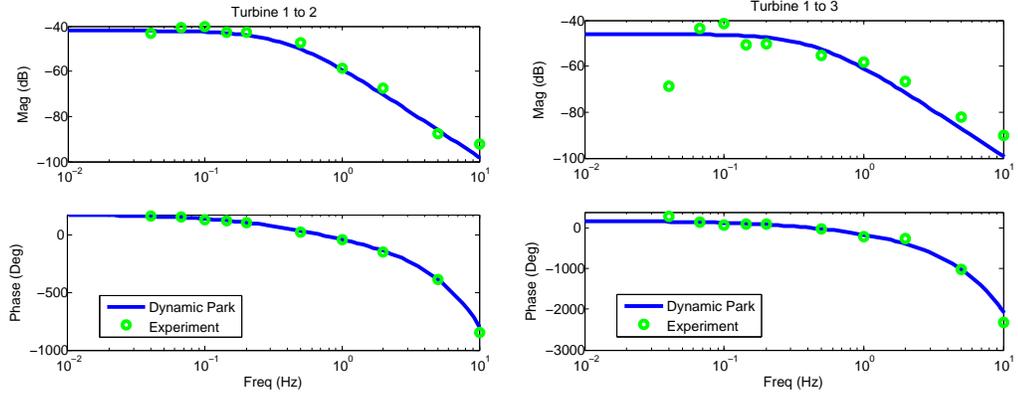


Figure 3.10: Input/output dynamics in the three turbine setup compared to the dynamic Park model.

of Turbine 1. This is due to the time it takes for the effects of the input at Turbine 1 to propagate downstream and impact the performance of Turbine 2 and Turbine 3. At 2 Hz, the input at Turbine 1 is changing nearly as fast as it takes for the previous input to propagate downstream. This results in a phase lag greater than  $360^\circ$ .

From the figures, note the amplitudes of Turbine 2 and Turbine 3 are scaled by a factor of 10 for 0.07 Hz and 0.2 Hz and scaled by a factor of 100 for 2 Hz. This allows for better visual comparisons of the phase and magnitude between Turbines 1, 2, and 3. There is a significant decrease in amplitude between Turbine 2 and Turbine 3. Turbine 3 is located farther downstream from Turbine 1 (10D). The wake has more time to evolve from the control actions of Turbine 1. Lastly, the model turbines have a certain amount of rotational inertia that reduces the turbine response to high frequencies. For example, a utility-scale turbine can respond to large gusts of wind, but due to its large rotational inertia, it cannot respond to the microscales of turbulence.

Frequency domain techniques were used to construct dynamic models from an upstream voltage input,  $u$ , to a downstream voltage output,  $y$ . For each experiment, the Fourier Transform of the domain data samples  $\{u_i, y_i\}_{i=1}^{n_s}$  was computed, where  $n_s$  is the number of samples of  $(u, y)$ . This data consists of complex number  $\{\hat{U}(\omega_j), \hat{Y}(\omega_j)\}_{j=1}^{n_s}$  that depend on the frequency,  $\omega_j$ . The power spectrums ( $\hat{U}, \hat{Y}$ ) contain noise, e.g., due to 60 Hz electric interference. To eliminate the effects of this noise the ratio  $\hat{G}(jf) = \hat{Y}(jf)/\hat{U}(jf)$  was computed only at the excitation frequency,  $f$ , of the square wave input. The magnitude and phase of this complex number,  $\hat{G}$ , computed for each experiment, is shown as a function of frequency (green dots) in Figure 3.10. The magnitude is shown in units of decibels:  $20 \log_{10} |\hat{G}|$ .

The dynamic Park model, described in Section 3.1, can be used to construct a model that agrees with the frequency response of the experimental data and can provide additional physical insight into the system. In these experiments, the voltage input varied at Turbine 1. This voltage input can be thought of as the signal,  $\tau_{g1}$ , in the block diagram in Figure 3.1. It is important to note that the DC motor on the scale model turbine (mentioned in Section 3.3.2) is not specifically modeled, so the relation between input voltage on the scale turbine and generator torque on a utility-scale turbine is inexact. Similarly, the voltage input to Turbine 2 and Turbine 3, is held constant. These inputs correspond to  $\tau_{g2}$  and  $\tau_{g3}$ . Also, the wind speed input in the wind tunnel,  $U_\infty$ , was held constant at 4.5 m/s. The measurements taken during these experiments include the voltage production,  $P_{g,2}$  and  $P_{g,3}$  in Figure 3.1, at Turbine 2 and Turbine 3, respectively. In these experiments, the connection between voltage output on the downstream turbines is roughly related to the power produced on a utility-scale turbine. The frequency plot shown in Figure 3.10, shows the frequency response of the experiments from the input,  $\tau_{g1}$ , to the output,  $P_{g,2}$ , and the input,  $\tau_{g1}$ , to the output,  $P_{g,3}$ .

The parameters in the dynamic Park model (3.8) are estimated to fit the experimental data. Specifically, the physics of the dynamic Park model imply that the transfer function from Turbine 1 to Turbine 2 is parameterized by a DC (steady-state) gain, two real poles for the rotor dynamics, and a time delay for the convection of the wake. The DC gain of 0.0079 was fit as the mean of the low frequency magnitude data. The two real poles were selected as 2.5 rad/s to fit the roll-off in the magnitude data. Finally, a time delay of 0.22 s was selected to match the phase data. These parameters yielded the following gray-box model

$$\hat{G}_{1 \rightarrow 2} = \frac{-0.0497}{s^2 + 5s + 6.25} e^{-0.22s}. \quad (3.9)$$

These parameters could be selected to optimally fit the experimental data, e.g. using `tfest` in the MATLAB<sup>®</sup> System Identification toolbox. However, this yielded second order systems with non-physical complex poles. Taylor’s hypothesis of frozen turbulence can be used to provide an alternative estimate for the time delay. This hypothesis assumes that the turbulence is unchanged as it is advected downstream. i.e. transported by the wake from upstream to downstream. A consequence of this hypothesis is that the wake advection is only a function of the mean wind speed. For example, Turbine 1 and Turbine 2 are spaced 5D, or 64 cm, apart with an inflow velocity of 4.5 m/s. As a result, the time delay should be approximately  $\tau_d = 0.14$  s. This is a conservative estimate as the wake will be moving downstream at a reduced velocity, not the mean inflow velocity, due to the presence of the upstream turbine. Hence the larger delay in (3.9) is reasonable as it represents the effect of the slower velocity in the wake. A similar procedure was used to obtain the following fit

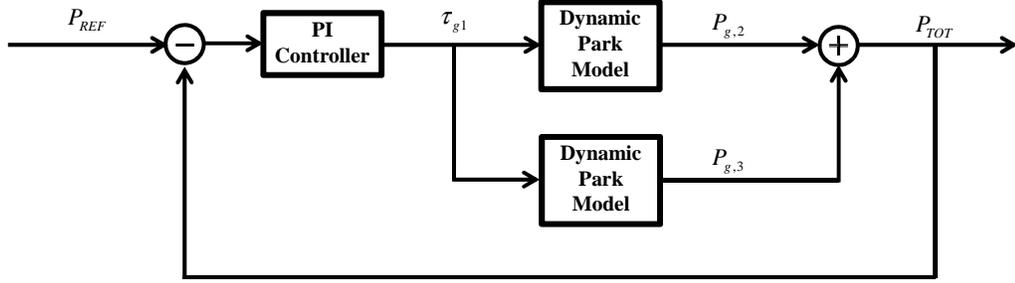


Figure 3.11: PI control implemented using the dynamic Park model.

from Turbine 1 to Turbine 3

$$\hat{G}_{1 \rightarrow 3} = \frac{-0.0439}{s^2 + 6s + 8.75} e^{-0.57s}. \quad (3.10)$$

This second order model assumes that Turbine 2, which lies between Turbines 1 and 3, has negligible impact on the dynamics from Turbine 1 to 3 (although it may impact the steady state gain).

At low frequencies, Figure 3.10 shows that the model and experimental data are essentially static. However, as the frequency increases, the magnitude of the system rolls off. At frequencies greater than 0.5 Hz, the downstream turbine does not have a significant response to inputs from Turbine 1. The turbines are unable to respond to higher frequencies due to the rotational inertia of the turbine. In addition to the magnitude, the phase in Figure 3.10 contains the rotational dynamics of the turbine as well as information about the mean wake velocity. Specifically, the phase information from the experiments can be used to determine the approximate convection velocity in the wake. The amount of phase between the input and output response reflects how long the signal took to propagate through the flow field to the downstream turbine. Turbine 3 experiences a greater phase lag than Turbine 2 due to the distance from Turbine 1 and the time it takes to propagate the effects from Turbine 1 to Turbine 3. This shows how the wake is evolving in time based on the input/output response of this three turbine array.

### 3.4.3 Example Model Application: Wind Farm Control

The dynamic Park model can be used to implement preliminary wind farm control strategies in the wind tunnel. It is important to note that this controller is only implemented in simulation. This section demonstrates the feasibility of designing a controller based on the data-driven model constructed in this chapter. A simple proportional-integral (PI) controller is implemented in simulation to demonstrate the potential applications of using

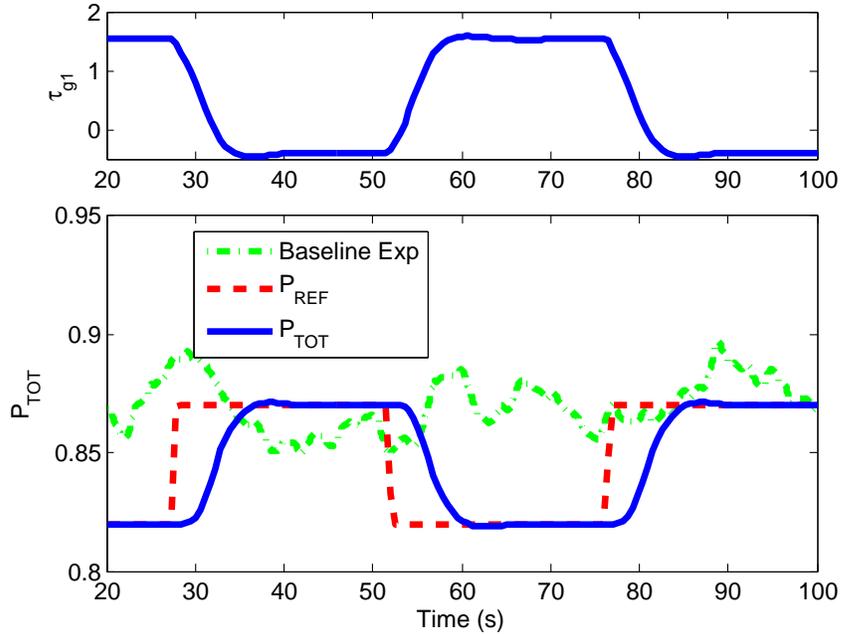


Figure 3.12: Results of implementing a PI controller with the dynamic Park model. This is compared to the results from the wind tunnel with no control. Note  $\tau_{g1}$  is in terms of voltage applied.

the dynamic Park model for wind farm control in the wind tunnel. The PI controller has a proportional gain of -5 and an integral gain of -15. For simplicity turbulence is neglected. The dynamic Park model is used in the configuration seen in Figure 3.11. The goal of this controller is to track the total power, or voltage in this case, produced by Turbine 2 and Turbine 3. This would be similar to a wind farm tracking a power command provided by the electric grid operators [67]. Due to limitations of the hardware, only the output voltages of Turbines 2 and 3 are able to be measured. The wind tunnel/scaled models currently do not allow for closed-loop control, and they can only operate between a TSR of 3 and 4. This is a limiting factor in doing closed-loop control in the wind tunnel. Changes to the current model turbines, such as redesigning the blades, could result in a larger operating region.

Figure 3.12 shows the result of using a PI controller with the dynamic Park model. The top figure shows the simulated voltage input,  $\tau_{g1}$  to Turbine 1 and the bottom figure shows the simulated closed-loop response,  $P_{TOT}$  of Turbine 2 and 3 to the input at Turbine 1. The green dotted-dashed line shows the baseline voltage output of Turbine 2 and Turbine 3 from the wind tunnel with no control from the experimental data.

Using a PI controller, the combined output voltage of Turbine 2 and 3 can follow a specific reference signal. In addition, Figure 3.12 shows that the turbines are able to respond quickly

and accurately to a command. This type of control is significant for wind farms as they may be asked to follow specific energy dispatch signals as more wind is added to the electric grid [67,68].

### **3.5 Conclusions**

An experimental investigation was conducted to analyze the effect of individual turbine control on wind farm dynamics and the results were incorporated to improve the existing static Park model. Model improvements included the incorporation of turbine dynamics that captured the input/output characteristics of a three-turbine array, which can be used for wind farm control in the wind tunnel. Wall parallel PIV was used to identify physical wake characteristics based on varying turbine operating conditions. Voltage tests were used to characterize the frequency response of the system.

Alternative techniques for developing reduced-order wake models using data from experiments (as well as simulations) will be addressed in the following chapter. The data collected from simulations and experiments can be used to extract the dominant characteristics of a wind farm, which can be used for control design and analysis.

## Chapter 4

# Reduced-Order Modeling

### 4.1 Introduction

This chapter describes a method to construct a reduced-order model for high-dimensional systems. The work is motivated by the control of systems that involve fluid and/or structural dynamics. One specific example that is addressed throughout this thesis is wind farm control. As mentioned previously, the overall performance of a wind farm can be improved through proper coordination of the turbines [8]. High-fidelity computational fluid dynamic models have been developed for wind farms [11, 12]. These high-fidelity models are accurate but are not suitable for controller design due to their computational complexity. Simplified, control-oriented models are needed.

A variety of reduced-order modeling techniques have been developed by the fluid dynamics and controls communities. Several of these methods are summarized in Sections 4.3-4.7. These methods range from analytical reduced-order modeling, such as balanced truncation [15], to data-driven reduced-order modeling such as system identification [16, 69] where a low-dimensional system is identified to describe the dynamics of a high-dimensional system. Subspace techniques have been applied to balanced truncation to perform model reduction on nonlinear systems [17]. In the fluid dynamics literature, proper orthogonal decomposition (POD) is a standard method to describe the dominant characteristics of a high-dimensional system where the state is projected onto a low-dimensional subspace of POD modes constructed using data from the higher-order system [18–20]. Dynamic mode decomposition (DMD) is a more recent approach that fits time-domain data with linear dynamics on a reduced-order subspace [21–23]. DMD computes spatial modes of a system at a single frequency. This approach has ties to the Koopman operator [24, 70]. In their basic form, both approaches are for autonomous (unforced) systems. These modes have been used to construct reduced-order models that can be used for control such as balanced

POD (BPOD) and DMD with control (DMDc) [17, 25, 71, 72].

This chapter formulates an extension of DMDc to construct a reduced-order model from simulations or experiments. The details of this method are described in Section 4.8. The model reduction approach has two main advantages. First, it relies on input/output data from a forced response and does not require the formulation/simulation of the system adjoint. Second, the reduced-order model is constructed in a way that retains the physical meaning of the states. This becomes useful for parameter varying systems, which will be discussed in Chapter 5. In other words, the reduced-order state can be mapped back to approximate the full-order state of the system. This method projects the states onto a reduced-order subspace using the dominant modes of the system and then uses direct N4SID to define the reduced-order dynamic model of the system. Lastly, this chapter will analyze the effects of adding inputs to identify a reduced-order model. Specifically, Section 4.9 will address the impact of noise on identifying a reduced-order model and Section 4.10 will analyze how the DMD modes are affected by using a sparsity promoting approach similar to [73]. In particular, it will be shown that the selection of DMD modes is dependent on the choice of input (forcing). Similar to sparsity-promoting DMD, an optimization problem is formulated using a regularization term and alternating direction method of multipliers. Finally, this approach will be applied to the linear channel flow problem and will be compared with standard DMD.

## 4.2 Criteria

There are a variety of methods in the controls and fluids literature that attempt to construct low-order models to describe the dominant dynamics of a high-dimensional system. The motivation of this chapter is to develop a method that can achieve the following:

- *Handle  $> 10^5$  states*: The approach can be used to develop low-order representations of fluid dynamic problems. Typical fluid dynamic problems can have more than a million states.
- *Handle inputs and outputs*: The objective is to develop low-order models for control design on systems with controllable inputs and measurable outputs.
- *Adjoint free*: Some existing methods require the use of a model adjoint [71, 72]. The goal of this chapter is to avoid the use of such adjoints so that the proposed method can be applied to either experimental data or to simulation models.
- *Reduced-order states have physical meaning*: The state of the reduced-order system can be used to approximately reconstruct the full-order state and can assist in the state consistency issue faced by parameter varying systems.

- *Can be used for parameter varying systems:* This is important for nonlinear systems where the dynamics change significantly over the entire operating range.

Sections 4.3-4.7 will address several existing methods as well as their advantages and limitations with respect to the specific criteria defined above. These methods range from analytical approaches such as balanced truncation to data-driven approaches that construct a low-dimensional representation of a high-dimensional system.

### 4.3 Balanced Truncation

Balanced truncation is a typical model reduction approach used in the controls community [15]. Consider a continuous-time linear system:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}\tag{4.1}$$

where (4.1) is a high-dimensional system with  $x \in \mathbb{R}^{n_x}$  as the states,  $u \in \mathbb{R}^{n_u}$  as the inputs, and  $y \in \mathbb{R}^{n_y}$  as the outputs. The state matrices  $(A, B, C, D)$  have dimensions that correspond to  $(x, u, y)$ . To perform balanced truncation on this system, the controllability and observability Gramians need to be computed to understand the influence of the states on the inputs and outputs of the system. Specifically, the controllability Gramian specifies the minimum control energy required to reach any specific state. States that require less energy to reach are more controllable and hence have a greater influence on the input/output dynamics. Similarly, the observability Gramian specifies the energy in the output measurement when the system evolves from a given initial state (with zero input). States that produce more energy in the output are more observable and hence have a greater influence on the input/output dynamics. The Gramians can be computed by solving the Lyapunov equations

$$\begin{aligned}AW_c + W_cA^* + BB^* &= 0 \\ A^*W_o + W_oA + C^*C &= 0,\end{aligned}\tag{4.2}$$

where  $W_c$  is the controllability Gramian and  $W_o$  is the observability Gramian.

The Gramians are defined by specific coordinates. These coordinates define in which directions the strongest states are aligned. The controllability and observability Gramians can have different coordinates. This makes it difficult to choose states to keep since a state may be strongly observable, but not controllable and vice versa. A transformation can be applied to align the properties of the controllability and observability Gramians, which allows you to retain states that are strongly controllable and/or observable. A coordinate transformation

$T$  can be constructed to diagonalize both the controllability and observability Gramians

$$T^{-1}W_c(T^{-1})^* = T^*W_oT = \Sigma = \text{diag}(\sigma_1, \dots, \sigma_n), \quad (4.3)$$

where  $\sigma$  are the Hankel singular values that are independent of the coordinate transformation. Under this transformed system, the states that are significantly influenced by the inputs are also states that have a significant impact on the outputs. With respect to the overall objectives that are defined in Section 4.2, there are advantages and limitations to using this approach.

- **Advantages:** Balanced truncation is adjoint-free and can handle inputs and outputs.
- **Limitations:** This method becomes intractable for large systems (states  $> 1000$ ) as it requires the solution of the two Lyapunov equations (4.2). In addition, the extension of balanced truncation to parameter varying systems requires the solutions of generalized Gramians via semidefinite programming optimizations [74, 75]. This optimization is even more computationally intensive than solving Lyapunov equations.

## 4.4 System Identification

### 4.4.1 Numerical Algorithm for Subspace State Space System Identification

Another method that is popular in the controls community is system identification. Many variation of system identification have been proposed. One approach this chapter will address is the Numerical Algorithm for Subspace State Space System Identification (N4SID) [16, 69, 76, 77]. System identification characterizes the input/output behavior of a system. This is done by fitting a low-order ordinary differential equation that approximately describes the relationship between the input and the output. The general idea of N4SID is to use input/output data to form a low-dimensional state-space representation of (4.1). To construct a reduced-order model using N4SID, input and output data is recorded and arranged in block Hankel matrices,  $U_p, U_f, Y_p, Y_f$ , and an LQ decomposition is performed

$$\begin{bmatrix} U_f \\ U_p \\ Y_p \\ Y_f \end{bmatrix} = \begin{bmatrix} U_{k|2k-1} \\ U_{0|k-1} \\ Y_{0|k-1} \\ Y_{k|2k-1} \end{bmatrix} \begin{bmatrix} L_{11} & 0 & 0 & 0 \\ L_{21} & L_{22} & 0 & 0 \\ L_{31} & L_{32} & L_{33} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} \end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \\ Q_3^T \\ Q_4^T \end{bmatrix}. \quad (4.4)$$

An oblique projection,  $\xi$ , of  $Y_f$  (future outputs) is formed by projecting  $Y_f$  onto past inputs and outputs  $\begin{bmatrix} U_p & Y_p \end{bmatrix}^T$  along  $U_f$  (future inputs). The technical details of computing the oblique projection can be found in [69, 76]. The final equation for the oblique projection

can be written in terms of the LQ decomposition in (4.4)

$$\xi = \begin{bmatrix} L_{42} & L_{43} \end{bmatrix} \begin{bmatrix} L_{22} & 0 \\ L_{32} & L_{33} \end{bmatrix}^\dagger \begin{bmatrix} U_p & Y_p \end{bmatrix}. \quad (4.5)$$

The singular value decomposition of the oblique projection  $\xi = U\Sigma V^T$  is used to estimate the states

$$X_f = \Sigma^{\frac{1}{2}} V^T. \quad (4.6)$$

Once the states are estimated then the corresponding  $\hat{A}$ ,  $\hat{B}$ ,  $\hat{C}$ , and  $\hat{D}$  matrices can be found by solving the regression equation using a least squares technique

$$\begin{bmatrix} \hat{A} & \hat{B} \\ \hat{C} & \hat{D} \end{bmatrix} = \left( \begin{bmatrix} X_{k+1} \\ Y_k \end{bmatrix} \begin{bmatrix} X_k \\ U_k \end{bmatrix}^T \right) \left( \begin{bmatrix} X_{k+1} \\ Y_k \end{bmatrix} \begin{bmatrix} X_k \\ U_k \end{bmatrix}^T \right)^{-1}, \quad (4.7)$$

where  $X_k := [x(1), \dots, x(N-1)]$ ,  $X_{k+1} := [x(2), \dots, x(N)]$ ,  $U_k := [u(1), \dots, u(N-1)]$ , and  $Y_k := [y(1), \dots, y(N-1)]$ . Note that the states  $x(k)$  are components from  $X_f$ . If the states are known, then direct N4SID can be performed to estimate the state matrices. As with balanced truncation, there are advantages and limitations to this approach.

- **Advantages:** This approach is focused on preserving the input-output relationship, which means that it can provide a low-order representation of a high-dimensional system (states  $> 10^5$ ). In addition, this approach does not require an adjoint.
- **Limitations:** The states do not have physical meaning which would help provide some physical intuition about the system. In addition, when addressing parameter varying systems, there is a state consistency issue that stems from the lack of physical meaning in the states.

#### 4.4.2 Eigensystem Realization Algorithm

Lastly, a system identification technique, known as Eigensystem Realization Algorithm (ERA), was developed by Juang [78]. Assuming the linear system in (4.1), the output data is collected in response to an impulse. This data is used to construct a low-order representation of the state space matrices. ERA collects outputs at every time step. Two

Hankel matrices,  $H$  and  $H'$ , are formed

$$H = \begin{bmatrix} Y(t_1) & Y(t_2) & \dots & Y(t_{1+p}) \\ Y(t_2) & \ddots & \dots & \vdots \\ \vdots & & & \\ Y(t_{1+r}) & \dots & & Y(t_{1+r+p}) \end{bmatrix} \quad (4.8)$$

$$H' = \begin{bmatrix} Y(t_2) & Y(t_3) & \dots & Y(t_{2+p}) \\ Y(t_3) & \ddots & \dots & \vdots \\ \vdots & & & \\ Y(t_{2+r}) & \dots & & Y(t_{2+r+p}) \end{bmatrix}, \quad (4.9)$$

where  $r$  and  $p$  are user defined. The SVD of  $H$  is computed  $H = U_r \Sigma_r V_r^T$  where  $r$  is the dimension of the identified model. The low-dimensional system matrices can be computed such that

$$A_r = \Sigma_r^{-\frac{1}{2}} U_r^T H' V_r \Sigma_r^{-\frac{1}{2}} \quad (4.10)$$

$$B_r = \text{the first } n_u \text{ columns of } \Sigma_r^{\frac{1}{2}} V_r^T \quad (4.11)$$

$$C_r = \text{the first } n_y \text{ rows of } U_r^T \Sigma_r^{\frac{1}{2}}, \quad (4.12)$$

where  $n_u$  is the number of inputs,  $n_y$  is the number of outputs. More technical details can be found in [78]. ERA is a special case of system identification and has been shown to be related to dynamic mode decomposition (see Section 4.7) in [23] and has been related to balanced proper orthogonal decomposition in [79]. It is important to note that the model generated by ERA is in balanced coordinates, which can be useful in identifying the most observable and controllable states of the system. However, this is also a drawback in the sense that these coordinates do not have physical meaning and are difficult to incorporate into a scheduling/parameter varying model. In addition, ERA has been extended to include non-impulse inputs using observer Kalman identification (OKID) [80]. Lastly, ERA has been extended to a time-varying framework in [81–83].

- **Advantages:** Similar to N4SID, ERA is also able to provide a low-order representation of a system with many states (states  $> 10^5$ ). ERA can also handle inputs and outputs and is adjoint-free. The additional benefit is that, for linear systems, ERA can compute the modes and eigenvalues of a system providing some physical intuition about the system.
- **Limitations:** As noted above, ERA was developed for linear systems and has difficulty identifying nonlinear systems. ERA cannot handle many inputs [23]. As with

N4SID, there is still a state consistency issue when using ERA.

#### 4.4.3 Linear Parameter Varying Subspace Identification

It should be noted that LPV subspace identification has been investigated recently [84–86]. LPV modeling has also been extended using ERA [87]. This takes care of the state consistency issue exhibited in N4SID and ERA. However, significant amounts of input/output data are required to characterize the response of the system. In addition, the states do not have physical meaning. This method is a good alternative approach when state information is not known. However, in CFD simulations and certain experiments including particle image velocimetry (PIV) [88], the high-dimensional states are known and the state information combined with input/output data should be used to provide a more accurate representation of the system.

#### 4.5 Proper Orthogonal Decomposition

POD provides a low-order approximation of a fluid flow that is capable of capturing the dominant structures in the flow. Specifically, POD can be used to extract dominant spatial features from both simulation and experimental data that can be used to uncover the structures in the flow field [18–20]. This can be done by projecting the velocity field onto a set of orthogonal basis functions.

Consider a system modeled by the following continuous-time nonlinear dynamics

$$\dot{x}(t) = f(x(t), u(t)), \quad (4.13)$$

where  $x \in \mathbb{R}^n$  is the state vector and  $u \in \mathbb{R}^m$  is the input vector. A projection matrix is constructed to minimize the error between the full and reduced-order systems

$$\int_0^{T_{max}} \|x(t) - P_r x(t)\|^2 dt, \quad (4.14)$$

where  $P_r$  is the projection space and  $r$  represents the reduced-order of the system. The eigenfunctions of the flow field are shown to produce the optimal projection that minimizes the total error between the full system and the reduced-order system [20].

Specifically, the POD modes of this system can be computed from the snapshots of the nonlinear system. A data matrix of the snapshots is formed by

$$X_0 = \begin{bmatrix} x(t_0) & x(t_1) & \dots & x(t_m) \end{bmatrix}, \quad (4.15)$$

where  $m$  is the number of snapshots. The POD modes are then computed by taking the

SVD of the data matrix

$$X_0 = U\Sigma V^T. \quad (4.16)$$

The POD modes are contained in the columns of  $U$  and the relative energy of each mode is contained in the singular values in  $\Sigma$ . These modes provide the spatial component of the flow with the first POD mode being the spatial mode that contains the most energy.

The computed POD modes can be used to reconstruct the flow using the Galerkin projection [17,20]. This projection uses a separation of variables approach where the flow field variable can be defined as

$$\mathbf{u}(\mathbf{x}, t) = \sum_{j=1}^r b_j(t)\varphi_j(\mathbf{x}), \quad (4.17)$$

where  $b$  are the temporal coefficients and  $\varphi$  are the POD modes. In this way, the system (4.13) can be rewritten as an ordinary differential equation

$$\dot{b}_k = \sum_{i=1}^r \sum_{j=1}^r b_i b_j Q_{ijk} + \sum_{i=1}^r b_i K_{ik} + c_k + f'_x, \quad (4.18)$$

where  $Q$  represents the nonlinear terms in the flow field,  $u$ ,  $K$  represents the linear terms,  $c$  represents the constant terms, and  $f'_x$  represents the fluctuations in the forcing term in the  $x$  direction. The matrices,  $Q$  and  $K$ , are made up of the POD modes which provide spatial information about the system. Detailed expressions of these matrices can be found in [89]. In this way, a set of POD modes can be used to approximately describe the evaluation of the flow field [17].

- **Advantages:** POD modes are good at representing specific data sets. Specifically, the POD modes capture the energy of the system in an optimal way. The POD/Galerkin approach has been used in the context of control [90]. This approach can handle many states and is adjoint-free. Lastly, this approach has been extended to parameter varying systems [91].
- **Limitation:** Despite the connection of POD/Galerkin to controls, there have been many studies done that show that POD/Galerkin is sensitive to inputs and to the data used [92]. Also, it can often produce unstable results near stable equilibria [93]. Lastly, POD modes mix multiple frequencies. This is important to note since low-energy dynamics can be neglected, which may make POD modes difficult to use for control.

## 4.6 Balanced Proper Orthogonal Decomposition

For some systems, a more relevant objective is to compute a low-order model that captures the most relevant flow dynamics that connect the input to the output. When applying BPOD, the inputs of (4.1) have been set to zeros, such that  $\dot{x} = Ax$ . The initial conditions of  $x$  are defined as the columns of  $B$ , i.e., an impulse response. One simulation needs to be run for each column of  $B$  (i.e. each input). This system will be referred to as the forward system.

In addition to computing the solution for the forward system, the solution to the adjoint system needs to be computed and can be found by integrating the system

$$\dot{v}(t) = A^T v(t), \quad (4.19)$$

with the initial conditions defined as the columns of  $C^T$  and  $v$  is the adjoint variable. As with the forward system, one simulation needs to be run for each output. There is an extension to this method where there are many outputs. This requires an additional projection on the outputs to limit the number of adjoint simulations [72]. Physically, the adjoint system is used to evaluate the sensitivity of the system due to the small perturbations near the output measurements of the system [94].

The BPOD modes can be computed by taking the snapshots from the forward and adjoint systems

$$\begin{aligned} X &= [x_1(t_0), \dots, x_1(t_{n_s}), \dots, x_{n_u}(t_0), \dots, x_{n_u}(t_{n_s})] \\ Y &= [v_1(t_0), \dots, v_1(t_{n_s}), \dots, v_{n_y}(t_0), \dots, v_{n_y}(t_{n_s})], \end{aligned} \quad (4.20)$$

where  $n_s$  is the number of snapshots,  $n_u$  is the number inputs,  $n_y$  is the number of outputs, and  $x_i(t_j)$  denotes the state at time  $j$  caused by the input in the  $i^{\text{th}}$  channel. The SVD of  $Y^T X$  is computed as

$$Y^T X = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}, \quad (4.21)$$

where  $\Sigma_1$  is an  $(r \times r)$  matrix and  $r$  is the reduced-order of the system. Oftentimes, the order  $r$  of the reduced-order model is chosen such that the modes capture 99% of the energy of the system. The reduced-order system can be written as

$$\dot{x}_r(t) = SATx(t) + SBf \quad (4.22)$$

$$y(t) = CTx(t), \quad (4.23)$$

where  $T = XV_1\Sigma_1^{-\frac{1}{2}}$  and  $S = \Sigma_1^{-\frac{1}{2}}U_1^TY^T$ . The BPOD modes are contained in  $T$ .

Note that  $X$  and  $Y$  matrices are  $n_x \times n_s$  matrices where  $n$  is the state dimension, which is typically very large, i.e., tens of thousands or more, while  $n_s$  is the number snapshots, which is typically on the order of hundreds. The Lyapunov equations in (4.2) are of dimension  $n_x$  and directly solving these equations is prohibitive as solving a Lyapunov equation scales with  $O(n_x^3)$  [95]. The product of  $Y^TX$  requires  $O(n_s^2n_x)$  operations which scales linearly in  $n_x$ . The resulting matrix is only  $n_s \times n_s$  and hence the singular value decomposition (SVD) in (4.21) can be performed at a reasonable computational cost.

- **Advantages:** This approach can handle many states (states  $> 10^5$ ). In addition, it can handle inputs and outputs. It should be noted that BPOD has been compared to ERA in [79]. Specifically, it has been shown that, under some circumstances, ERA can produce the same balanced models as BPOD without the adjoint.
- **Limitations:** BPOD has not been extended to parameter varying systems and requires an adjoint. In addition, a linearized simulation needs to be run for every input and the adjoint system needs to be run by every output.

#### 4.7 Dynamic Mode Decomposition

Another popular method in the fluids community is dynamic mode decomposition (DMD). DMD extracts the dominant spatial and temporal information about the flow [21–23]. This method attempts to fit a discrete-time linear system to a set of snapshots from simulation or experiments. Consider a system modeled by the following discrete-time nonlinear dynamics

$$x_{k+1} = f(x_k), \quad (4.24)$$

where  $x \in \mathbb{R}^{n_x}$  is the state vector. A collection of snapshot measurements  $\{x_k\}_{k=0}^{n_s} \in \mathbb{R}^{n_x}$  is obtained for the system either via simulation or experiments where  $n_s$  is the number of snapshots.

The objective of DMD is to approximate the system by projecting the snapshots onto a low-dimensional subspace. Assume there is a matrix  $A$  that relates the snapshots in time by

$$x_{k+1} = Ax_k. \quad (4.25)$$

The snapshots of the system are defined as

$$\begin{aligned} X_0 &= \begin{bmatrix} x_0 & x_1 & \dots & x_{n_s-1} \end{bmatrix} \in \mathbb{R}^{n_x \times (n_s-1)} \\ X_1 &= \begin{bmatrix} x_1 & x_2 & \dots & x_{n_s} \end{bmatrix} \in \mathbb{R}^{n_x \times (n_s-1)}, \end{aligned} \quad (4.26)$$

where  $x_k$  are the snapshots. The full-order  $A$  matrix can be computed such that

$$A = X_1 X_0^\dagger, \quad (4.27)$$

where  $\dagger$  indicates the pseudoinverse. The DMD method attempts to fit the snapshots in time using a low rank matrix that captures the dynamics of the data set. This matrix can be used to construct the DMD modes that correspond to specific temporal frequencies. A low-order representation of  $x_k$  can be written as  $z_k = Q^T x_k$ , where  $Q$  is the projection subspace. The truncated, reduced-order model take the form

$$z_{k+1} = (Q^T A Q) z_k := F z_k. \quad (4.28)$$

The state matrix  $F := Q^T A Q \in \mathbb{R}^{r \times r}$ , describes the dynamics of the reduced-order subspace. Solutions to this reduced-order model can be used to construct the approximate solutions to the full-order model.

The typical choice for the projection subspace,  $Q$ , is the POD modes of  $X_0$  (4.16), i.e.  $Q = U_r$ , where  $r$  indicates the order of the reduced-order model. The optimal reduced-order state matrix,  $F$ , for this choice is

$$F := U_r^T A U_r = U_r^T X_1 (U_r^T X_0)^\dagger = U_r^T X_1 V_r \Sigma_r^{-1}, \quad (4.29)$$

where the corresponding low rank approximation for the full-order state matrix is

$$A \approx U_r F U_r^T = U_r U_r^T X_1 X_0^\dagger. \quad (4.30)$$

If an eigenvalue decomposition is done on  $F$ , then  $A$  is now

$$A \approx (U_r T) \Lambda (T^{-1} U_r^T), \quad (4.31)$$

where  $U_r T$  are defined as the DMD modes and the corresponding values of  $\Lambda$  provide the specific temporal frequency for each DMD mode. DMD has been connected to the Koopman operator that can be used to describe nonlinear systems [96, 97]. There are advantages to using this approach.

- **Advantages:** This approach can handle many states (states  $> 10^5$ ), is adjoint-free,

and the states have a physical meaning. DMD has been extended to include inputs and outputs, termed DMD with control (DMDc) in [25].

- **Limitations:** One limitation of this approach is that DMD has not been extended to parameter varying systems. In addition, it is often not robust to noise (details provided in 4.9).

## 4.8 Input-Output Reduced-Order Modeling

The main contribution of this chapter is provided in this formulation and analysis of input-output reduced-order modeling. The approach will be summarized in this section for time-invariant systems, but will be extended to include LPV models in Chapter 5. This section addresses the use of this technique to develop an input-output model that does not require adjoints and where the states have some physical meaning. The proposed approach combines POD with system identification to produce an input-output reduced-order model (IOROM). This term has been used in the flexible aircraft literature [26]. The proposed method closely follows the procedure used in the formulation of DMDc [25].

Consider a discrete-time nonlinear system with inputs

$$x_{k+1} = f(x_k, u_k) \quad (4.32)$$

$$y_k = h(x_k, u_k), \quad (4.33)$$

where  $x \in \mathbb{R}^{n_x}$ ,  $u \in \mathbb{R}^{n_u}$ , and  $y \in \mathbb{R}^{n_y}$  are the state, input, and output vectors.

A collection of snapshot measurements are obtained via simulation or experiments by exciting the system. Snapshots are taken from the nonlinear system and the states, inputs, and outputs are recorded as

$$\begin{aligned} X_0 &= \begin{bmatrix} x_0 & x_1 & \dots & x_{n_s-1} \end{bmatrix} \in \mathbb{R}^{n_x \times (n_s-1)} \\ X_1 &= \begin{bmatrix} x_1 & x_2 & \dots & x_{n_s} \end{bmatrix} \in \mathbb{R}^{n_x \times (n_s-1)} \\ U_0 &= \begin{bmatrix} u_0 & u_1 & \dots & u_{n_s-1} \end{bmatrix} \in \mathbb{R}^{n_u \times (n_s-1)} \\ Y_0 &= \begin{bmatrix} y_0 & y_1 & \dots & y_{n_s-1} \end{bmatrix} \in \mathbb{R}^{n_y \times (n_s-1)}. \end{aligned} \quad (4.34)$$

This method attempts to fit the snapshot measurements at a particular operating point in time by

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k. \end{aligned} \quad (4.35)$$

The dimensions of the state matrices  $(A, B, C, D)$  are compatible to those of  $(x, u, y)$ . This is similar to direct N4SID [16]. However, this becomes intractable for large systems. Typical fluid dynamic systems have on the order of millions of states. As a result, the state is projected onto the low-dimensional subspace to make the computation tractable.

A truncated model can be expressed in terms of this reduced-order state, i.e.,  $z_k = Q^T \delta x_k \in \mathbb{R}^r$  where  $Q \in \mathbb{R}^{n_x \times r}$  is a generic orthonormal basis that forms the projection subspace

$$z_{k+1} = (Q^T A Q) z_k + (Q^T B) u_k := F z_k + G u_k \quad (4.36)$$

$$y_k = (C Q) z_k + D u_k := H z_k + D u_k. \quad (4.37)$$

The matrices in the reduced-order system have dimensions  $F \in \mathbb{R}^{r \times r}$ ,  $G \in \mathbb{R}^{r \times n_u}$ , and  $H \in \mathbb{R}^{n_y \times r}$ . The form of (4.36) is equivalent to the following low rank approximations for the full-order state matrices

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \approx \begin{bmatrix} Q F Q^T & Q G \\ H Q^T & D \end{bmatrix} = \begin{bmatrix} Q & 0 \\ 0 & I_{n_y} \end{bmatrix} \begin{bmatrix} F & G \\ H & D \end{bmatrix} \begin{bmatrix} Q^T & 0 \\ 0 & I_{n_u} \end{bmatrix}. \quad (4.38)$$

The optimal choice for the reduced-order state matrices  $(F, G, H, D)$  can be computed given the subspace spanned by  $Q$ . This is a similar setup for standard DMD seen in [21–23].

The optimal (reduced-order) state matrices are obtained by minimizing the error of the Frobenius norm using a least-squares approach

$$\min_{\begin{bmatrix} F & G \\ H & D \end{bmatrix}} \left\| \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} - \begin{bmatrix} Q & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} F & G \\ H & D \end{bmatrix} \begin{bmatrix} Q^T & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} X_0 \\ U_0 \end{bmatrix} \right\|_F^2. \quad (4.39)$$

This is the direct N4SID subspace method for estimating state matrices given measurements of the (reduced-order) state, input, and output. A sub-optimal, but useful, choice for the projection space is given by the POD modes of  $X_0$  (4.16). The state of the linear system can be approximated on a subspace defined by the first  $r$  POD modes of  $X_0$ , i.e.,  $Q := U_r$ . The optimal reduced-order state matrices for this choice are given by

$$\begin{bmatrix} F & G \\ H & D \end{bmatrix}_{\text{opt}} = \begin{bmatrix} U_r^T X_1 \\ Y_0 \end{bmatrix} \begin{bmatrix} \Sigma_r V_r^T \\ U_0 \end{bmatrix}^\dagger. \quad (4.40)$$

This yields input-output information for the model. As with standard DMD, an eigenvalues decomposition of  $F_{\text{opt}}$  can be used to construct these modes, which provide spatial modes associated with a specific temporal frequency of the system. The  $F$  matrix describes the dynamics of the system. The  $G$ ,  $H$ , and  $D$  matrices obtained using this method are com-

puted such that the input-to-output behavior of the reduced-order model is preserved from the full-order system. This proposed method is a tractable implementation of the existing direct N4SID (subspace) method that can be applied for very large systems. This is not simply a black-box (input-output) approach because the state of the reduced-order system  $z_k$  can be used to approximately reconstruct the full-order state by

$$x_k \approx U_r z_k. \quad (4.41)$$

Moreover, the approach requires input/output/state data from the system. Construction and simulation of an adjoint system is not required. The next chapter will show how to extend this approach to parameter varying systems.

One way to choose the order of the reduced-order model computed using this approach is to analyze how much energy is captured by the number of modes chosen. The number of modes retained in the system typically amounts to 99% of the energy of the system. However, for a sufficient model that is suitable for control, the primary metric is the amount of model error incurred from the selection of the number of modes. The model error can be computed using the Frobenius norm

$$\left\| \begin{bmatrix} X_1 \\ Y_0 \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_0 \\ U_0 \end{bmatrix} \right\|_F^2. \quad (4.42)$$

Again, computation of this model error is intractable for systems with extremely large state dimensions. However, the properties of the Frobenius norm can be used to equivalently write this error in a more useful form

$$\left\| \begin{bmatrix} X_1 \\ Y_0 \end{bmatrix} - \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_0 \\ U_0 \end{bmatrix} \right\|_F^2 = \left\| \begin{bmatrix} Q^T X_1 \\ Y_1 \end{bmatrix} - \begin{bmatrix} F & G \\ H & D \end{bmatrix} \begin{bmatrix} Q^T X_0 \\ U_0 \end{bmatrix} \right\|_F^2 + \left( \|X_1\|_F^2 - \|Q^T X_1\|_F^2 \right). \quad (4.43)$$

The first term represents the model error on the projected subspace. The second term represents the energy lost in the snapshot data  $X_1$  by using the projection. The right hand side of this equation is now tractable because the states are projected onto a subspace and have dimension  $r$ . More importantly, only the reduced-order state matrices,  $(F, G, H, D)$ , are needed for the calculation rather than the full state matrices  $(A, B, C, D)$ . In many fluid dynamic examples, it is infeasible to compute  $A$ , which would have dimensions of  $10^6 \times 10^6$ . Increasing the number modes will generally decrease the total error. However, there is a point when adding additional modes will not significantly improve the model error. In fact, adding the modes beyond this point could result in a model that is trying to overfit the nonlinearities in the system resulting in a degradation of the performance.

## 4.9 Impact of Process Noise in Identifying Systems

The presence of noise in the snapshots has a significant impact in the ability for DMD or the proposed IOROM approach to fit the snapshots to a matrix  $A$  that captures the dynamics of the dataset. Some work has been done in analyzing the effects of noise with DMD [98,99]. There has been some additional work on computing the noise bias of least-squares solutions [100]. This section will address the impacts of process noise on identifying the dynamics of the system. This process noise could arise from disturbances in the system or nonlinearities. In particular, this section will illustrate the effects of adding an external forcing to the system (the IOROM approach) to aid in identifying the dominant characteristics of the system. For this analysis, only the evolution equation will be considered.

Consider a system (4.35) that has process noise such that

$$x_{k+1} = Ax_k + Bu_k + n_k, \quad (4.44)$$

where  $n_k \in \mathbb{R}^{n_x}$  is the process noise. For simplicity, it will be assumed that the noise is Gaussian. Assume that snapshots of the noise can be recorded:  $N_0 = \begin{bmatrix} n_0 & \dots & n_{n_s-1} \end{bmatrix}$ . The problem, using the state snapshot matrices  $X_0$  and  $X_1$  and the input snapshot matrix  $U_0$ , can be written as

$$X_1 = AX_0 + BU_0 + N_0. \quad (4.45)$$

If there were no noise and assuming  $B$  is known,  $A_{\text{dmd}} = (X_1 - BU_0)X_0^\dagger$ , where  $\dagger$  indicates the pseudo-inverse. In the presence of noise, there is an extra term that is contributed by the noise such that

$$A_{\text{true}} = A_{\text{dmd}} - N_0X_0^\dagger. \quad (4.46)$$

To understand the effects of process noise on the error between the identified  $A$  matrix and the true  $A$  matrix, the expected value of (4.46) can be computed as

$$E[A_{\text{dmd}} - A_{\text{true}}] = E \left[ N_0X_0^\dagger \right]. \quad (4.47)$$

The snapshot matrix  $X_0$  has noise embedded in the snapshots. The snapshots in  $X_0$  can

be written so that they are in terms of the initial condition,  $x_0$

$$\underbrace{\begin{bmatrix} x_0^T \\ x_1^T \\ \vdots \\ x_{n_s-1}^T \end{bmatrix}}_{X_0^T} = \underbrace{\begin{bmatrix} I_n \\ A \\ \vdots \\ A^{n_s-1} \end{bmatrix}}_{A_1^T} x_0^T + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ B & 0 & & 0 \\ \vdots & & \ddots & \vdots \\ BA^{n_s-2} & \dots & B & 0 \end{bmatrix}}_{B_2^T} \underbrace{\begin{bmatrix} u_0^T \\ u_1^T \\ \vdots \\ u_{n_s-1}^T \end{bmatrix}}_{U_0^T} \\ + \underbrace{\begin{bmatrix} 0 & 0 & \dots & 0 \\ I & 0 & & 0 \\ \vdots & & \ddots & \vdots \\ A^{n_s-2} & \dots & I & 0 \end{bmatrix}}_{B_1^T} \underbrace{\begin{bmatrix} n_0^T \\ n_1^T \\ \vdots \\ n_{n_s-1}^T \end{bmatrix}}_{N_0^T}.$$

The term  $N_0 X_0^\dagger$  can be written as a least-squares problem, i.e.,  $N_0 X_0^T (X_0 X_0^T)^{-1}$  where  $X_0^T$  can be substituted by the above equation. To simplify the analysis, an approximation can be used when noise is considered to be small such that  $(M + P)^{-1} = M^{-1} - M^{-1} P M^{-1}$  assuming  $M \gg P$ . This is a linearization of the term  $(X_0 X_0^T)$  such that the contributions of the noise to the error in identifying the  $A$  matrix can be analyzed. Only up to quadratic terms in  $N_0$  are kept. To simplify notation, the non-noise are collected and expressed as  $K = x_0 A_1 + U_0 B_2$ . The resulting expression of the expected value of  $A_{\text{true}} - A_{\text{dmd}}$  can be simplified to

$$E[A_{\text{dmd}} - A_{\text{true}}] = E[N_0 X_0^\dagger] \quad (4.48)$$

$$= E[N_0 K^T M^{-1}] + E[N_0 K^T M^{-1} K B_1^T M^{-1}] \quad (4.49)$$

$$+ E[N_0 K^T M^{-1} N_0 B_1 K^T M^{-1}] + E[N_0 B_1^T N_0^T M^{-1}] \quad (4.50)$$

$$= \sigma_n^2 (\text{tr}(K^T M^{-1} K B_1^T) I_{n_x} + M^{-1} K B_1 K^T) M^{-1}$$

where  $M = (K K^T)^{-1}$  and  $\sigma_n^2$  is the variance of the process noise. This indicates that if the variance of the noise is small, the mean error between  $A_{\text{dmd}}$  and  $A_{\text{true}}$  is small. Note that if the input term ( $U_0$ ) is set to zero, then the result for standard DMD is recovered. The error is influenced by the selection of the input. The influence becomes more apparent by analyzing the squared error of the identified  $A$  matrix and the true  $A$  matrix. Again, noise is assumed to be small and only up to the quadratic terms in  $N_0$  are kept

$$(A_{\text{dmd}} - A_{\text{true}})^2 = (N_0 K^T M^{-1})^2. \quad (4.51)$$

After some algebraic manipulation, the expected value of the squared error can be computed

as

$$E[(A_{\text{dmd}} - A_{\text{true}})^2] = \sigma_n^2 \text{tr}(M^{-1}). \quad (4.52)$$

This indicates that the inputs,  $U_0$  and the initial condition are important in contributing to this squared error term. It appears that the larger the amount of energy in the input and the initial condition, the smaller the squared error between the identified  $A$  and the true  $A$  matrix. The effects of the input and the initial condition can be seen when evaluating a one-state system,

$$X_1 = aX_0 + bU_0 + N_0. \quad (4.53)$$

The variance in identifying the  $a_{\text{true}}$  and the  $a_{\text{dmd}}$  of this one-state system can be computed as

$$E[(a_{\text{dmd}} - a_{\text{true}})^2] = \sigma_a^2 = \sigma_n^2 \text{tr} \left( ((x_0 A_1 + U_0 B_2)(A_1^T x_0 + B_2^T U_0^T))^{-1} \right) \quad (4.54)$$

$$= \frac{\sigma_n^2}{x_0^2 + \sum_{k=1}^{n_s-1} \left( a^k x_0 + \sum_{m=1}^k a^{k-m} b u_{m-1} \right)^2}. \quad (4.55)$$

The variance computed in this one state example using the IOROM approach is dependent on the variance of the noise. When the variance of the noise increases, the variance of the system,  $a$ , increases. In addition, if the energy in the initial condition,  $x_0$  increases, the variance will decrease. The benefit of using this IOROM approach can be seen in the input term,  $u$ . If an appropriate input is selected, then the variance of the system decreases. This can be useful in identifying a low-dimensional model for a high-dimensional system. The main drawback of this IOROM approach is that the system must be of the form where it is easily excitable by a controllable input.

#### 4.10 Analyzing the DMD modes

The DMD modes computed using the proposed IOROM approach are influenced by the choice of the external forcing on the system. The next subsections will detail the influence of the input on the selection of the DMD modes. This will follow the sparsity-promoting framework for standard DMD presented in [73].

#### 4.10.1 Optimal Amplitudes of DMD modes

As is done in standard DMD, an eigenvalue decomposition of the  $F$  matrix, obtained using the IOROM approach, can be used to construct DMD modes at a specific temporal frequency

$$F = \underbrace{\begin{bmatrix} \psi_1 & \dots & \psi_r \end{bmatrix}}_{\Psi} \underbrace{\begin{bmatrix} \mu_1 & & \\ & \ddots & \\ & & \mu_r \end{bmatrix}}_{D_\mu} \underbrace{\begin{bmatrix} \zeta_1^T \\ \vdots \\ \zeta_r^T \end{bmatrix}}_{Z^T}, \quad (4.56)$$

where  $\Psi$  is a matrix that contains the left eigenvectors,  $Z$  is a matrix that contains the right eigenvectors, and  $D_\mu$  contains the associated eigenvalues. The solution to the reduced-order system (4.36) is given by

$$z_t = F^t z_0 + \sum_{m=0}^{t-1} F^{t-m-1} G u_m \quad (4.57)$$

$$= (\Psi D_\mu^t Z^T) z_0 + \sum_{m=0}^{t-1} (\Psi D_\mu^{t-m-1} Z^T) G u_m. \quad (4.58)$$

Equivalently, the modal contribution of the initial condition and the input to the state  $z_t$  is captured by

$$z_t = \sum_{i=1}^r \psi_i \mu_i^t \underbrace{\zeta_i^T z_0}_{\alpha_i} + \sum_{m=0}^{t-1} \sum_{i=1}^r \psi_i \mu_i^{t-m-1} \underbrace{\zeta_i^T G u_m}_{\beta_i^T}, \quad (4.59)$$

where  $\alpha_i$  and  $\beta_i$  determine the contribution of the  $i$ th mode to the response.

The DMD modes can be expressed as  $\phi_i := U_r \psi_i$  where  $U_r$  contains the POD modes. The state of the full-order system can be approximated by (4.41). Thus, the solution to the full-order state can be written as

$$\begin{aligned} x_t &= U_r \left( \sum_{i=1}^r \psi_i \mu_i^t \alpha_i \right) + U_r \left( \sum_{m=0}^{t-1} \sum_{i=1}^r \psi_i \mu_i^{t-m-1} \beta_i^T u_m \right) \\ &= \sum_{i=1}^r \phi_i \mu_i^t \alpha_i + \sum_{m=0}^{t-1} \sum_{i=1}^r \phi_i \mu_i^{t-m-1} \beta_i^T u_m. \end{aligned}$$

By expressing the matrix of snapshots  $X_0$  as

$$X_0 = \Phi \sum_{i=1}^r \underbrace{\begin{bmatrix} \mu_i^t & \sum_{m=0}^{t-1} \mu_i^{t-m-1} u_m^T \end{bmatrix}}_{q_i^T(t)} \underbrace{\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}}_{\eta_i}. \quad (4.60)$$

The spectral coefficients  $\alpha_i$  and  $\beta_i$  capture contribution of individual DMD modes. Note that columns of the matrix  $\Phi$  are determined by the DMD modes  $\phi_i$ . Equivalently, in the vector form

$$\underbrace{\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n_s-1} \end{bmatrix}}_{\tilde{\Psi}} = \begin{bmatrix} U_r (\sum_{i=1}^r \psi_i q_i^T(0) \eta_i) \\ U_r (\sum_{i=1}^r \psi_i q_i^T(1) \eta_i) \\ \vdots \\ U_r (\sum_{i=1}^r \psi_i q_i^T(N-1) \eta_i) \end{bmatrix}, \quad (4.61)$$

or

$$\tilde{X} = \text{diag}(U_r) \mathcal{Q} \eta$$

where  $\text{diag}(U_r)$  is a block-diagonal matrix containing the matrix  $U_r$  on the main diagonal and

$$\mathcal{Q} = \begin{bmatrix} \psi_1 q_1^T(0) & \psi_2 q_2^T(0) & \dots & \psi_r q_r^T(0) \\ \psi_1 q_1^T(1) & \psi_2 q_2^T(1) & \dots & \psi_r q_r^T(1) \\ \vdots & & \ddots & \vdots \\ \psi_1 q_1^T(n_s-1) & \psi_2 q_2^T(n_s-1) & \dots & \psi_r q_r^T(n_s-1) \end{bmatrix}$$

and

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_r \end{bmatrix}.$$

The vector of amplitudes of the DMD modes,  $\eta$ , can be found as the solution to the least-squares problem

$$\underset{\eta}{\text{minimize}} \quad J(\eta) := \left\| \tilde{X} - \text{diag}(U_r) \mathcal{Q} \eta \right\|_2^2 \quad (4.62)$$

and it is determined by

$$\eta_{\text{dmd}} = (\mathcal{Q}^T \mathcal{Q})^{-1} \mathcal{Q}^T \text{diag}(U_r^T) \tilde{X}. \quad (4.63)$$

#### 4.10.2 Sparsity-Promoting DMD for Systems with Inputs

Next, the problem of identifying a subset of DMD modes that strikes an optimal balance between fidelity to available data and model complexity is addressed. In contrast to POD modes, DMD modes are not orthogonal, and there is no natural ordering. Dominant modes can be identified by solving the regularized least-squares problem where the regularization term is introduced as a proxy for inducing sparsity. This approach represents an extension of the sparsity-promoting DMD algorithm [73] with the main difference being that it is desired to promote block sparsity instead of elementwise sparsity [101].

Consider the regularized optimization problem

$$\underset{\eta}{\text{minimize}} \quad J(\eta) + \gamma \sum_{i=1}^r \|\eta_i\|_2, \quad (4.64)$$

where the Euclidean norm of the DMD amplitudes  $g(\eta) := \sum_i \|\eta_i\|_2$  is introduced to promote block sparsity. A similar approach is typically used in statistics literature to drive a set of variables (in this case, coefficients  $\alpha_i$  and  $\beta_i$ ) jointly to zero.

Once this optimization problem is solved, a sparsity structure is fixed based on the non-zero coefficients that determine the contribution of each DMD mode to a particular snapshot. Specifically, the “polished” amplitudes are found by solving a constrained quadratic program

$$\begin{aligned} &\underset{\eta}{\text{minimize}} \quad J(\eta) \\ &\text{subject to} \quad E^T \eta = 0, \end{aligned} \quad (4.65)$$

where  $E$  provides information about the sparsity structure of the coefficients contained in  $\eta$ .

### 4.10.3 Alternating Direction Method of Multipliers Algorithm

The algorithm used to solve (4.64) is based on Alternating Direction Method of Multipliers (ADMM) [102]. The objective function in optimization problem (4.64) can be rewritten as:

$$J(\eta) = \eta^T P \eta - \tilde{q}^T \eta - \eta^T \tilde{q}, \quad (4.66)$$

where  $P := Q^T Q$  and  $\tilde{q} = Q^T \text{diag}(U_r^T) \tilde{X}$ . This is a convex optimization problem that can be solved using standard algorithms, such as ADMM. To bring the problem into a form amenable to ADMM, an auxiliary optimization variable  $\varepsilon$  is introduced,

$$\begin{aligned} &\underset{\eta, \varepsilon}{\text{minimize}} \quad J(\eta) + \gamma g(\varepsilon) \\ &\text{subject to} \quad \eta - \varepsilon = 0 \end{aligned} \quad (4.67)$$

and define the augmented Lagrangian

$$\begin{aligned} \mathcal{L}_\rho(\eta, \varepsilon, \rho) &:= J(\eta) + \gamma g(\varepsilon) \\ &+ \frac{1}{2} (\lambda^T (\eta - \varepsilon) + (\eta - \varepsilon)^T \lambda + \rho \|\eta - \varepsilon\|_2^2). \end{aligned} \quad (4.68)$$

Here,  $\lambda$  is a vector of Lagrange multipliers and  $\rho$  is a positive parameter. ADMM minimizes the augmented Lagrangian separately with respect to  $x$  and  $z$  followed by a dual ascent

update of the Lagrange multiplier  $\lambda$ ,

$$\begin{aligned}\eta^{k+1} &= \underset{\eta}{\operatorname{argmin}} \mathcal{L}_\rho(\eta, \varepsilon^k, \lambda^k) \\ \varepsilon^{k+1} &= \underset{\varepsilon}{\operatorname{argmin}} \mathcal{L}_\rho(\eta^{k+1}, \varepsilon, \lambda^k) \\ \lambda^{k+1} &= \lambda^k + \rho(\eta^{k+1} - \varepsilon^{k+1}).\end{aligned}\tag{4.69}$$

It can be shown that  $\eta$  and  $\varepsilon$  can be explicitly updated

$$\begin{aligned}\eta^{k+1} &= (P + (1/\rho)I)^{-1} \left( \tilde{q} + \frac{\rho}{2} \left( \varepsilon^k - \frac{1}{\rho} \lambda^k \right) \right) \\ \varepsilon^{k+1} &= \begin{cases} \left( 1 - \frac{a}{\|v^k\|_2} \right) v^k, & \|v^k\|_2 > a \\ 0, & \|v^k\|_2 < a. \end{cases}\end{aligned}$$

where  $a = \gamma/\rho$  and  $v^k = \eta^{k+1} + (1/\rho)\lambda^k$ .

#### 4.11 Example: Linearized Channel Flow

This chapter uses the three-dimensional incompressible linearized Navier-Stokes equations in a channel flow to illustrate these developments; see Fig. 4.1 for geometry. The application of the Fourier Transform in the streamwise ( $x$ ) and the spanwise ( $z$ ) directions along with the use of a pseudospectral method in the wall-normal direction yields the finite-dimensional state-space representation

$$\dot{x}(k_x, k_z, t) = A(k_x, k_z)x(k_x, k_z, t) + B(k_x, k_z)u(k_x, k_z, t),\tag{4.70}$$

where  $k_x$  and  $k_z$  are the wave numbers in the  $x$  and  $z$  direction,  $x$  is the state and  $u$  is a spatially distributed and temporally varying body forcing. This system governs the dynamics of infinitesimal flow fluctuations around the parabolic velocity profile  $U(y) = 1 - y^2$ . In what follows, the Reynolds number is fixed to  $Re = 2000$  and is confined to a pair of horizontal wavenumbers ( $k_x = 1$ ,  $k_z = 1$ ). In the wall-normal direction, 200 collocation points are used, resulting in 400 total states. This system is advanced in time and snapshots are recorded with  $\Delta t = 1$ . A random initial condition that satisfies proper boundary conditions in the wall-normal direction is used and all numerical computations are performed in MATLAB. For additional details about the linearized Navier-Stokes equations, the reader is referred to [103].

##### 4.11.1 DMD vs. DMD with Exogenous Inputs

First, a reduced-order model of the channel flow problem was constructed using standard DMD and the IOROM approach to demonstrate the benefit of adding an exogenous input

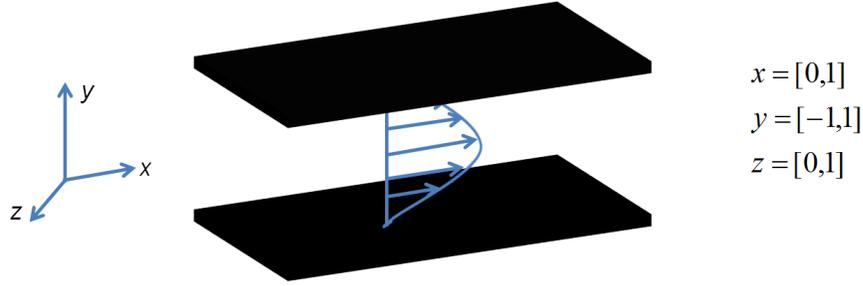


Figure 4.1: Depiction of the channel flow problem.

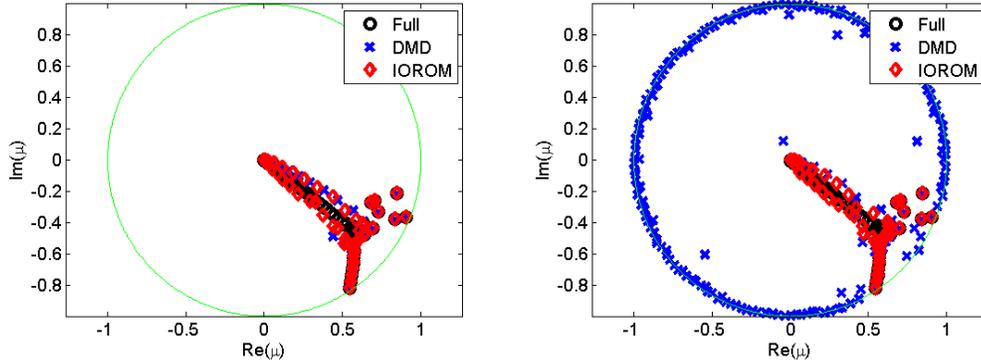


Figure 4.2: (Left) Eigenvalues obtained with standard DMD and the IOROM approach. (Right) Eigenvalues in the presence of noise with variance  $10^{-11}$ .

to the system. For the IOROM approach, the system is excited for 100 seconds using a streamwise body forcing that enters as a temporal chirp excitation in the middle of the channel, i.e.  $y = 0$ . This results in 100 snapshots that are used as a basis for system identification.

Figure 4.2 shows the eigenvalues of the full-order  $A$  operator, the  $A$  operator identified by DMD, and the  $A$  operator identified by the IOROM approach. This approach can be used to identify the eigenvalues of the system in the same way as standard DMD. The input excitation provides additional energy to the system to aid in identifying the dynamics. In particular, the IOROM approach can handle small amounts of process noise. Figure 4.2 compares the eigenvalues resulting from standard DMD and the IOROM approach for the system subject to process noise with a variance of  $10^{-11}$ . The IOROM approach is more robust than standard DMD. This is in concert with the results in system identification literature [69], where it has been shown that the choice of input plays an important role in the ability to characterize the dynamics of the system.

It should be noted that this particular example is sensitive to small amounts of noise. When

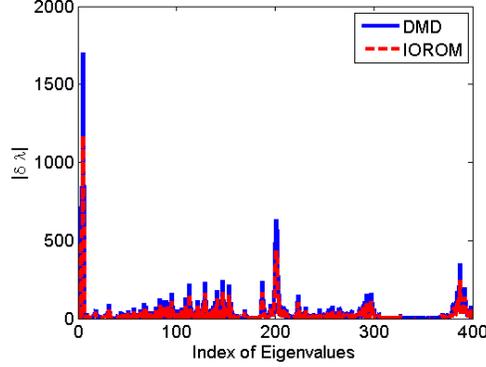


Figure 4.3: This figure shows the sensitivity of the eigenvalues,  $\delta\lambda$ , using the DMD and IOROM approaches for the channel flow problem. The values plotted are the upper bound.

evaluating the full-order  $A$  matrix, it can be seen that many of the eigenvalues are clustered around zero. The sensitivity of the eigenvalues can be evaluated by their condition number defined as

$$c = \frac{1}{|\zeta^T \psi|} \quad (4.71)$$

where  $\zeta$  and  $\psi$  are the left and right eigenvectors corresponding to the same eigenvalue, respectively [104]. Large condition numbers of eigenvalues indicate that they are very sensitive to small perturbations in the  $A$  matrix. The additional excitation signal provided by the IOROM approach, on average, decreases the error between the full-order  $A$  operator and the identified  $A$  operator. The sensitivity of the eigenvalues to small perturbations in the  $A$  matrix can be bounded by

$$\delta\lambda \leq \frac{\|\delta A\|}{|\zeta^T \psi|}, \quad (4.72)$$

where  $\delta\lambda$  denote the variations in eigenvalues and  $\delta A$  denote the differences in the identified  $A$  matrix (using DMD and the IOROM approach) and the true  $A$  matrix. Figure 4.3 shows the upper bound of the variations in the eigenvalues. The  $A$  matrix identified by DMD results in higher variations in the eigenvalues than the  $A$  matrix identified by the IOROM approach. This is due to the presence of an excitation signal. It should be noted that the IOROM approach still results in large variations; however, choosing the appropriate excitation signal can decrease those variations significantly.

A number of simulations were run (between 10 and 100 simulations) and the  $A$  matrices were identified using DMD and the IOROM approach. The resulting  $A$  matrices were averaged and the eigenvalues were computed from the averaged  $A$  matrix. Figure 4.4 shows that as the identified  $A$  matrix is averaged over more simulations, the eigenvalues identified with DMD and the IOROM approach are moving toward the eigenvalues of the full-order system, especially when the noise variance is small. Here the noise variance is  $10^{-11}$ .

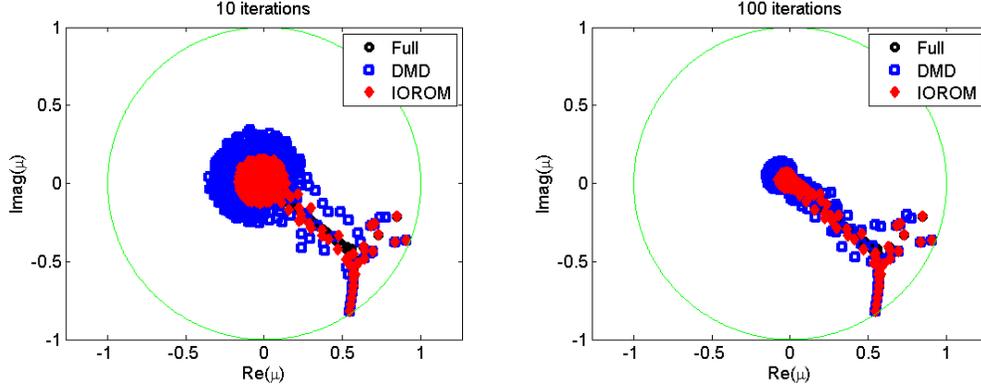


Figure 4.4: (Left) Average eigenvalues identified by DMD and the IOROM approach after 10 simulations. (Right) Average eigenvalues for 100 simulations.

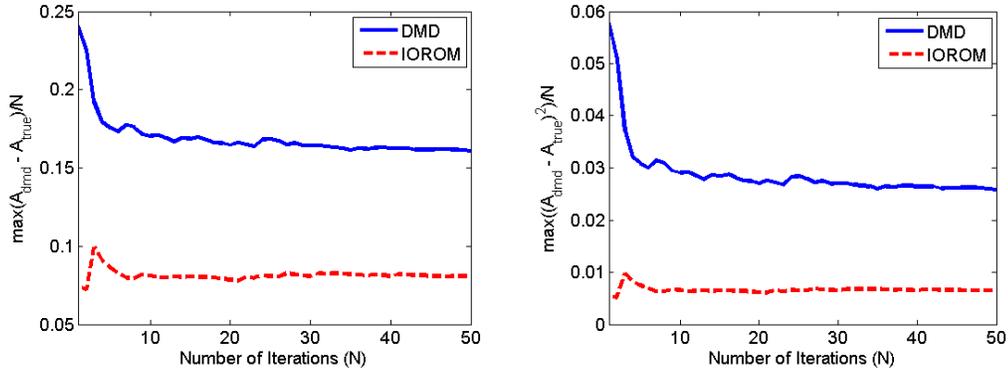


Figure 4.5: (Left) This figure shows the average maximum error between entries in the full-order  $A$  matrix and the  $A$  matrices identified by the DMD (blue) and the IOROM approach (red). (Right) This figure shows the average maximum variance between entries in the full-order  $A$  matrix and the  $A$  matrices identified by the DMD (blue) and the IOROM approach (red).

However, there is still a bias in the error between the identified  $A$  matrix and the true  $A$  matrix, see Figure 4.5 (left). Figure 4.5 (left) shows the mean maximum error between entries in the full-order  $A$  matrix and the  $A$  matrices identified by DMD (blue) and the IOROM approach (red). Similarly, the maximum variance between entries in the identified  $A$  matrix and the full-order  $A$  matrix is shown in Figure 4.5 (right). This agrees with the analysis in Section 4.9. The bias is reduced when an appropriate excitation signal is chosen.

#### 4.11.2 Sparsity-Promoting DMD for Systems with Inputs

Next, the dominant DMD modes are identified using the sparsity-promoting DMD algorithm for systems with inputs. For this setup, the rank of the snapshots matrix,  $X_0$ , is  $r = 50$ . The

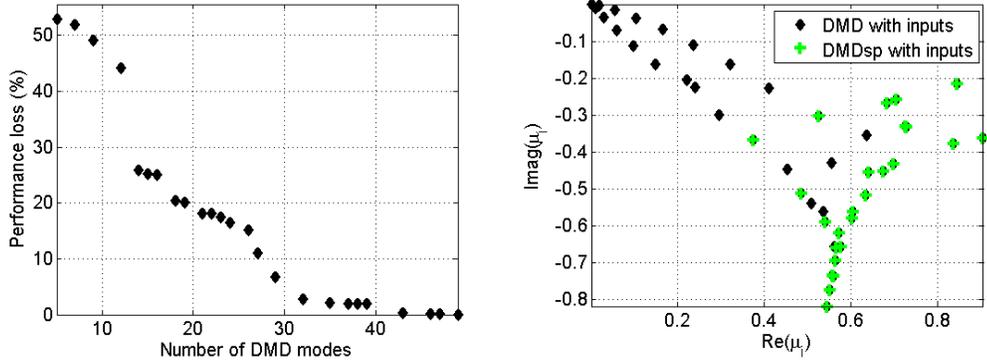


Figure 4.6: (Left) Performance loss with respect to the number of DMD modes selected. (Right) Eigenvalues that are omitted when using the sparsity promoting approach. This is shown for  $N_z = 31$ .

sparsity-promoting algorithm eliminates the DMD modes that have a weak contribution to the available data.

Figure 4.6 (left) shows the performance loss as the sparsity-promoting algorithm eliminates DMD modes. The performance loss (PL) is computed by:

$$PL(\%) := 100 \sqrt{\frac{J(\eta)}{J(0)}}. \quad (4.73)$$

There is minimal loss in performance using only 35 DMD modes ( $< 1\%$ ). As the number of DMD modes get smaller the performance loss increases. For example, reducing the number of DMD modes to 25 introduces a 15% performance loss.

Figure 4.6 (right) shows the eigenvalues that are preserved with 31 DMD modes. This number offers a performance loss of only 7%. The number of DMD modes retained is determined by the number of non-zero  $\eta_i$  vectors where each  $\eta_i$  vector contains the spectral coefficients associated with a particular mode. A vector,  $\eta_i$  is considered non-zero if at least one of its entries is non-zero. In other words, all of the elements of the vector  $\eta_i$  have to be zero for a DMD mode to be removed. These eigenvalues are associated with the DMD modes that can capture the essential dynamics of the system.

The tradeoff between model performance and the number of DMD modes kept can be analyzed by changing the regularization parameter  $\gamma$ . As  $\gamma$  increases, more emphasis is placed on sparsity rather than model performance. This is shown in Figure 4.7. In particular, Figure 4.7 (left) demonstrates that as  $\gamma$  increases, the number of non-zero vectors  $\eta_i$  decreases. The number of non-zero vectors is associated with the number of DMD modes retained. As the number of DMD modes decreases, it is expected that the performance

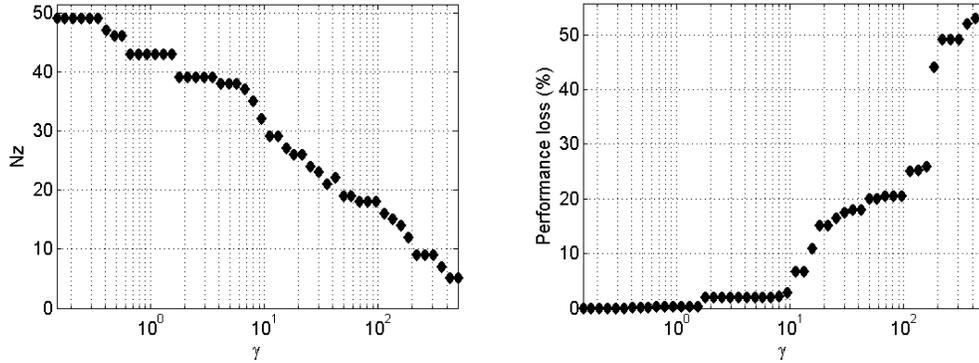


Figure 4.7: (Left) Non-zero (Nz)  $x_i$  vectors as  $\gamma$  increases. The number of non-zero vectors correspond to the number of DMD modes retained. (Right) Performance loss as  $\gamma$  increases.

of the reduced-order model also decreases. Figure 4.7 (right) shows the performance loss as  $\gamma$  increases. This indicates that increasing sparsity reduces quality of approximation of available snapshots.

## 4.12 Conclusions

This chapter introduced an extension to DMDc that can construct an input-output reduced-order model that can be used for high-dimensional systems. This approach takes advantage of characterizing the dominant dynamics in the flow and provides a low-order approximation of the flow. Using this low-order approximation, a reduced-order model can be constructed that retains the input-output behavior seen in the full-order model. This reduced-order model has a low computational cost and contains the necessary dynamics that are important for problems such as wind farm control. This technique avoids computing the adjoint, as is done in BPOD, and the states of the reduced-order model maintain some physical meaning. In addition, this IOROM technique is able to maintain robustness to small amounts of process noise.

Lastly, this chapter presented the sparsity-promoting DMD technique, which has been extended to include exogenous inputs to the system. This approach was demonstrated on the linear channel flow example and was compared to the standard DMD approach. It can be seen that the addition of an external input can improve the performance of the reduced-order model with minimal loss in model performance. It was also shown that the addition of an exogenous input to the system may impact the selection of DMD modes. The next chapter will extend this IOROM technique to parameter varying systems which is necessary for the wind farm problem.

## Chapter 5

# Parameter Varying Reduced-Order Modeling

### 5.1 Introduction

This chapter describes an extension of the IOROM approach, described in Section 4.8, to construct reduced-order linear parameter varying (LPV) models that approximate high-order nonlinear models. The nonlinear system is assumed to have a parameterized collection of equilibrium operating points. For example, the free-stream wind speed and wind direction parameterize the equilibrium condition in a wind farm. Related work on this particular topic has been done looking at flexible aircraft [26, 105, 106]. Many studies use linear methods such as Krylov methods [106, 107]. The proposed LPV approach involves two steps. The first step, described in Section 4.8, uses POD [18–20] and direct subspace identification [16] to construct an input-output reduced-order model (IOROM). This step is similar to DMDc [25], which can be used to construct a reduced-order linear model at one operating condition. Second, the reduced-order models constructed at fixed operating conditions are “stitched” together using a parameter varying linearization (discussed in Section 5.2). The key technical issue is that the states of the reduced-order model must have a consistent meaning across all operating conditions as described in Section 5.3. This state consistency issue has been addressed in other studies [108]. The difference in this chapter is that the parameter is allowed to vary in time for a given simulation. The IOROM approach handles this issue by constructing a single reduced-order subspace that is used at all operating conditions. This approach and the LPV linearization method are demonstrated on a nonlinear mass-spring-damper example with 200 states in Section 5.5.

## 5.2 Linearization for Parameter Varying Systems

To introduce the parameter varying nature to the nonlinear system described in (4.32), a time-varying parameter  $\rho$  is added, describing the operating condition, to the system such that the system is now defined as

$$\begin{aligned}x_{k+1} &= f(x_k, u_k, \rho_k) \\ y_k &= h(x_k, u_k, \rho_k).\end{aligned}\tag{5.1}$$

The proposed approach in Chapter 4 can be extended to this case using a parameter varying linearization similar to the one described in [109].

Assume there is a collection of equilibrium points  $(\bar{x}(\rho), \bar{u}(\rho), \bar{y}(\rho))$  such that

$$\begin{aligned}\bar{x}(\rho) &= f(\bar{x}(\rho), \bar{u}(\rho), \rho) \\ \bar{y}(\rho) &= h(\bar{x}(\rho), \bar{u}(\rho), \rho).\end{aligned}\tag{5.2}$$

If the state is initialized at  $x_0 = \bar{x}(\rho)$ , the input is held fixed at  $u_k = \bar{u}(\rho)$ , and the operating condition is frozen at  $\rho_k = \rho$  then the state and output will remain in equilibrium at  $x_k = \bar{x}(\rho)$  and  $y_k = \bar{y}(\rho)$  for  $k = 0, 1, \dots$ . Thus  $(\bar{x}(\rho), \bar{u}(\rho), \bar{y}(\rho))$  defines an equilibrium condition for each *fixed* value of  $\rho$ .

The nonlinear dynamics can be linearized around the equilibrium points defined for each fixed value of  $\rho$ . Define perturbations from the equilibrium condition as

$$\delta x_k := x_k - \bar{x}(\rho), \quad \delta u_k := u_k - \bar{u}(\rho), \quad \delta y_k := y_k - \bar{y}(\rho).\tag{5.3}$$

For a fixed operating condition ( $\rho_k = \rho$  for  $k = 0, 1, \dots$ ), a standard linearization yields

$$\begin{aligned}\delta x_{k+1} &= A(\rho)\delta x_k + B(\rho)\delta u_k \\ \delta y_k &= C(\rho)\delta x_k + D(\rho)\delta u_k,\end{aligned}\tag{5.4}$$

where the linearized state matrices are defined by

$$\begin{aligned}A(\rho) &:= \left. \frac{\partial f}{\partial x} \right|_{(\bar{x}(\rho), \bar{u}(\rho), \rho)}, & B(\rho) &:= \left. \frac{\partial f}{\partial u} \right|_{(\bar{x}(\rho), \bar{u}(\rho), \rho)} \\ C(\rho) &:= \left. \frac{\partial h}{\partial x} \right|_{(\bar{x}(\rho), \bar{u}(\rho), \rho)}, & D(\rho) &:= \left. \frac{\partial h}{\partial u} \right|_{(\bar{x}(\rho), \bar{u}(\rho), \rho)}.\end{aligned}\tag{5.5}$$

Next consider the case where the operating condition, specified by the parameter  $\rho_k$ , varies in time. In general,  $(x_k, u_k, y_k) := (\bar{x}(\rho_k), \bar{u}(\rho_k), \bar{y}(\rho_k))$  is not a valid solution of the nonlinear system. In other words, the parameterized state/input/output values only define an

equilibrium condition for fixed values of  $\rho$ . Despite this fact, it is still possible to construct a time varying linearization around the parameterized values  $(\bar{x}(\rho), \bar{u}(\rho), \bar{y}(\rho))$ . Re-define perturbation variables as follows for the case where  $\rho$  varies in time

$$\delta x_k := x_k - \bar{x}(\rho_k), \quad \delta u_k := u_k - \bar{u}(\rho_k), \quad \delta y_k := y_k - \bar{y}(\rho_k). \quad (5.6)$$

A valid (time-varying) linearization can be obtained when  $\rho_k$  is time-varying as follows

$$\delta x_{k+1} = f(x_k, u_k, \rho_k) - \bar{x}(\rho_{k+1}). \quad (5.7)$$

A Taylor series expansion of  $f(x_k, u_k, \rho_k)$  yields

$$f(\bar{x}_k(\rho_k) + \delta x_k, \bar{u}_k(\rho_k) + \delta u_k, \rho_k) \approx \bar{x}(\rho_k) + A(\rho_k)\delta x_k + B(\rho_k)\delta u_k. \quad (5.8)$$

where  $A$  and  $B$  are as defined in (5.5). A similar Taylor series approximation can be performed to linearize the output function  $h$  (5.1) in terms of the matrices  $C$  and  $D$  defined in (5.5). Combining (5.7) and (5.8) leads to the following parameter varying linearization

$$\begin{aligned} \delta x_{k+1} &= A(\rho_k)\delta x_k + B(\rho_k)\delta u_k + (\bar{x}(\rho_k) - \bar{x}(\rho_{k+1})) \\ \delta y_k &= C(\rho_k)\delta x_k + D(\rho_k)\delta u_k. \end{aligned} \quad (5.9)$$

This differs from the standard linearization at a single fixed operating point in two respects. First,  $\rho_k$  varies in time and hence this is a time-varying system. More precisely, the time variations in the state matrices  $(A, B, C, D)$  arise due to  $\rho_k$  and hence this is called a linear parameter varying (LPV) system. There are many tools in the controls literature that address this class of systems [110–113]. Second, the equilibrium state varies in time due to the changing operating condition. This effect is retained by the term  $\bar{x}(\rho_k) - \bar{x}(\rho_{k+1})$  which provides a forcing term for the dynamics. This model linearizes the dependence on the state and the input. The linearization approximation is accurate as long as the state, input, and output remain near the manifold of equilibrium points  $(\bar{x}(\rho_k), \bar{u}(\rho_k), \bar{y}(\rho_k))$ . It is important that the nonlinear dependence on the operating condition (specified by  $\rho_k$ ) is retained.

### 5.3 State Consistency Issue

Parameter varying linearizations can be constructed using data from steady operating conditions specified by constant values of  $\rho$ . Specifically, the linearization only requires state matrices  $(A(\rho), B(\rho), C(\rho), D(\rho))$  and equilibrium conditions  $(\bar{x}(\rho), \bar{u}(\rho), \bar{y}(\rho))$  at each fixed value of  $\rho$ . In principal, the proposed IOROM method or another model reduction method can be used to identify reduced-order state matrices at each operating point,  $\rho$ . One key

technical issue is state consistency.

To clarify this issue, consider an autonomous system without inputs and outputs. Let  $A(\rho)$  denote the state matrix that appears in the parameter varying linearization. For example, DMD can be used at each fixed value of  $\rho$  to identify a subspace spanned by  $Q(\rho) \in \mathbb{R}^{n_x \times r(\rho)}$  and a reduced-order matrix  $F(\rho) \in \mathbb{R}^{r(\rho) \times r(\rho)}$  such that  $A(\rho) \approx Q(\rho)F(\rho)Q(\rho)^T$ . The reduced-order state at the operating point  $\rho$  is defined as  $z := Q(\rho)^T \delta x \in \mathbb{R}^{r(\rho)}$ . This reduced-order state,  $z$ , will lack consistency if the projection subspace  $Q(\rho)$  depends on the parameter  $\rho$ . In other words, the meaning of  $z$  and dimension  $r(\rho)$  will depend on the value of  $\rho$ . Hence the state,  $z$ , at  $\rho_1$  will not, in general, be consistent in either meaning or dimension with the state  $z$  at  $\rho_2 \neq \rho_1$ .

To circumvent this issue, a single consistent subspace  $Q \in \mathbb{R}^{n_x \times r}$  should be used at all operating conditions. The reduced-order state  $z := Q^T \delta x \in \mathbb{R}^r$  then has a consistent meaning for all parameter values. Moreover, a reduced-order matrix  $F(\rho) \in \mathbb{R}^{r \times r}$  can be identified at each value of  $\rho$  such that  $A(\rho) \approx QF(\rho)Q^T$ . This would lead to a reduced-order parameter varying linearization of the form

$$z_{k+1} = F(\rho_k)z_k + (\bar{z}(\rho_k) - \bar{z}(\rho_{k+1})), \quad (5.10)$$

where  $\bar{z}(\rho) := Q^T \bar{x}(\rho)$  is the equilibrium state projected onto the reduced subspace. The next section presents a method to address this state consistency issue.

## 5.4 Reduced-Order Parameter Varying Linearizations

The proposed parameter varying IOROM approach, shown in Algorithm 1, constructs a reduced-order model for a discrete-time nonlinear system (5.1) in four steps. First, a collection of parameter values are selected (Line 1). Second, data is collected from the nonlinear system at these selected parameter values (Lines 2-5). The collected data includes the equilibrium conditions as well as state/input/output snapshots obtained by exciting the nonlinear system. The algorithm, as written, assumes that the same number of snapshots  $n_s$  are obtained at each grid point. However, the number of snapshots can easily change with each grid point. Third, a single  $r$ -dimensional subspace of the state space is constructed (Lines 6-9). The subspace, defined by an orthogonal matrix  $Q \in \mathbb{R}^{n_x \times r}$ , is constructed from the POD modes of the snapshots obtained from *all* parameter values. Fourth, reduced-order state matrices are computed at each parameter vector using the IOROM approach described in Section 4.8 (Lines 10-11).

The outcome of Algorithm 1 is a single  $r$ -dimensional subspace  $Q$  as well as equilibrium conditions and reduced-order state matrices computed at the selected grid points. This

---

**Algorithm 1** Reduced-Order Parameter Varying Linearization
 

---

- 1: **Given:** Collection of parameter grid points  $\{p^j\}_{j=1}^{n_g}$ .
- 2: **Data Collection:** At each grid point  $j = 1, \dots, n_g$  do the following:
  - 3: **Equilibrium:** Compute the equilibrium condition  $(\bar{x}(p^j), \bar{u}(p^j), \bar{y}(p^j))$  at  $\rho = p^j$ .
  - 4: **Experiment:** Excite the nonlinear system (5.1) with fixed  $\rho_k = p^j$ . The initial condition  $x_0$  and input  $u_k$  should be near the equilibrium condition  $(\bar{x}(p^j), \bar{u}(p^j))$ .
  - 5: **Snapshots:** Define the matrices of snapshot deviations from the equilibrium conditions at  $p^j$ :

$$X_0(p^j) := [x_0 - \bar{x}(p^j), \dots, x_{n_s-1} - \bar{x}(p^j)] \in \mathbb{R}^{n_x \times n_s} \quad (5.11)$$

$$X_1(p^j) := [x_1 - \bar{x}(p^j), \dots, x_{n_s} - \bar{x}(p^j)] \in \mathbb{R}^{n_x \times n_s} \quad (5.12)$$

$$U_0(p^j) := [u_0 - \bar{u}(p^j), \dots, u_{n_s-1} - \bar{u}(p^j)] \in \mathbb{R}^{n_u \times n_s} \quad (5.13)$$

$$Y_0(p^j) := [y_0 - \bar{y}(p^j), \dots, y_{n_s-1} - \bar{y}(p^j)] \in \mathbb{R}^{n_y \times n_s} \quad (5.14)$$

- 6: **Construct Subspace for State Reduction:**

- 7: **Stack Data:** Define matrix of all state data:  $X_0 := [X_0(p^1), \dots, X_0(p^{n_g})] \in \mathbb{R}^{n_x \times (n_s n_g)}$ .

- 8: **POD Modes:** Compute POD modes of  $X_0$ .

- 9: **Subspace:** Choose  $r$  modes, denoted  $Q \in \mathbb{R}^{n_x \times r}$ , to capture sufficient energy in  $X_0$ .

- 10: **Reduced-Order State Matrices:** At each grid point  $j = 1, \dots, n_g$  do the following:
    - 11: **IOROM:** Use snapshot data  $(X_0(p^j), X_1(p^j), U_0(p^j), Y_0(p^j))$  and subspace  $Q$  to compute state matrices  $(F(p^j), G(p^j), H(p^j), D(p^j))$  via IOROM.
- 

yields a reduced-order parameter varying linearization of the nonlinear system of the form

$$\begin{aligned} z_{k+1} &= F(\rho_k)z_k + G(\rho_k)\delta u_k + (\bar{z}(\rho_k) - \bar{z}(\rho_{k+1})) \\ \delta y_k &= H(\rho_k)z_k + D(\rho_k)\delta u_k, \end{aligned} \quad (5.15)$$

where  $z := Q^T \delta x \in \mathbb{R}^r$  is the reduced-order state and  $\bar{z}(\rho) := Q^T \bar{x}(\rho) \in \mathbb{R}^r$  is the reduced-order equilibrium state. The single subspace defined by  $Q$  is used to construct state matrices at all parameter values. Hence the reduced-order state  $z$  retains a consistent meaning across the parameter domain. Note that Algorithm 1 only computes the state matrices and equilibrium conditions on a grid of specified parameter values. Interpolation (e.g., linear, spline, etc.) must be used to evaluate the state matrices and equilibrium conditions at any parameter value not contained in this grid. It is assumed that the grid of parameter values is sufficiently dense that this interpolation is accurate.

The parameter varying linearization can be used to approximate the response of the nonlinear system for an initial condition  $x_0$ , input  $u_k$ , and parameter trajectory  $\rho_k$ . Specifically,

---

**Algorithm 2** Hybrid POD/Gram-Schmidt Approach to Construct Subspace

---

- 1: **Given:** Collection of parameter grid points  $\{p^j\}_{j=1}^{n_g}$  and snapshots  $X_0(p^j)$  from each grid point.
  - 2: **Initial Point:** Use standard POD to compute  $r_1$  modes  $Q(p^1) \in \mathbb{R}^{n_x \times r_1}$  to capture sufficient energy in  $X_0(p^1)$ .
  - 3: **Iterative Processing:** For  $j = 2, \dots, n_g$ , iteratively compute additional modes at each grid point. Given  $Q_1 := [Q(p^1), \dots, Q(p^{j-1})]$ , use iterative POD to compute  $r_j$  additional modes  $Q(p^j) \in \mathbb{R}^{n_x \times r_j}$  to capture sufficient energy in  $X_0(p^j)$ .
  - 4: **Subspace:** Stack modes to form a single subspace defined by  $Q := [Q(p^1), \dots, Q(p^{n_g})]$ .
- 

the initial condition and input for the nonlinear system define a corresponding initial condition  $z_0 = Q^T(x_0 - \bar{x}(\rho_0))$  and input  $\delta u_k = u_k - \bar{u}(\rho_k)$  for the parameter varying system. The reduced-order, parameter varying linearization (5.15) can be simulated to obtain the state response  $z_k$  and output  $\delta y_k$ . These correspond to the state response  $x_k = Qz_k + \bar{x}(\rho_k)$  and output  $y_k = \delta y_k + \bar{y}(\rho_k)$  for the full-order nonlinear system.

The subspace construction step of Algorithm 1 (Lines 6-9) requires the SVD of the matrix  $X_0$  that contains the snapshot data from all the operating points. This matrix has  $n_s n_g$  columns and hence the SVD of  $X_0$  may be computationally intractable if there are many grid points. A suboptimal, but more computationally efficient approach, is to iteratively process the snapshot data from each grid point. The basic idea is to determine a set of modes  $Q(p^1)$  that capture the energy in the snapshots at the first grid point. Next, additional modes  $Q(p^2)$  are computed so that the combined set  $[Q(p^1), Q(p^2)]$  captures the energy in the snapshots at the second grid point. The procedure continues iteratively computing new modes  $Q_2 := Q(p^j)$  to combine with previously computed modes  $Q_1 := [Q(p^1), \dots, Q(p^{j-1})]$ . The benefit is that only snapshots obtained from one grid point are required for the calculations. This method can be thought of as a hybrid POD/Gram-Schmidt approach. Also, it is assumed that the snapshots from one simulation or experiment have comparable amounts of energy, which makes this approach reasonable. This iterative procedure requires a method to compute the optimal (new) modes  $Q_2$  that should be added to some given modes  $Q_1$ . Theorem 1 in the appendix provides a POD-type result to perform this iterative calculation. Algorithm 2 gives the detailed steps for the iterative subspace construction. This iterative method can replace the single-step method described in Lines 6-9 of Algorithm 1. The remaining steps of Algorithm 1 are unchanged even when combined with the iterative subspace calculation.

There are several benefits of the proposed reduced-order LPV models. The main benefit is that the models can be used for standard, gain-scheduled control. In addition, more formal control analysis and synthesis tools have been developed for LPV systems [110–112]. This LPV modeling approach can yield models that are accurate over a wide range of operating conditions for a nonlinear system. This is in contrast to the existing reduced-order modeling

approaches described in Chapter 4 which, for the most part, are used to construct a single linear model. In addition, the proposed method relies on input/output data from a forced response and does not require linearization of the system or the construction/simulation of the system adjoint as in balanced POD. Finally, the reduced-order model involves a projection onto a well-defined reduced-order subspace. This retains a physical meaning in the reduced-order states. In other words, the reduced-order state can be used to approximate the full-order state of the system. This provides insight into the key spatial modes of fluid/structure systems and is not simply a black-box identification technique, often seen in subspace identification. The remainder this chapter emphasizes this application of this method to develop LPV models for nonlinear systems.

## 5.5 Example: Nonlinear Mass-Spring-Damper

This section demonstrates the proposed parameter varying IOROM approach on a nonlinear mass-spring-damper example. The purpose of this model is to show how this method is implemented in detail as well as the feasibility of this method. A more complex example is implemented in Chapter 6.

### 5.5.1 Model Formulation: 1-Block

A simple nonlinear mass-spring-damper example is presented to illustrate the parameter varying linearization method described in Section 5.2. A single mass, shown in Figure 5.1, is connected to a wall by a linear damper and nonlinear spring. Two forces,  $F$  and  $u$ , are applied to the block. Newton's second law yields the following continuous-time, nonlinear model

$$\frac{d}{dt} \begin{bmatrix} p \\ v \end{bmatrix} = \begin{bmatrix} v \\ \frac{1}{m} (F + u - bv - k_s(p)p) \end{bmatrix}. \quad (5.16)$$

where  $p$  is the position of the block (units of m),  $v$  is the velocity (m/s),  $b$  is the (linear) damping constant (in N/(m/s)), and  $k_s(p)$  is a nonlinear spring constant (in N/m). The spring constant has the form  $k_s(p) = k_1 + k_2 p^2$  and is assumed to be a stiffening spring  $k_2 > 0$ . The corresponding values for these system constants are given in Figure 5.1. Finally, the block has two forces: an exogenous (disturbance) force  $F$  [N] and a controllable input force  $u$  [N].

A continuous-time parameter varying linearization can be performed. Instead, the model is discretized to align with the discrete-time development used in this chapter. A simple

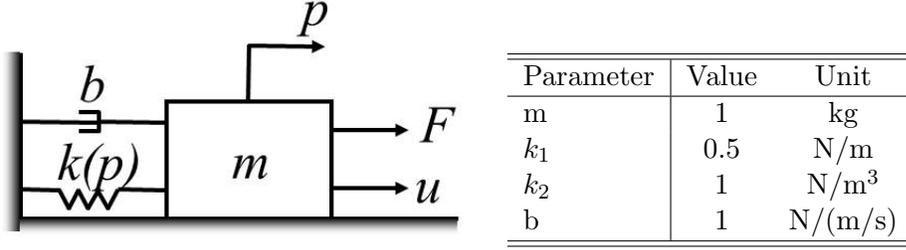


Figure 5.1: Nonlinear mass-spring-damper (Left) and system constants (Right).

Euler approximation with sample time  $\Delta t$  yields a two-state discrete-time model

$$x_{k+1} = f(x_k, u_k, F_k) \quad (5.17)$$

where  $x := [p, v]^T$  is the state and the discretized dynamics are given by:

$$f(x_k, u_k, F_k) := x_k + \Delta t \begin{bmatrix} v_k \\ \frac{1}{m} (F_k + u_k - b v_k - k_s(p_k) p_k) \end{bmatrix}. \quad (5.18)$$

For simplicity, the outputs for this model will be the full state, i.e.  $y := x$ . The disturbance force  $F_k$  is treated as the varying parameter in the dynamics:  $\rho_k := F_k$ . If  $u \equiv 0$  and the disturbance force is held constant at  $F \equiv \rho$  then the mass moves to an equilibrium position  $\bar{p}(\rho)$ . This equilibrium position can be determined by solving the cubic equation  $\rho = k_s(\bar{p}) \bar{p}$  for  $\bar{p}$ . Thus a parameterized collection of equilibrium conditions for the mass-spring-damper system is

$$(\bar{x}(\rho), \bar{u}(\rho)) := \left( \begin{bmatrix} \bar{p}(\rho) \\ 0 \end{bmatrix}, 0 \right). \quad (5.19)$$

The parameter varying linearization around this collection of equilibrium points is given by

$$\delta x_{k+1} = A(\rho_k) \delta x_k + B(\rho_k) \delta u_k + (\bar{x}(\rho_k) - \bar{x}(\rho_{k+1})), \quad (5.20)$$

where:  $A(\rho) := I_2 + \Delta t \begin{bmatrix} 0 & 1 \\ -\frac{k_{s,lin}(\rho)}{m} & -\frac{b}{m} \end{bmatrix}$  and  $B(\rho) := \Delta t \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}$ . The linearized spring constant at the operating condition  $\rho$  is  $k_{s,lin}(\rho) := k_1 + 3k_2 \bar{p}(\rho)^2$ . The spring stiffens as it is stretched and, as a result, the dynamics of the system change when the exogenous disturbance force  $F$  is applied.

Figure 5.2 shows responses of the nonlinear system and the corresponding parameter varying (LPV) linearization. The parameter varying linearization is in discrete-time with a sample time of  $\Delta t = 0.01$  s. The external force is  $\rho_k := F_k = 0.8 \cos(0.1 k_s \Delta t)$  N. The initial condition is  $x(0) = [\bar{p}(\rho_0), 0]^T = [0.751, 0]^T$  which corresponds to  $\delta x_0 = 0$ . The controllable

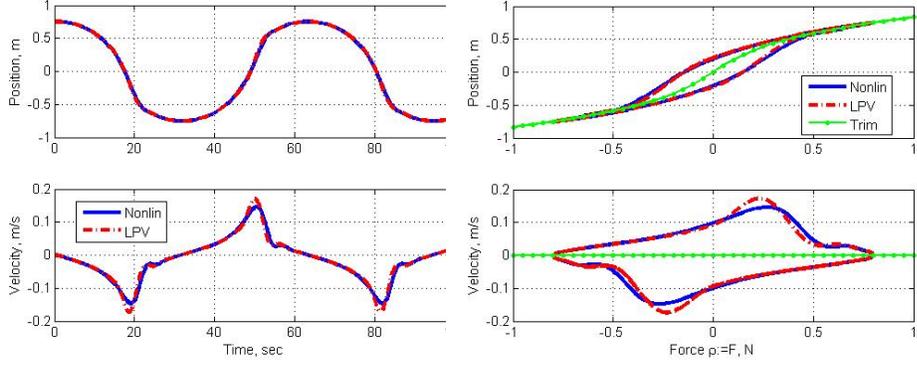


Figure 5.2: System response vs. time (left) and vs. parameter (right)

input is set to  $\delta u_k = 0$  for the entire simulation. Thus the only forcing arises due to the changing external disturbance force. This appears as a forcing term in the parameter varying linearization (5.20) due to the varying equilibrium term. The left subplot of Figure 5.2 shows the position and velocity vs. time. The parameter varying linearization agrees closely with the true nonlinear response. There are some small discrepancies between the two velocity trajectories. The parameter varying linearization is accurate even though the exogenous disturbance force, position, and velocity all vary over a large range. The right subplot shows the position and velocity vs. exogenous force. This subplot also shows the equilibrium (trim) relation  $\bar{x}(\rho)$  given in (5.19). The key point is that the position and velocity stay near this equilibrium manifold throughout the simulation. As a consequence,  $(\delta x_k, \delta u_k)$  remain small and the Taylor series approximation is accurate. If the frequency of  $\rho_k$  is increased then the trajectories will deviate more significantly from the equilibrium manifold. As a consequence, faster changes in the exogenous force lead to larger linearization errors.

### 5.5.2 Model Formulation: M-Blocks

This nonlinear mass-spring-damper is extended to consist of  $M$  blocks. Each block has mass  $m$  and is connected to its neighboring block by a linear damper and nonlinear spring. In addition, each block is connected to “ground” (fixed wall) by a linear damper and nonlinear spring. All dampers have constant  $b$  and all springs are given by  $k_s(p) = k_1 + k_2 p^2$ . Finally, the  $M^{\text{th}}$  block has an exogenous (disturbance) force  $F$  and a controllable input force  $u$ .

Let  $(p_i, v_i)$  denote the position and velocity of block  $i$ . The continuous-time dynamics for this  $M$ -block mass-spring-damper system are described by

$$\begin{aligned} m\dot{v}_i &= -F_i - F_{i,i-1} - F_{i,i+1} & (i = 1, \dots, M-1) \\ m\dot{v}_M &= F + u - F_M - F_{M,M-1}, \end{aligned} \tag{5.21}$$

where  $F_i$  denotes the force due to a spring and damper connecting block  $i$  to the fixed wall.  $F_{i,j}$  denotes the force due to a spring and damper connecting blocks  $i$  and  $j$ . These forces are defined as  $F_i := b v_i + k_s(p_i) p_i$  and  $F_{i,j} := b(v_i - v_j) + k_s(p_i - p_j)(p_i - p_j)$ . The convention for the first block is  $F_{1,0} \equiv 0$ . The position of the  $M^{\text{th}}$  block is the output for the system, i.e.  $y = p_M$ . All mass, spring, and damping constants are the same as those given in Figure 5.1. The result is a nonlinear system with state  $x := [p^T, v^T]^T \in \mathbb{R}^{2M}$ , a single controllable input  $u$  and single output  $y$ . The exogenous force  $F$  is again considered as a parameter in this model ( $\rho := F$ ). All results are given for  $M = 100$  blocks. This is small enough that the reduced- and full-order models can be directly compared to assess the feasibility of the method.

### 5.5.3 IOROM Method: Single Operating Point

The IOROM method described in Section 4.8 is used to construct a single time-invariant model with the exogenous force held constant at  $\rho \equiv 1$  N. The system is simulated with  $\rho \equiv 1$  and  $u = 0$  to determine the equilibrium point  $\bar{x}(\rho)$  for this parameter value. Next, the nonlinear system is excited for  $T_f = 20$  s with a chirp excitation input  $u(t) = 0.1 \sin(\omega(t)t)$ . The chirp frequency is  $\omega(t) = 0.1 (20)^{t/T_f}$  which excites the nonlinear system with frequencies from 0.1 to 2 rad/s. The nonlinear system is simulated with a time step of 0.01 s and snapshots of the state/output are collected every  $\Delta t = 0.1$  s. This yields 200 snapshots over the  $T_f = 20$  s simulation.

The left subplot of Figure 5.3 shows the POD and least-squares fitting errors obtained by the IOROM procedure. The horizontal axis is the number of POD modes selected from the snapshots  $X_0$ . The POD error is the relative energy lost in  $X_0$  by selecting the specified number of POD modes. The model relative fitting error is obtained by evaluating the IOROM least-squares cost with the optimal fit and dividing by  $\left\| \begin{bmatrix} X_1 \\ Y_0 \end{bmatrix} \right\|_F$ . Both relative errors decrease rapidly with increasing number of POD modes. The fitting error levels off around 4 modes hence this was selected for the model fit. The relative fit error does not go to zero because the discrete-time linear system cannot perfectly fit the snapshots from the nonlinear system. The right subplot shows a Bode plot for the four-state, reduced-order, discrete time model. The plot also shows a 200-state, full-order, continuous time linearization. The reduced-order model agrees quite closely with the full-order model in the frequency band of input excitation. The mismatch between the two models at high frequencies can be reduced by reducing the snapshot sampling time  $\Delta t$  and increasing the chirp excitation frequency.

A time-domain step response was used as another validation of the reduced-order model. Specifically, the left subplot of Figure 5.4 shows the output of the discrete-time, reduced-

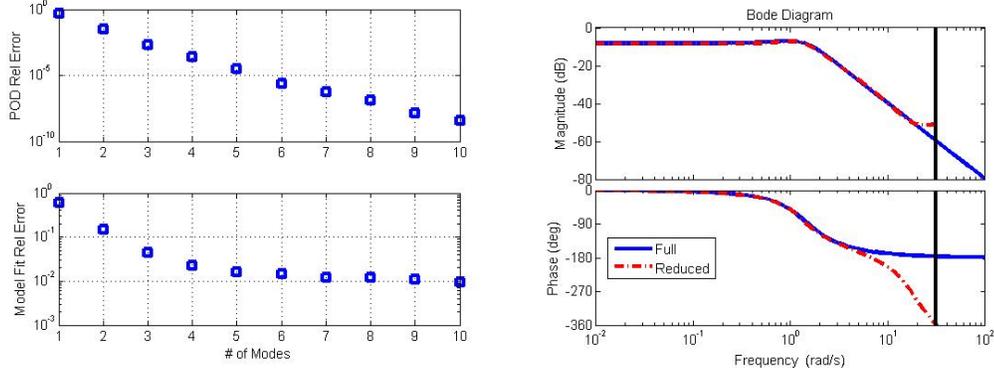


Figure 5.3: IOROM Results: POD/Fit Relative Errors (left) and Bode Plot (right). Note that the last block,  $x(100)$ , also corresponds to the output of the system.

order four-state model with a step input  $\delta u_k = 0.1$ . The exogenous force remains constant at  $F = \rho = 1$ . The vertical axis shows the position of the  $M^{th}$  block relative to the trim position. The plot also shows the response of the full-order, nonlinear system. The nonlinear system is initialized at the trim condition  $x(0) = \bar{x}(\rho)$ . The response of the reduced-order model is in close agreement with the nonlinear model. Moreover, the states of the reduced-order model  $z_k$  retain a physical significance. These can be “lifted” to estimate the response of the full-order system using the chosen POD modes ( $\delta x_k \approx Qz_k$ ). The right subplot of Figure 5.4 compares a subset of these state estimates (position of blocks 98-100) for the reduced-order system with the true response of the nonlinear system. Again there is close agreement although some discrepancies appear in the response for block 98. This error is due to the lack of input excitation. Specifically, the input force  $u$  is applied to block 100 and the chosen chirp amplitude provides less excitation for blocks located further from block 100.

Finally, it is important to note that the quality of the reduced-order model degrades if too many POD modes (i.e., much more than 4) are selected for the fit. This is due to overfitting a linear model to a nonlinear response. Inverse modeling techniques may be used to select the number of modes to accurately represent the system without overfitting [114]. The POD and model fit errors shown in Figure 5.3 provide useful criteria to avoid such overfitting.

#### 5.5.4 Parameter Varying IOROM Method

Next, the parameter varying IOROM method described in Section 5.4 is used to construct parameter varying reduced-order models. A grid of eleven parameter values was selected for the exogenous forces as  $\{0, 0.2, \dots, 2\}$ . At each fixed value of  $\rho = p^j$  in this grid ( $j = 1, \dots, 11$ ) the system was simulated with  $u = 0$  to determine the corresponding equilibrium

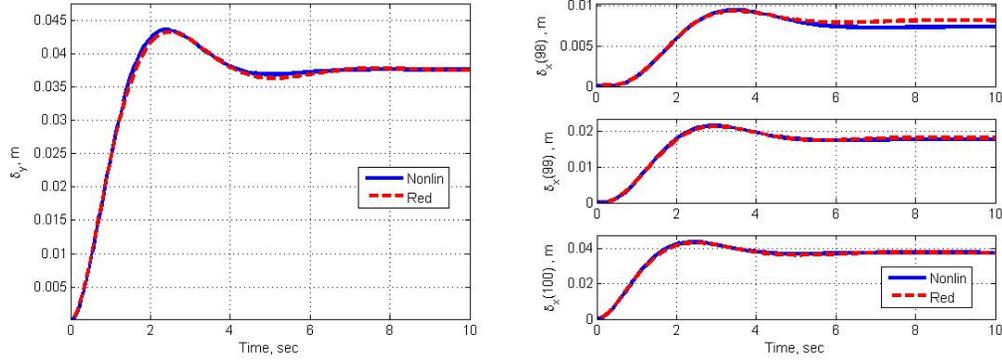


Figure 5.4: IOROM Results: Output vs. Time (left) and States vs. Time (right)

point  $\bar{x}(p^j)$ . Next, the nonlinear system is excited for  $T_f = 20$  s at each  $p^j$  with a chirp excitation input  $u(t) = 0.1 \sin(\omega(t)t)$ . The exogenous force  $F = \rho$  stretches the springs and this increases the system speed of response due to the spring stiffening. Thus the chirp frequency was chosen to depend on the parameter value as  $\omega(t) = 0.1(10 + 10\rho)^{t/T_f}$ . This excites the nonlinear system with frequencies from 0.1 to  $(1 + \rho)$  rad/sec. The nonlinear system is simulated with a time step of 0.01 s and snapshots of the state/output are collected every  $\Delta t = 0.1$  s. This yields 200 snapshots over the  $T_f = 20$  s simulation for each parameter grid point.

The basic single-step procedure (Algorithm 1) is used to construct the subspace modes  $Q$ . This example is sufficiently small such that it was possible to compute the SVD on snapshots obtained at all grid points. Five modes were selected and reduced-order models were constructed at each grid point using the IOROM procedure. The model relative fitting error was less than 0.05 at each grid point. This indicates that the reduced-order models at each “fixed” grid point accurately match the recorded snapshot data. The iterative method (Algorithm 2) was also implemented. This also yielded a five-state subspace  $Q$  although with slightly larger fitting errors. The single-step and iterative approach gave similar results and hence the remainder of the example focuses on the single-step algorithm.

Figure 5.5 compares a time-domain step response of the full-order nonlinear system and the reduced-order LPV model. The exogenous disturbance varies as  $F(t) = \rho(t) = -\cos(0.5t) + 1$ . The controllable input is a step  $u(t) = 0$  for  $t < 25$  s and  $u(t) = 0.1$  for  $t \geq 25$  s. The nonlinear system was simulated and the reduced-order parameter varying model was simulated in discrete time. Linear interpolation was used to compute the state matrices and equilibrium point appearing in the LPV model for parameter values not contained in the eleven point grid. The left subplot of Figure 5.5 shows the position of the  $M^{th}$  block. This is obtained from the parameter varying model as  $y_k = \bar{y}(\rho_k) + \delta y_k$ . The response of

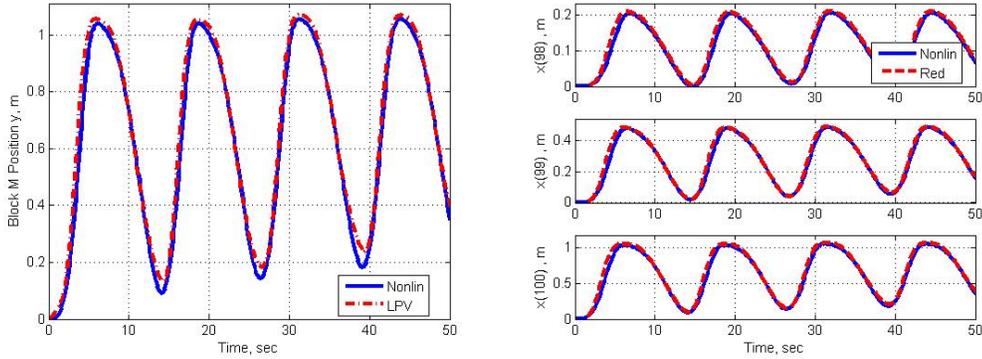


Figure 5.5: LPV IOROM Results: Output vs. Time (left) and States vs. Time (right). Again, note that the last block,  $x(100)$ , also corresponds to the output of the system.

the reduced-order model is in close agreement with the nonlinear model. The effect of the step input  $u$  is apparent after  $t = 25$  s. Again, the states of the reduced-order model  $z_k$  can be “lifted” to estimate the response of the full-order system using the chosen POD modes as  $x_k = \bar{x}(\rho_k) + \delta x_k \approx \bar{x}(\rho_k) + Qz_k$ . The right subplot of Figure 5.5 compares a subset of these state estimates (position of blocks 98-100) for the reduced-order system with the true response of the nonlinear system. Again, there is close agreement.

## 5.6 Conclusion

This chapter described a data-driven method to construct reduced-order models for high-dimensional nonlinear systems. It is assumed that the nonlinear system has a collection of equilibrium operating points. The method has two main components. First, a reduced-order linear system is constructed at each equilibrium point using input/output/state data. Second, a parameter varying linearization is used to connect these linear models. A nonlinear mass-spring-damper example was used to demonstrate this method. The following chapters will illustrate the use of this method on more complex models related to wind farm control.

## 5.7 Appendix

**Theorem 1.** Let  $Q_1 \in \mathbb{R}^{n_x \times r_1}$  be a given matrix with  $Q_1^T Q_1 = I_{r_1}$ . Let  $X \in \mathbb{R}^{n_x \times n_s}$  be given snapshot data. Define the SVD of the projected snapshot matrix  $(I_{n_x} - Q_1 Q_1^T)X = U \Sigma V^T$ . Let  $r_2$  be any non-negative integer such that  $r_2 \leq \text{rank}((I_{n_x} - Q_1 Q_1^T)X)$  and  $\sigma_{r_2} > \sigma_{r_2+1}$ . Then

$$\min_{\substack{Q_2 \in \mathbb{R}^{n_x \times r_2}, Q_2^T Q_2 = I_{r_2} \\ C_1 \in \mathbb{R}^{r_1 \times n_s} \\ C_2 \in \mathbb{R}^{r_2 \times n_s}}} \|X - Q_1 C_1 - Q_2 C_2\|_F^2 = \sum_{k=r_2+1}^{\text{rank}(X)} \sigma_k^2 \quad (5.22)$$

An optimal solution is given by  $C_{1,\text{opt}} = Q_1^T X$ ,  $Q_{2,\text{opt}} = U_r$ , and  $C_{2,\text{opt}} = \Sigma_r V_r^T$  where  $\Sigma_r$ ,  $U_r$ , and  $V_r$  are associated with the first  $r$  singular values and vectors of  $(I_{n_x} - Q_1 Q_1^T)X$ .

*Proof.* Use Gram-Schmidt orthogonalization to construct a matrix  $Q_{1,\perp} \in \mathbb{R}^{n_x \times (n_x - r_1)}$  such that  $\begin{bmatrix} Q_1 & Q_{1,\perp} \end{bmatrix}$  is orthogonal. The orthogonal invariance of the Frobenius norm thus implies

$$\|X - Q_1 C_1 - Q_2 C_2\|_F^2 = \left\| \begin{bmatrix} Q_1^T \\ Q_{1,\perp}^T \end{bmatrix} (X - Q_1 C_1 - Q_2 C_2) \right\|_F^2 = \left\| \begin{bmatrix} Q_1^T X - C_1 - Q_1^T Q_2 C_2 \\ Q_{1,\perp}^T X - Q_{1,\perp}^T Q_2 C_2 \end{bmatrix} \right\|_F^2 \quad (5.23)$$

The second equality follows from  $Q_1^T Q_1 = I_{r_1}$  and  $Q_{1,\perp}^T Q_1 = 0_{(n_x - r_1) \times r_1}$ . The error can be split as:

$$\|X - Q_1 C_1 - Q_2 C_2\|_F^2 = \|Q_1^T X - C_1 - Q_1^T Q_2 C_2\|_F^2 + \|Q_{1,\perp}^T X - Q_{1,\perp}^T Q_2 C_2\|_F^2 \quad (5.24)$$

The second term is unaffected by the choice of  $C_1$ . Moreover, for any  $(Q_2, C_2)$  the first term can be made equal to zero by the choice  $C_{1,\text{opt}} = Q_1^T X - Q_1^T Q_2 C_2$ . In fact,  $Q_1^T Q_2 = 0$  may be assumed without loss of generality. Specifically, the choice of  $Q_2$  only affects the second term of Equation (5.24) (assuming the optimal choice for  $C_1$  just specified). Perform a change of variables  $Q_2 = \begin{bmatrix} Q_1 & Q_{1,\perp} \end{bmatrix} \begin{bmatrix} \tilde{Q}_2 \\ \hat{Q}_2 \end{bmatrix}$ . This change of variables from  $Q_2$  to  $\begin{bmatrix} \tilde{Q}_2 \\ \hat{Q}_2 \end{bmatrix}$  is invertible since  $\begin{bmatrix} Q_1 & Q_{1,\perp} \end{bmatrix}$  is orthogonal. Substitute this change of variables into the second error term to obtain  $\|Q_{1,\perp}^T X - Q_{1,\perp}^T Q_2 C_2\|_F^2 = \|Q_{1,\perp}^T X - \hat{Q}_2 C_2\|_F^2$ . Thus  $\tilde{Q}_2$  has no effect on the second term and may be set to zero.  $Q_2$  can be selected to have the form  $Q_{1,\perp} \hat{Q}_2$  and hence  $Q_1^T Q_2 = 0$ . In this case, the optimal choice for  $C_1$  simplifies to  $C_{1,\text{opt}} = Q_1^T X$ .

Retaining the assumption that  $Q_1^T Q_2 = 0$  as well as  $C_{1,\text{opt}} = Q_1^T X$ , the total error is given by:

$$\|X - Q_1 C_1 - Q_2 C_2\|_F^2 = \|(I_{n_x} - Q_1 Q_1^T)X - Q_2 C_2\|_F^2 \quad (5.25)$$

By the standard POD result this cost is minimized by the choice  $Q_{2,\text{opt}} = U_r$ , and  $C_{2,\text{opt}} = \Sigma_r V_r^T$  where  $\Sigma_r$ ,  $U_r$ , and  $V_r$  are associated with the first  $r$  singular values and vectors of  $(I_{n_x} - Q_1 Q_1^T)X$ . It can be shown that this construction satisfies the assumption  $Q_1^T Q_{2,\text{opt}} = 0$ .  $\square$

## Chapter 6

# Wind Farm Application: Actuator Disk Model

### 6.1 Introduction

This chapter uses a medium-fidelity actuator disk model to highlight some of the advantages of the parameter varying IOROM approach developed in Chapter 5. This example provides a more complex example that illustrates the usefulness of the IOROM approach on high-dimensional fluid dynamic systems. In particular, this approach is shown to generate input-output reduced-order models that can be used for control design and analysis.

The actuator disk model presented in this chapter has been used to represent the flow within a wind farm. This particular implementation is a low Reynolds number actuator disk example that has been used in wind farm literature [46,47]. This example has approximately 20,000 grid points/states and exhibits highly nonlinear behavior.

### 6.2 Nonlinear Governing Equations

The actuator disk model is considered in this section [46,47]. The model was modified (from the description in Chapter 2) to incorporate viscosity. This model solves the two-dimensional unsteady, incompressible Navier-Stokes equations. The typical operating wind speeds in a wind farm do not exceed 25 m/s. This is low relative to the speed of sound at sea level (300 m/s) and hence it is sufficient to assume incompressibility [65]. Let  $(u, v)$  denote the streamwise and spanwise velocity components and  $(x, y)$  denote the downstream and spanwise distances. Under these assumptions, the dynamics for  $(u, v)$  are governed by

the following partial differential equations

$$\begin{aligned} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0 \\ \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} &= -\frac{\partial P}{\partial x} + \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + f_{x,i} \\ \frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} &= -\frac{\partial P}{\partial y} + \frac{1}{Re} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + f_{y,i}, \end{aligned} \quad (6.1)$$

where  $P$  is the pressure in the flow,  $Re$  is the Reynolds number, and  $f_{x,i}$  and  $f_{y,i}$  are the nondimensionalized forcing components of turbine  $i$  in the  $x$  and  $y$  direction.  $Re$  is defined as the ratio of inertial forces to viscous forces:  $U_\infty D / \nu$  where  $U_\infty$  is the freestream velocity [m/s],  $D$  is the diameter of the turbine [m], and  $\nu$  is the kinematic viscosity [m<sup>2</sup>/s].

The loading of each turbine is defined linearly. Assume that all spatial units have been nondimensionalized by the turbine diameter  $D$ . If the hub of the upstream turbine  $i$  is placed at  $x = x_i$  and  $y = y_i$  then the rotor plane lies within  $y_i - \frac{1}{2} \leq y \leq y_i + \frac{1}{2}$ . The forcing term introduced by the turbines is then given by

$$f_{x,i}(x, y, t) := \begin{cases} \rho k_{Tx,i} C_{T,i}(t)(y - y_i), & \text{if } x = x_i \text{ \& } |y - y_i| \leq 0.5; \\ 0, & \text{else.} \end{cases} \quad (6.2)$$

$$f_{y,i}(x, y, t) := \begin{cases} \rho k_{Ty,i} C_{T,i}(t)(1 - |x - x_i|), & \text{if } y = y_i \text{ \& } |x - x_i| \leq 0.1; \\ 0, & \text{else.} \end{cases} \quad (6.3)$$

where  $i$  indicates the number of turbines. Note that  $f_x$  is defined such that it will introduce an asymmetry to the flow that will induce wake meandering. The constant  $k_{Tx} := U_{in,i}$ , where  $U_{in}$  is the average nondimensionalized velocity across the rotor and  $k_{Ty} := V_{in,i}$  is the average nondimensionalized velocity across the rotor, and  $C_{T,i}$  is the thrust coefficient of the turbine  $i$ , which is a function of the turbine operation, i.e. the axial induction factor,  $a$ . According to actuator disk theory (described in Chapter 2), the thrust coefficient for each turbine is a function of the axial induction factor and is defined as  $C_{T,i}(t) = 4a_i(t)(1 - a_i(t))$  where  $a_i(t)$  is the time varying single input to the turbine  $i$ . The optimal operating point is  $a_i = \frac{1}{3}$ . The loading magnitude, as specified by the input  $a_i(t)$ , can be changed on a real turbine via blade pitch or changing the tip speed ratio via generator torque control [29].

These equations are solved using standard CFD methods [48]. Specifically, a central differencing scheme was used for the two-dimensional actuator disk model. The grid is defined by  $N_x$  points in the streamwise ( $x$ ) direction and  $N_y$  points in the spanwise ( $y$ ) direction. For this actuator disk model, the typical spacing between grid points is  $dx = 0.1$  and  $dy = 0.1$  with a typical time step of  $dt = 0.01$ . Figure 6.1 shows an example of a  $2 \times 3$  wind farm where mean streamwise and spanwise velocity are computed for the actuator disk model.

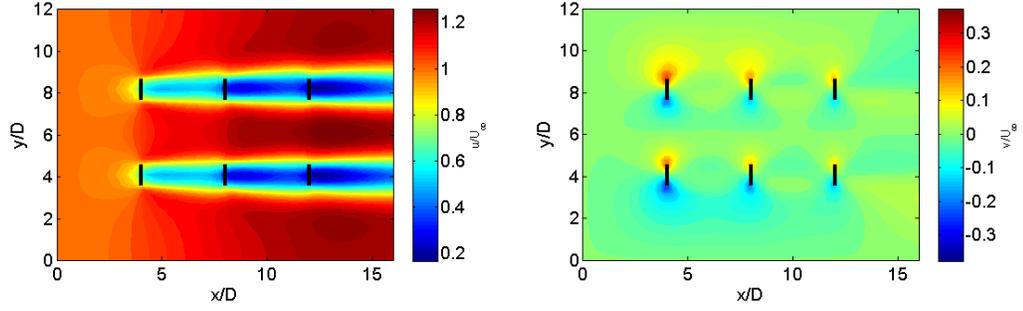


Figure 6.1: (Left) Mean streamwise velocity computed using the actuator disk model for a  $2 \times 3$  wind farm with  $4D$  spacing in the  $y$  direction and  $4D$  spacing in the  $x$  direction. (Right) Mean spanwise velocity computed using the actuator disk model for a  $2 \times 3$  wind farm with  $4D$  spacing in the  $y$  direction and  $4D$  spacing in the  $x$  direction.

In these figures, the turbines are separated by  $3D$  in the spanwise ( $y$ ) direction and  $4D$  in the streamwise ( $x$ ) direction. The velocities are normalized by  $U_\infty$ . The inflow boundary conditions of this model are

$$\begin{aligned} u(x=0, y, t) &= U_\infty \\ v(x=0, y, t) &= 0, \end{aligned} \quad (6.4)$$

and the outflow boundary conditions at  $x = L$  are

$$\begin{aligned} \frac{\partial u}{\partial t} + U_\infty \frac{\partial u}{\partial x} &= 0 \\ \frac{\partial v}{\partial t} + U_\infty \frac{\partial v}{\partial x} &= 0, \end{aligned} \quad (6.5)$$

where  $L$  is the total streamwise distance. In this example,  $L = 16D$ , the spanwise distance is  $12D$ ,  $N_x = 121$ ,  $N_y = 161$ , and the Reynolds number is  $50$ . Note that the code is nondimensionalized by  $U_\infty$ . This would amount to approximately  $40,000$  states for this particular example by having 2 states,  $(u, v)$  per grid point. More realistic, higher fidelity codes will have even larger state dimensions. Note that the Reynolds number is significantly lower than the typical Reynolds number, which is approximately  $10^6$ . However, for the purposes of this chapter, the flow is restricted to a low Reynolds number to demonstrate the feasibility of the reduced-order modeling approach proposed in Chapters 4 and 5. Chapter 7 will include simulations of wind farms with the appropriate Reynolds number.

Lastly, note the turbine are modeled as actuator disks. The wakes directly behind real turbines are dominated by tip vortices that are generated based on the blade geometry. The

blades are not modeled in this simulation and as a result, this model cannot accurately depict this near wake region. However, this model captures the effects of the flow far downstream, greater than 3D, where the flow is less dependent on turbine geometry. Therefore, this model can be useful for studying the far wake of a turbine in steady and unsteady flows.

### 6.3 Linearized Equations

Typically, the first step in producing a model suitable for controls is to linearize the equations of the actuator disk model. For the purposes of this chapter, the actuator disk equations will be linearized around a base flow of  $\mathbf{U} = (U(x, y), V(x, y))$  where  $U(x, y)$  and  $V(x, y)$  define the baseflow that corresponds to all turbines operating at their peak efficiency.

The linearized governing equations about the baseflow, after some algebraic manipulation, can be rewritten as

$$\frac{d}{dt} \begin{bmatrix} \mathbf{u}' \\ \mathbf{v}' \end{bmatrix} = A \begin{bmatrix} \mathbf{u}' \\ \mathbf{v}' \end{bmatrix} + Ba(t), \quad (6.6)$$

where  $\mathbf{u}' \in \mathbb{R}^{(N_x N_y) \times 1}$  denotes the fluctuations from the baseflow in the streamwise direction,  $\mathbf{v}' \in \mathbb{R}^{(N_x N_y) \times 1}$  denotes the fluctuations from the baseflow in the spanwise direction,  $A \in \mathbb{R}^{(2N_x N_y) \times (2N_x N_y)}$  contains the spatial discretization information of the flow field,  $B \in \mathbb{R}^{(2N_x N_y) \times N_{\text{turb}}}$  contains the location of the turbines, and  $a(t) \in \mathbb{R}^{N_{\text{turb}} \times 1}$  is the input to the turbines with  $N_{\text{turb}}$  denoting the number of inputs, which in this case is the number of turbines.

#### 6.3.1 Linear Outputs

One possible output measurement of a wind farm model would be the wind speeds within the wind farm. By knowing the wind speed in the wind farm, a turbine could be operated in a more optimal way. For example, control strategies are being developed with the use of LiDAR (Light Detection and Ranging), which measures the wind speed in front of the turbine to improve the performance of an individual turbine [115, 116]. This would amount to measuring the states, or velocity fluctuations, at specific locations within the domain, i.e. setting the output to

$$y(t) = Cx(t), \quad (6.7)$$

where  $C$  contains the location of the measurement points within the wind farm,  $x$  is the velocities,  $y$  is the output wind speed.

### 6.3.2 Nonlinear Outputs

Another wind farm output is the vector of all power produced by all turbines denoted  $P \in \mathbb{R}^{N_{\text{turb}}}$ . Recall that the power,  $P_i$ , produced by turbine  $i$  is  $P_i = \frac{1}{2}\rho AU_i^3 C_{P,i}$  given in (2.1). Linearizing (2.1) for each turbine, about the optimal power output, yields the following measurement equation for power fluctuations

$$P' = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} u' \\ v' \end{bmatrix} + Da(t), \quad (6.8)$$

where  $C \in \mathbb{R}^{N_{\text{turb}} \times (2N_x N_y)}$  contains the location of the measurements with  $N_{\text{turb}}$  number of outputs, and  $D \in \mathbb{R}^{N_{\text{turb}} \times N_{\text{turb}}}$  contains information about turbine efficiency.

With either output representation, the linearized system can now be given by a dynamic system of the form

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (6.9)$$

$$y(t) = Cx(t) + Du(t), \quad (6.10)$$

where  $x := \begin{bmatrix} u' \\ v' \end{bmatrix}$ ,  $u$  is the input axial induction factor, and  $y$  are the specified outputs. This linearized model contains  $2N_x N_y$  states and is not suitable for control design and analysis. Despite being able to linearize these equations, the state matrices are too large for classical model reduction techniques such as balanced truncation [15]. The next section uses the proposed IOROM technique to generate a low-order representation of the system that is suitable for control.

## 6.4 Example: Two Turbine Array

### 6.4.1 Setup

To demonstrate the feasibility of the reduced-order modeling approaches presented in Chapter 4 and 5, the actuator disk model is used for a two turbine example. Specifically, the two turbines are spaced 5D apart in the streamwise direction. There is a point 3D downstream of the second turbine where the wind speed is measured (indicated by the white triangle in 6.2). Each turbine has an input axial induction factor,  $a_i$ . In this particular example,  $a_2$  is the single input to this system, i.e.  $a_1$  is fixed at the optimal operating point and  $a_2$  is allowed to vary about the optimal operating point. It is thought that  $a_2$  has the highest impact on the measurement point 3D downstream. The loading magnitude can be changed by the thrust coefficient. Recall  $C_T = 4a(1 - a)$ , i.e. the loading the turbine can be varied by changing the induction factor. On utility-scale turbines, this would be equivalent to

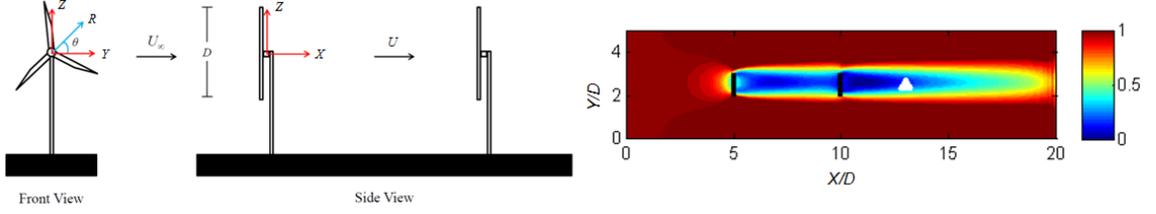


Figure 6.2: Two-turbine setup (left) and the corresponding baseflow (right) for the low  $Re$  flow specified. It should be noted that the flow is depicted as if you are looking at the two-turbine array from above.

changing the blade pitch angle or generator torque [29]. For this actuator disk model, the spacing between grid points is  $\delta x = 0.1$  and  $\delta y = 0.1$  with a typical time step of  $\delta t = 0.01$ .

Again, for simplicity, it is assumed that only the second turbine can be used to control the wind speed at this point. As a result, the system that will be identified will be a single-input-single-output system where the input is the axial induction factor of the downstream turbine,  $a_2$ , and the output is the spanwise velocity fluctuations,  $v'$ , at the white triangle in Figure 6.2. In addition, the domain in the streamwise direction was set to  $20D$  with 201 grid points and in the spanwise direction was set to  $5D$  with 51 grid points. This would amount to 20,000 states for this particular example by having 2 velocity components per grid point. More realistic, higher fidelity codes for wind farms will have even larger dimensions [11,12].

#### 6.4.2 IOROM Method: Single Operating Point

First, the IOROM method discussed in Section 4.8 will be implemented for the actuator disk problem at one particular operating point, i.e., a single  $Re$ . The actuator disk example is defined in terms of  $Re$ .  $Re$  is used as a proxy for wind speed. Algorithm 1 is used for this example, but as examples become larger than this, i.e. large eddy simulations where the states are on the order of millions rather than tens of thousands, algorithm 2 will have to be implemented. One way to do is to use the technique used for large streaming data sets in [105].

Similar to the mass-spring-damper example in Chapter 5, the IOROM method is used to construct a single time-invariant model with  $Re$  held constant. The system is simulated with  $Re = 50$  and  $u = a_2 = 1/3$ , i.e. the optimal operating point, to determine the equilibrium point  $\bar{x}(Re)$  for this parameter value. Next, the nonlinear system (6.1) is excited in discrete time for  $T_f = 100$  s (recall the time step for this simulation is 0.01 s). This system is excited at the second turbine with a chirp excitation input  $u(t) = 0.25 + 0.25 \sin(\omega(t)t)$ , where  $\omega(t) = 0.0628(50)^{\frac{t}{T_f}}$ . This will vary the thrust coefficient,  $C_T$  from 0 to 1 and excites the

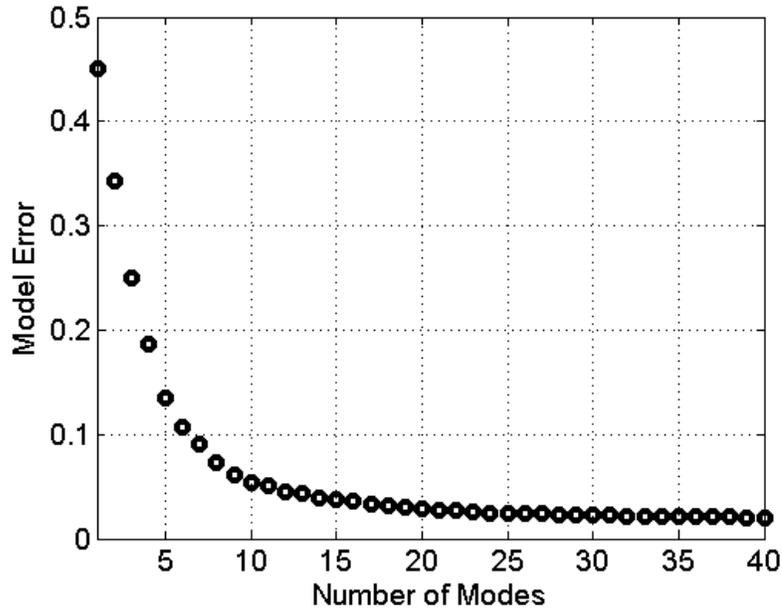


Figure 6.3: Model fitting error computed for the actuator disk example

nonlinear system with frequencies from 0.01 to 0.25 Hz. The nonlinear system is simulated and snapshots of the state/input/output are collected every 0.5 s. This yields 200 snapshots over the 100 s simulation.

Using the IOROM approach, a reduced-order model with 20 states could be constructed. This was selected by examining the model fitting error. Because the state dimension is large, the model fitting error cannot be computed directly. Instead, a way to compute model error was addressed in Chapter 4, where (4.43) was used to compute the model fitting error. The results are shown in Figure 6.3. In particular, the error in the model decreases and starts to level off at approximately 20 modes. At this point, there are no additional benefits to increasing the order of the model. As stated previously, it is possible that the performance of the model will degrade as the model order increases because the model will be overfitting to the nonlinearities of the system.

The reduced-order model was validated using a square wave as an input to the second turbine. The results can be seen in Figure 6.4. One of the main advantages to this method is that the full state can be reconstructed from the reduced-order state. Figure 6.4 shows the results of the reduced-order model as compared to the full-order model. The top plot shows the reconstructed flow field from the 20 state model identified from the IOROM approach. The middle plot shows the full-order actuator disk model. Visually, the top plot and the middle plot show good agreement. The bottom plot shows the output; i.e., the velocity measured 3D downstream of the second turbine. The output of the reduced-order

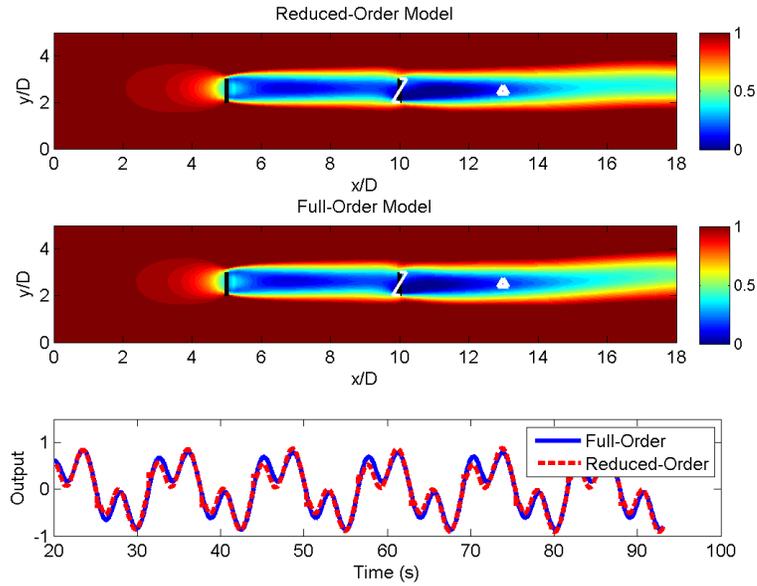


Figure 6.4: IOROM Results of the actuator disk example. The top plot shows the flow field reconstructed using 20 modes. The middle plot shows the full-order nonlinear flow. The bottom plot shows the output (indicated by the triangle in the top and middle plot) of the reduced-order model compared to the full-order model.

model closely matches the output of the full-order model.

The IOROM technique that was used to identify a low-order model for the actuator disk example was compared to another existing technique, ERA (discussed in Section 4.4), that is capable of constructing an input/output model. A separate simulation was run to obtain the impulse response data necessary for ERA using an impulse as an input. Figure 6.5 shows the results of the input/output relationship obtained using ERA and the IOROM approach on the actuator disk example. In Figure 6.5, the input to the system is shown on the top, where  $u(t) = 0.25 + 0.25 \sin(0.5t)$ . The output of the reduced-order models are shown on the bottom plot. The bottom plot shows that ERA has a difficult time identifying the system at this particular operating point. This is in part due to the excitation signal. With the IOROM technique, a chirp signal was used, which excites the dynamics at a range of frequencies. With ERA, an impulse is used and is unable to capture the essential dynamics of the system. This IOROM technique has a much larger excitation energy than the impulse used by ERA. This allows the IOROM to capture a better input/output model. However, to use this IOROM approach, the system must be of the form where it is easily excitable by a controllable input. It should be noted that ERA has been extended to include non-impulse inputs using observer Kalman identification (OKID) [80]. This ERA-OKID approach was applied to the actuator disk problem and no performance benefit was

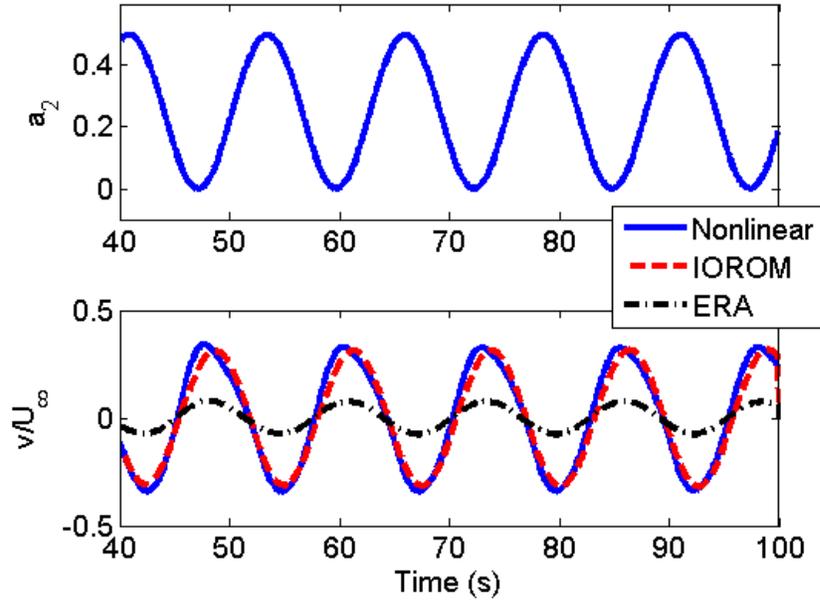


Figure 6.5: This plot shows the output of the reduced-order models constructed using ERA and the IOROM approach proposed in this thesis. The output for this particular example is the velocity fluctuations at the measurement point 3D downstream of the second turbine.

observed in comparison with ERA. This is likely due to the strong nonlinearities present in this particular example. This indicates that the additional state information used with the input/output data for the IOROM approach can help identify a low-dimensional model for the actuator disk example. Lastly, it should be noted that ERA has been extended to a time-varying framework in [81–83]. If the states cannot be directly recorded, using this time-varying ERA approach may be a reasonable alternative.

#### 6.4.2.1 Model-Based Controller Design Example

Once a sufficient reduced-order model that captures the appropriate input/output behavior is obtained, a controller can be designed to achieve a particular objective. In this example, the objective of the controller is to minimize wake meandering, and thereby minimize velocity fluctuations at a specified downstream location. To do this, a linear quadratic gaussian (LQG) controller was designed [117]. Other types controllers could be implemented to track a specific power command or to minimize specific structural loads on the turbine. For example, a specific power may be tracked to provide reliability to the grid [67,68]. Additional controller designs, such as linear quadratic regulators (LQR) and  $\mathcal{H}_\infty$  [117], can be used with the models obtained through this IOROM approach. In addition to model-based controllers, PI/PID controllers can be designed with these IOROMs with the idea that these

IOROMs run in seconds and make it easier to do iterative control design rather than using the nonlinear model, which for this actuator disk model, can take minutes to hours for one simulation depending on the number of turbines, size of the domain, etc. This section is intended to demonstrate one example of controller design that can be used for the wind farm control problem.

As mentioned previously, there is one input, which is the axial induction factor,  $a_2$ , that generates the thrust force at the last turbine (through the axial induction factor), and there is one output which is the spanwise velocity measurement. Again, the objective of the LQG controller is to minimize the output fluctuations. In addition, the LQG controller is able to handle disturbances and make corrections at the input to account for these disturbances.

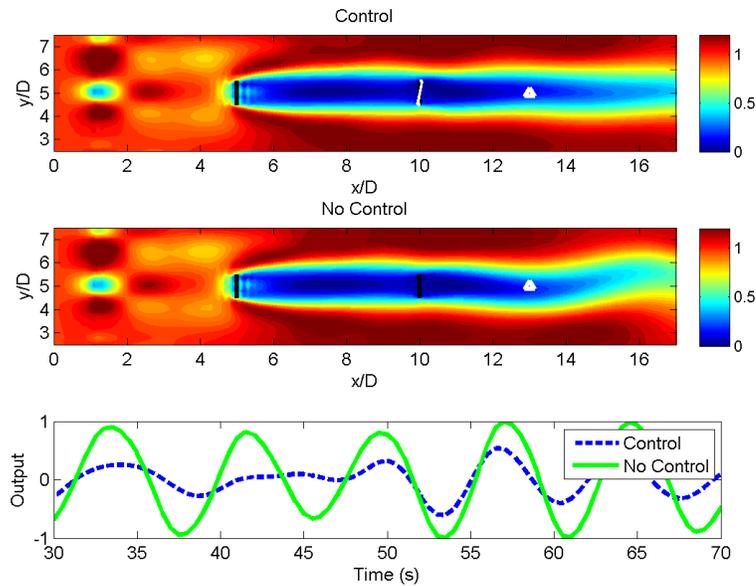


Figure 6.6: Time domain output of the actuator disk flow with and without the LQG controller designed using the IOROM.

This thesis will not go into detail about the implementation of an LQG controller. The details of implementing an LQG controller can be found in [117]. At a high level, an LQG controller tries to minimize a quadratic cost function that penalizes the transients in the state and the control effort. The contribution of the states and the control inputs to the cost function are defined by separate weights,  $Q_c$  and  $R_c$ . For this example, the weights for the state are set to  $Q_c = 100I_r$  and  $R_c = 0$ . This indicates that there is a penalty on the state transients and no penalty on the control effort. Note that when designing a wind farm controller, it may be more desirable to limit the control effort, which means that  $R_c$  would be set to a nonzero value. In addition to the cost function, an LQG controller also

includes a Kalman filter that is used to handle disturbances. To implement the Kalman filter, information about the process and measurement noise should be known. However, for the wind farm control problem, this information is not always known. As a result, the information about noise, i.e., the covariances, are estimated. In this case, the covariance of the process noise is set to  $Q_k = 5I_r$  and the covariance of the measurement noise is set to  $R_k = 0.00001I_{n_y}$ . These weighting matrices are used in the standard LQG implementation.

The results of implementing this controller can be seen in Figure 6.6. A random disturbance has been added to the inflow. The top plot shows the full-order system with the controller implemented. The middle plot shows a snapshot of the nonlinear simulation with no controller implemented. Lastly, the bottom plot shows the time domain data of the measurement point downstream (indicated by the white triangle). This bottom plot indicates that by controlling the thrust of the downstream turbine, the wake meandering or velocity fluctuations at the downstream turbine can be reduced significantly. This figure shows that the controller designed with the IOROM model is able to achieve the control objective of decreasing the spanwise velocity fluctuations at a point downstream of the turbines. Other controllers can be designed to track specific power reference signals for grid reliability, minimize structural loading on the turbines, maximize power within a wind farm, etc.

### 6.4.3 IOROM Parameter Varying Method

The parameter varying method described in Section 5.4 is used to construct parameter varying reduced-order models. A grid of five parameter values was selected,  $Re_j = \{10 : 10 : 50\}$ . At each fixed value of  $Re$  in this grid, the system was simulated with  $u = 0$  to determine the corresponding equilibrium point,  $\bar{x}(Re_j)$ . Next, the nonlinear system is excited for  $T_f = 100$  s at each  $Re_j$  with a chirp excitation input  $u(t) = 0.25 + 0.25 \sin(\omega(t)t)$ , where  $\omega(t) = 0.0628(50)^{\frac{t}{T_f}}$ . The  $Re$  changes the effective wind speed of the simulation which changes the dynamics within the system. For example, wake meandering is an oscillation that happens in this particular flow due to the instabilities in the wake. Wake meandering instabilities increase as the  $Re$  increases in this particular example. The chirp signal frequencies were chosen to capture the wake meandering frequency. Specifically, the chirp signal excites the nonlinear system at frequencies between 0.01 Hz to 0.5 Hz. The nonlinear discrete-time system is simulated and snapshots of the state/input/output are collected every 0.5 s. This yields 200 snapshots over the  $T_f = 100$  s simulation for each parameter grid point.

The basic single step procedure (Algorithm 1) is used to construct the subspace modes  $Q$ . This example is small enough that it was possible to compute the SVD on snapshots obtained at all grid points. In addition, a MapReduce technique introduced in [118] can be

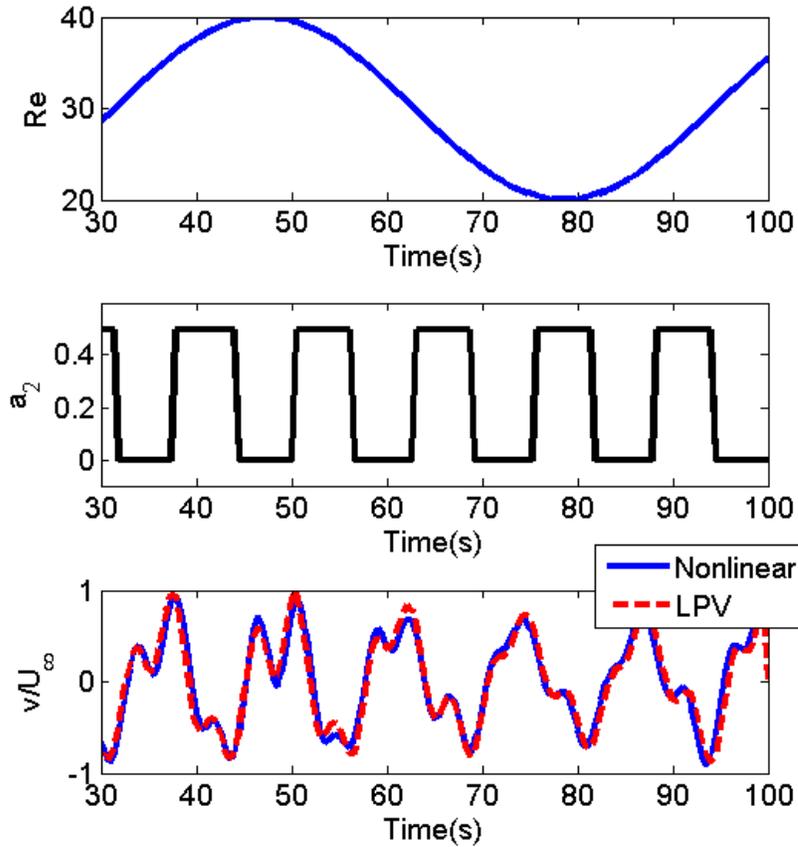


Figure 6.7: Time domain signal of the Reynolds number, input, and output. The output of the reduced-order LPV model is compared to the full-order nonlinear simulation.

used to compute the SVD of large, tall-skinny matrices. Twenty modes were selected and reduced-order models were constructed at each grid point using the IOROM procedure.

Figure 6.7 compares a time-domain step response of the full-order nonlinear system and the reduced-order LPV model. The  $Re$  varies as  $Re(t) = 25 - 10 \sin(0.1t)$ . The controllable input was set to  $u(t) = 0.25 + 0.25 \text{square}(0.5t)$ . The nonlinear system was simulated in discrete time with a time step of 0.01s and the reduced-order parameter varying model was simulated in discrete time with a time step of 0.5s. Linear interpolation was used to compute the state matrices and equilibrium points appearing in the LPV model for parameter values not contained in the five point grid. The input/output relationship of the reduced-order LPV model is in close agreement with the full-order nonlinear system. In addition to the input-output behavior, Figure 6.7 shows the flow field of the nonlinear system (right) in comparison to the reconstructed flow field from the reduced-order LPV

model (left) at different instances in time. Visually, the reduced-order model shows good agreement. These results indicate that this reduced-order LPV model can be used for predictive purposes and/or control design.

## 6.5 Conclusions

The parameter varying IOROM approach proposed in this thesis have been applied to high-dimensional systems. These techniques are useful in identifying a model that can be used for control design and analysis. Specifically, the example presented in this chapter indicates that these techniques can be useful for designing controllers for wind farms that can manipulate the the flow dynamics.

The flow within a wind farm is highly nonlinear and parameter varying, i.e. wind speed and wind direction changes frequently. This chapter only addressed changing wind speeds. However, changing wind directions can be treated in a similar way. The parameter varying IOROM approach is able to construct an LPV model that represents the changing dynamics in a wind farm represented by the actuator disk model. The IOROM technique has also been successfully applied to a more realistic wind farm model using a large eddy simulation with 3.6 million states. In addition, this approach has been used on data from wind tunnel experiments. Both applications, at a single operating point, are discussed in more detail in the next chapter.

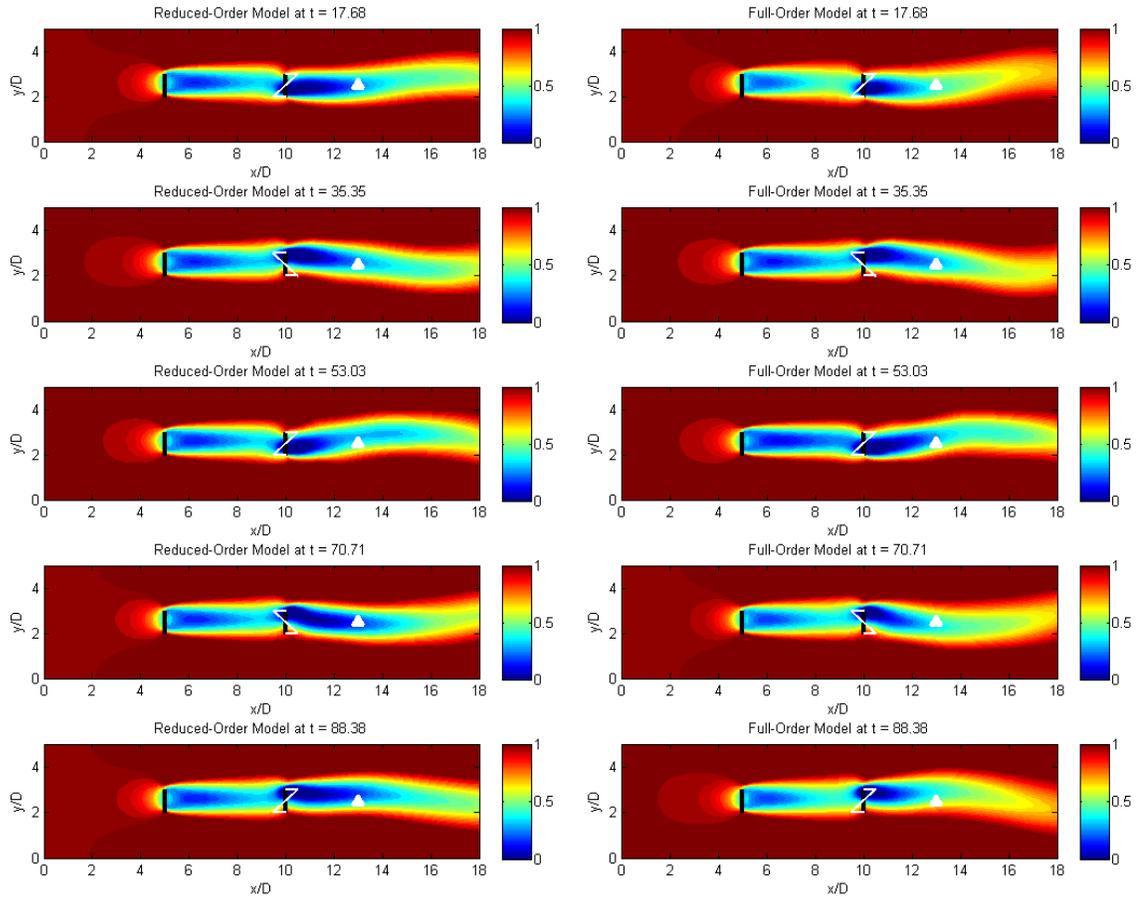


Figure 6.8: Flow Visualization at 5 different snapshots in the flow for the LPV formulation. These snapshots correspond to the same example in Figure 6.7. Note: the white lines at the turbine locations indicate the different thrust values on the turbines. As noted previously, the formulation of the actuator disk example introduces an asymmetric forcing on the turbine to induce wake meandering.

## Chapter 7

# Extension to Large Eddy Simulations and Wind Tunnel Experiments

### 7.1 Introduction

This chapter uses the IOROM approach, described in Chapter 4, to construct reduced-order models of wind farm simulations and experiments. In contrast to Chapter 6, this chapter addresses higher-fidelity simulations that are computationally expensive to run. Therefore, this chapter only addresses single operating points consisting of one wind speed and wind direction. A similar approach to the one discussed in Chapter 6 can be applied to these simulations to get a collection of reduced-order models at various operating points. Section 7.2 describes how to handle process noise, model uncertainty, and measurement noise when using a Kalman filter with a low-dimensional model constructed from simulations or experiments. This chapter also demonstrates that data collected during experiments can be used in the same way as simulations to obtain reduced-order models that capture the dominant characteristics and the input/output dynamics.

### 7.2 Input-Output Reduced-Order Model with a Kalman Filter

To apply this IOROM approach to higher fidelity simulations or experiments, this section introduces a way to handle the error between the reduced-order model and the nonlinear simulation/experiment. Note that the IOROM identified, using the technique described in Section 4.8, is the best fit model for the available data. For example, one issue that arises in high-dimensional systems is the influence of turbulence. This is a disturbance in the system

that cannot exactly be quantified in a linear, time invariant model. The IOROM method is useful in identifying the dominant dynamics and maintaining the approximate input-output behavior. To improve the performance of the model, it is recommended that a state estimator be used along with the IOROM to handle the influence of disturbances, such as turbulence. For this study, a Kalman filter will be used with the identified IOROM [119,120]. This is standard practice in the controls/system literature. A similar approach was proposed in [99] where a model was constructed using DMD and embedded in a Kalman filter. The Kalman filter is the best linear filter for a system even if the noise is not Gaussian [120]. Any improvements over the Kalman filter would require a nonlinear filter. The Kalman filter is able to provide a computationally efficient algorithm for estimating the state based on measured outputs. It should also be reiterated that the purpose of this Kalman filter is to aid the IOROM in predicting the dominant characteristics rather than the turbulent characteristics of the flow. The IOROM embedded in the Kalman filter is briefly described in this section.

Consider the reduced-order model of the system

$$\begin{aligned} z_{k+1} &= Fz_k + Gu_k + w_k \\ y_k &= Hz_k + Du_k + v_k, \end{aligned} \tag{7.1}$$

where  $w_k \in \mathbb{R}^r$  is the process noise,  $v_k \in \mathbb{R}^{n_y}$  is the measurement noise,  $z_k \in \mathbb{R}^r$ ,  $u_k \in \mathbb{R}^{n_u}$ , and  $y \in \mathbb{R}^{n_y}$  are the reduced-order states, inputs, and outputs respectively. The reduced-order state matrices  $(F, G, H, D)$  have the dimensions of  $(z, u, y)$ . For the wind farm example,  $z_k$  are the POD modes of the system,  $u_k$  are the inputs such as blade pitch angle and generator torque, and  $y_k$  are the measurements such as the power at each turbine, wind speed, or structural loads.

The standard Kalman filter can be implemented with the IOROM using the approach outlined in [119]. The Kalman filter uses measurements to update the estimate of the state at a particular time step. To implement the Kalman filter, the properties of the process noise  $w_k$  and measurement noise  $v_k$  should be known or estimated. Typically, the covariance matrices of the process noise,  $Q_k$ , and measurement noise,  $R_k$ , can be determined by  $Q_k = E(w_k w_k^T)$  and  $R_k = E(v_k v_k^T)$ . This indicates that the process noise and measurement noise are independent. However, in the wind farm application the information about the noise is not known. Hence  $Q_k$  and  $R_k$  are instead tuned to estimate the noise. Especially for wind farm simulations,  $R_k$  is expected to be small since exact measurements can be taken from the simulations.  $Q_k$  is tuned such that all of the modes of the reduced-order model are weighted equally. Another approach to defining  $Q_k$  could be to weight the modes differently,

e.g. assign higher weights to the less dominant modes. The Kalman filter is initialized by

$$\begin{aligned}\hat{x}_0^+ &= E(x_0) \\ P_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T]\end{aligned}$$

where  $x_0^+$  is the initial state estimate and  $P_0^+$  is the covariance on the state error. For these wind farm examples addressed in this chapter, the initial covariance,  $P_0^+$ , is set to be  $I_r$ . The remainder of the Kalman filter is evaluated as

$$\begin{aligned}\text{Covariance:} \quad & P_k^- = FP_{k-1}^+F^T + Q_{k-1} \\ \text{Kalman Gain:} \quad & K_k = P_k^-H^T(HP_k^-H^T + R_k)^{-1} \\ \text{State Estimate:} \quad & \hat{x}_k^- = F\hat{x}_{k-1}^+ + Gu_{k-1} \\ \text{Measurement Update:} \quad & \hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H\hat{x}_k^- - Du_{k-1}) \\ \text{Covariance Update:} \quad & P_k^+ = (I - K_kH)P_k^-\end{aligned}$$

where “−” indicates the settings of the filter before the measurement update and “+” indicates after the measurement update. This approach shows good performance when used with the identified IOROM. Note that the performance of the Kalman filter is dependent on the model identified by the IOROM approach. Specifically, if the selected model has too many modes, the IOROM is essentially overfitting the data. Although this may provide a good performance with the data the model originated from, the model shows poor performance even with a Kalman filter implemented. In this case, the Kalman filter has to compensate significantly to fit a different data set of the same properties. An example of this is shown in the results section.

The next sections will provide some useful examples for applying this IOROM approach to high-dimensional systems. The goal of this study is to construct reduced-order models that can be used for control design. Specifically, it will be shown that the reduced-order model constructed using this IOROM approach maintains the dominant dynamics of the system as well as maintain the appropriate input-output behavior. In particular, the next section will demonstrate the feasibility of this IOROM approach on large eddy simulations as well as wind tunnel experiments using particle image velocimetry (PIV).

### 7.3 Large Eddy Simulations

The IOROM method was used to obtain a reduced-order model from the Simulator fOR Wind Farm Applications (SOWFA) simulations [11]. SOWFA is a high-fidelity large eddy simulation tool that was developed at the National Renewable Energy Laboratory (NREL) for wind farm studies. SOWFA is a CFD solver based on OpenFOAM and is coupled with

NREL’s FAST modeling tool [30]. SOWFA has been used in previous wind farm control studies [40, 50]

SOWFA uses an actuator line model coupled with FAST to study turbines in the atmospheric boundary layer. Specifically, SOWFA solves the three-dimensional incompressible Navier-Stokes equations and transport of potential temperature equations, which take into account the thermal buoyancy and Earth rotation (Coriolis) effects in the atmosphere. The inflow conditions for this simulation are generated using a periodic atmospheric boundary layer precursor with no turbines. Additional details can be found in [40].

SOWFA calculates the unsteady flow field to compute the time-varying power, velocity deficits, and loads at each turbine in a wind farm. This level of computation, with high-fidelity accuracy, takes a number of days to run on a supercomputer using a few hundred to a few thousand processors, depending on the size of the wind farm. The simulations run for this study were performed on NREL’s high-performance computer Peregrine.

### 7.3.1 Two-Turbine Setup

A high-fidelity simulation of a two-turbine scenario was performed with SOWFA to provide the data for the IOROM. The two-turbine setup is shown in Figure 7.1 (left). The turbines were aligned with the dominant wind direction with a spacing of 5 diameters (5D). The simulated turbines are NREL 5 MW baseline turbines [54], which have a rotor diameter of  $D = 126$  m. The conditions simulated in SOWFA are based on the study reported in [40, 50]. They consist of a neutral atmospheric boundary layer with a low aerodynamic surface roughness value of 0.001 m, which is typical for offshore conditions. The generated inflow, coming from the southwest ( $300^\circ$ ), has a horizontally averaged wind speed of 8 m/s and a turbulence intensity of 10 % at the turbine hub height. A simulation time of 1800 s is used to let the wakes develop through the domain. The spatial discretization for the CFD solver is refined in two steps with the smallest cells containing the turbine rotors, the axial induction zones of the rotor, and a large part of the wake. Farther from the turbines, the mesh is coarser to reduce computation time.

### 7.3.2 IOROM with SOWFA

The IOROM method can be used to construct a reduced-order model of the two-turbine setup. For this particular example, there are four inputs: the blade pitch angle ( $\beta_1$ ) and the generator torque ( $\tau_{g1}$ ) of the upstream turbine and the blade pitch angle ( $\beta_2$ ) and the generator torque ( $\tau_{g2}$ ) of the downstream turbine. In this example, the generator torque is modulated by changing a scaling constant associated with the standard generator torque control law for Region 2 operation of a utility-scale turbine [29]. Specifically, the scaling

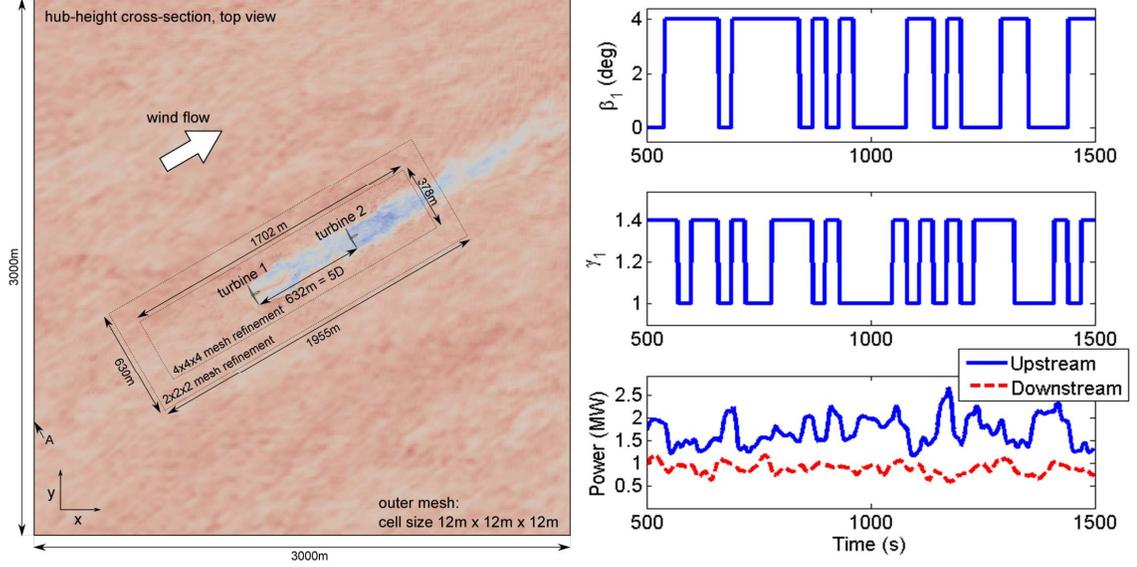


Figure 7.1: (Left) Setup for the two-turbine array in SOWFA. (Right) Forced input used for the two-turbine array.

factor,  $\gamma$ , is applied to the generator torque control law of the upstream turbine, so that the applied generator torque is  $\tau_g = \gamma K_g \omega^2$  with  $K_g = 0.0179 \text{ Nm/RPM}^2$  and  $\omega$  is the rotor speed [RPM], resulting in a deviation from the turbine-level optimal gain  $K_g$  for maximum power production. The outputs of interest are the power generated by the upstream ( $P_1$ ) and downstream turbine ( $P_2$ ). Overall, the goal is to generate a reduced-order model of this form:

$$\begin{aligned} z_{k+1} &= Fz_k + Gu_k \\ y_k &= Hz_k + Du_k \end{aligned} \quad (7.2)$$

where  $u := [\beta_1, \gamma_1, \beta_2, \gamma_2]$  and  $y := [P_1, P_2]$ .

To generate an IOROM of this example, forced inputs were applied to the upstream turbine by changing the collective blade pitch angle from  $0^\circ$  to  $4^\circ$  using a pseudo-random binary sequence (PRBS), see the top plot in Figure 7.1 (right). In addition, the generator torque constant,  $\gamma_1$ , was modulated between 1 and 1.4 using a different PRBS, see the middle plot in Figure 7.1 (right). The blade pitch angle and generator torque constant of the downstream turbine were held at constant at their optimal values, i.e.  $\beta_2 = 0$  and  $\gamma_2 = 1$ , see the bottom plot in Figure 7.1 (right). This indicates how the forcing inputs at the upstream turbine affect the power output of both the upstream and downstream turbines. By changing the blade pitch angle and generator torque constant at varying frequencies,

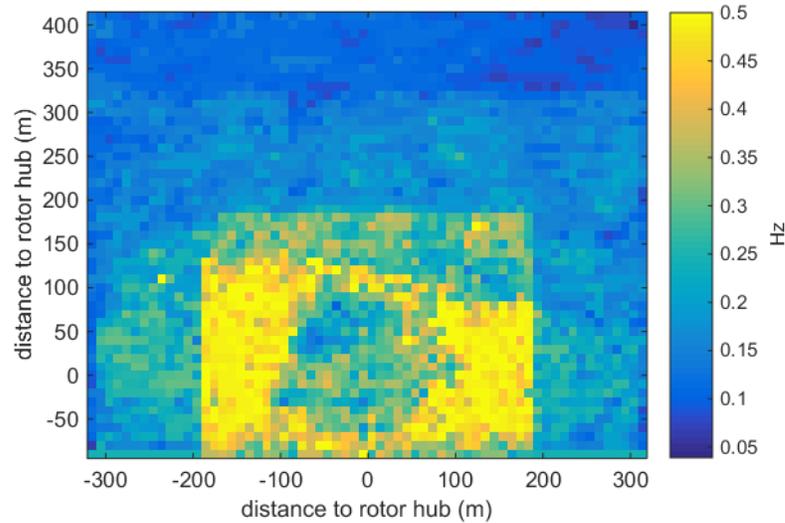


Figure 7.2: Frequency content of the flow velocity field at 4D downstream of the upstream turbine. Specifically, this shows the -20dB bandwidth of the fast Fourier transform of the velocity signal for sample points at 4D downstream. Note that the velocity signal was sampled at 1Hz. The pure yellow may indicate that the frequency at these points exceeds 0.5Hz.

various dynamics of the system are excited.

The data from the simulations were sampled by recording complete 3 dimensional snapshots of the flow field at 1 s intervals. The sampling time was determined by doing a frequency analysis of the flow in a two-turbine array. In particular, Figure 7.2 shows that a majority of the frequency content in the rotor swept area (63 m around the origin) of the wake is of lower frequency, i.e. less than 0.5 Hz. There is higher frequency at the edges of the wake due to the presence of a shear layer. For this analysis, it should be noted that the Fast Fourier Transform in Figure 7.2 is based on 1 Hz data. The Nyquist frequency is thus 0.5 Hz. Much of the information in the wake has a lower frequency than 0.5 Hz. It should also be noted that nothing definitive can be said about higher frequencies with this analysis. However, to capture the dominant characteristics of the system, i.e. the dynamics with lower frequencies, it appears that a 1 Hz sampling rate is sufficient for the construction of the IOROM based on SOWFA simulations.

The time step of the SOWFA simulation is 0.01 s. To collect data for 1800 s, this would amount to 180,000 snapshots of data with 1.2 million sampled grid points for each velocity component. This is impractical to store and analyze. As a result, only one snapshot of the flow is taken per 100 simulation time steps. The resulting IOROM will be a discrete time model with a time step of 1 s.

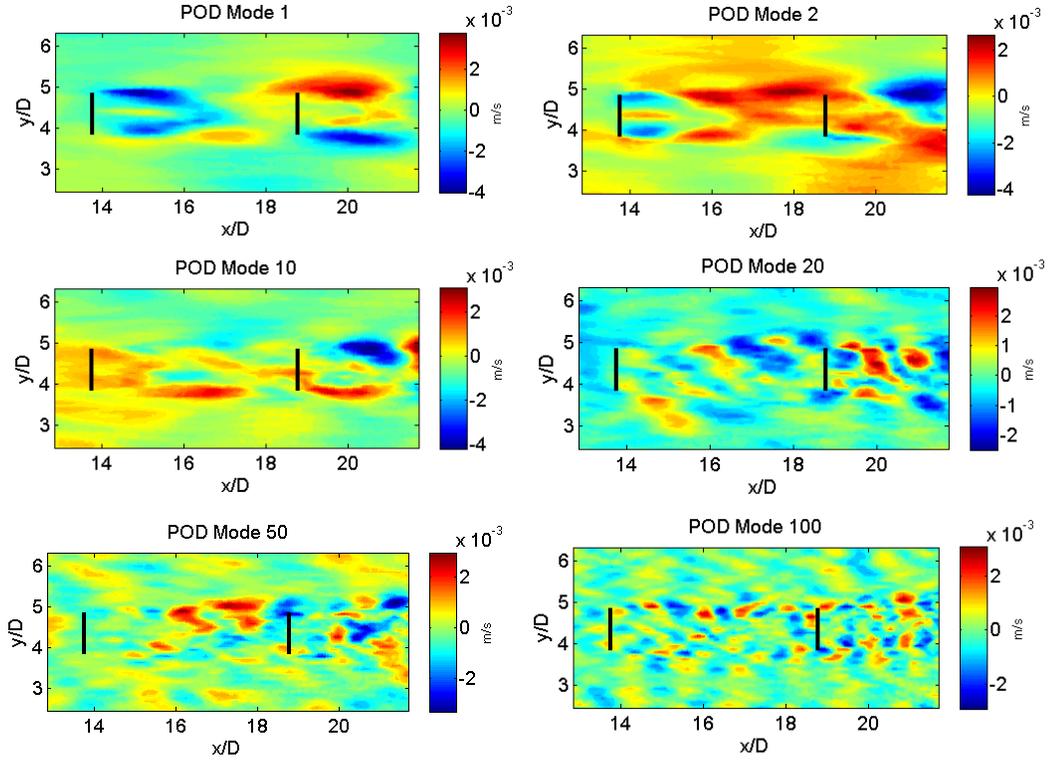


Figure 7.3: POD modes 1, 2, 10, 20, 50, and 100 of the SOWFA simulation.

### 7.3.3 Results

The results shown in this section were obtained using the IOROM constructed from data from the simulation scenario described in the previous sections. Note that the resulting flow field plots have been rotated in this section so that the flow is shown moving from left to right. The upstream and downstream turbines are indicated by black lines in the flow field figures in this section.

To construct the IOROM, the flow in the two-turbine array was sampled at 1.2 million grid points in the SOWFA simulations. Specifically, the three velocity components were recorded at each grid point resulting in 3.6 million states. The velocities are stacked into a single vector and collected in a snapshot matrix,  $X_0$  (4.34). Similarly, the inputs  $(\beta_1, \gamma_1, \beta_2, \gamma_2)$  and outputs  $(P_1, P_2)$  are recorded at each IOROM time step. The POD modes of the snapshot matrix,  $X_0$  (4.16), were computed using the MapReduce approach presented in [118] and summarized in the appendix (Section 7.6). The IOROM was constructed with 1000 snapshots at 1s intervals for a total time of 1000s. This indicates that the lowest frequency that this IOROM could capture is on the order of  $10^{-3}$ . Figure 7.3 shows the POD modes 1, 2, 10, 20, 50, and 100 of the streamwise velocity component at hub height (90m). The POD modes are used to project the system onto a low-dimensional subspace

such that direct N4SID can be used to identify a low-order model of the wind farm. It should be noted that these modes were computed after subtracting out the baseflow. As a result, the IOROM can be thought of as a linear model that represents the fluctuations about an equilibrium point, or mean flow. The first mode contains low-frequency spatial information and is the most energetic mode of the system. Mode 100 has high-frequency spatial information and represents a small amount of energy in the system. Figure 7.4 (left) shows the relative energy contained in each mode of the system. For this example, 150 modes were chosen. The order  $r$  of this model was selected based on the model fit error (4.43) plotted in Figure 7.4 (right). If the order of the model was chosen based on retaining 99% of the energy in the POD modes, almost all of the modes would have been used to construct the IOROM. Instead, the order of the model was chosen based on the model fit error, which indicates that the model performance does not get significantly better after 150 modes. In fact, by including more modes, the IOROM may be overfitting to the nonlinearities of the system which may degrade the performance of the model.

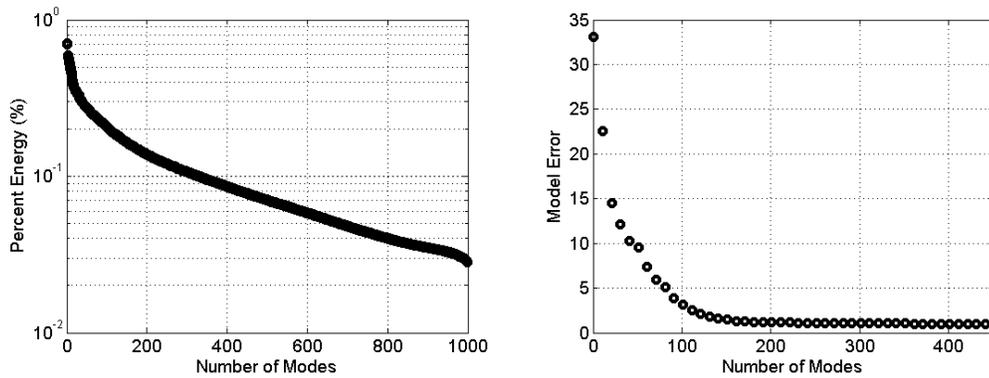


Figure 7.4: (Left) Percent energy of each mode in the SOWFA simulation. (Right) Model error computed based on the number of modes used to identify a model of the system.

The results of the IOROM can be seen in Figure 7.5. Specifically, the full flow field is computed from the results of the reduced-order model using (4.41). The full flow field is shown to demonstrate the capability of the reduced-order model to capture the dominant structures of the flow field, see the top plot in Figure 7.5. By only selecting 150 modes, this reduced-order model will not be able to capture the high frequency spatial turbulence. In addition to the reconstruction, the input-output behavior was examined. Specifically, the bottom plot of Figure 7.5 shows the output of the 150 state IOROM and it can be seen that the input-output behavior of the system is retained.

Lastly, the IOROM was applied to a set of validation data to verify that the identified model would work under similar wind conditions but with a different forced input. The same inflow fields were used in both cases. In addition, the same IOROM that was used in

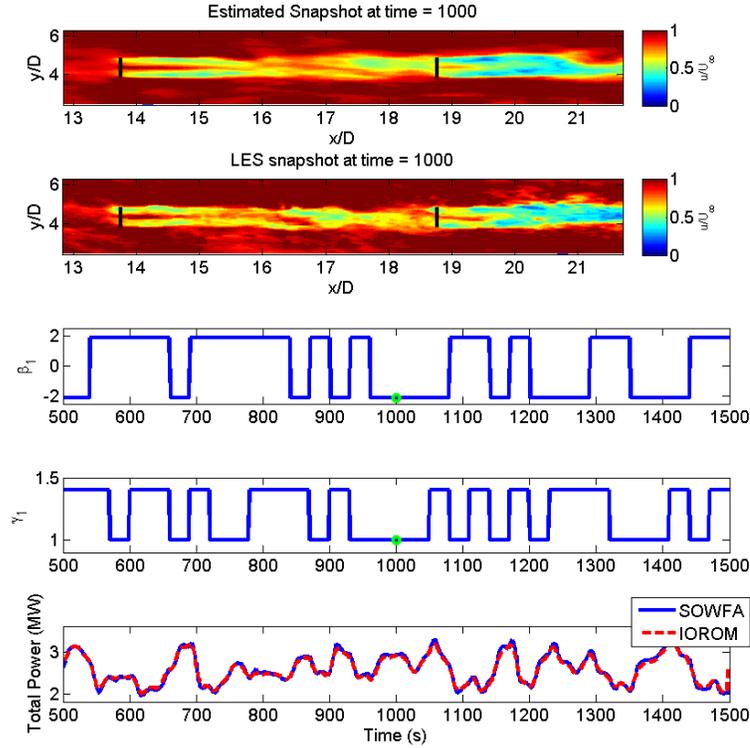


Figure 7.5: Flow reconstructed using the reduced-order model (top) and compared to SOWFA (second plot). The third and fourth plot show the inputs at the upstream turbine. Lastly, the bottom plot shows the output of the reduced-order model compared to the output of SOWFA.

Figure 7.5 is used with the validation data. Although the same inflow conditions are used, the turbulence generated by the upstream turbine will be different due to the different input. The inputs to the upstream turbine can be seen in the third and fourth plots in Figure 7.6. A Kalman filter (as described in Section 7.2) is implemented to handle the influence of this change in turbulence. Specifically, the filter parameters were set to  $P_0^+ = I_r$ ,  $Q_k = I_r$ , and  $R_k = 0.001I_{n_y}$ . These matrices were obtained by tuning them to the data set. Figure 7.6 shows that this IOROM is able to similarly reconstruct the dominant characteristics of the flow provided in the validation case. The results in Figure 7.6 indicate that this reduced-order model can be used as a predictive model for another similar data set. The power output was also examined to ensure that the input-output behavior is retained with this reduced-order model. The bottom plot of Figure 7.6 shows the output of the full-order SOWFA simulation and the reduced-order model. Again, the reduced-order model shows good agreement with the full-order simulation. This indicates that this reduced-order model

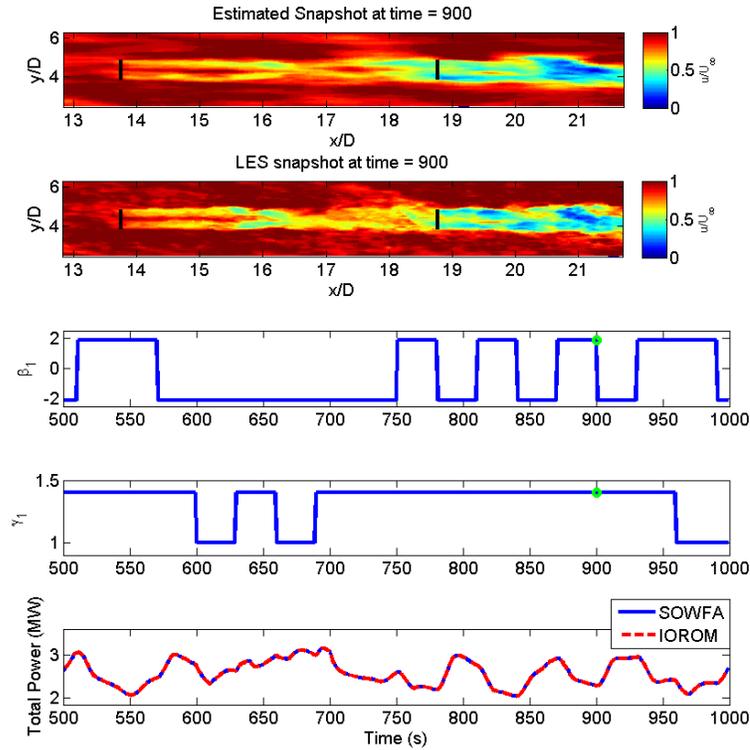


Figure 7.6: The model was validated using a different data set. The corresponding results are shown here. Flow reconstructed using the reduced-order model (top) and compared to SOWFA (second plot). The third and fourth plot show the inputs at the upstream turbine. Lastly, the bottom plot shows the output of the reduced-order model compared to the output of SOWFA.

is capable of predicting the output in similar scenarios which makes this model useful for control.

As mentioned previously, it is important to carefully choose the order of the model. If the IOROM is overfit to the data, the IOROM with the Kalman filter will not be able to predict other data sets, i.e., the Kalman gain of the Kalman filter will be large in an attempt to fit the estimated states with the measured outputs. Figure 7.7 shows the implications of selecting a 300 mode model. Although the input-output behavior is maintained (bottom plot), it can be seen that the reduced-order states (represented in the top plot) no longer represent the dominant dynamics of the system.

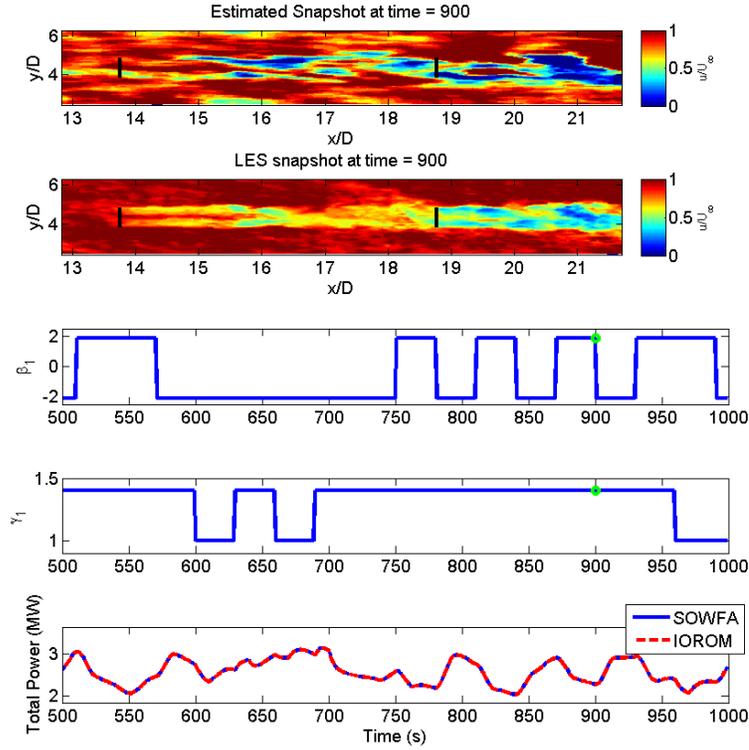


Figure 7.7: This figure shows an example of a IOROM that was overfit to the original data. The model used in this example had 300 states. When trying to use this IOROM (along with a Kalman filter) with a different data set, the input-output relationship is retained (bottom plot) but the reduced-order states are no longer able to represent the full-order spatial dynamics of the simulation.

## 7.4 Wind Tunnel Experiments

As a second example, data from wind tunnel experiments was used to construct an IOROM using the IOROM method. The experiments were completed in a closed loop wind tunnel at the Saint Anthony Falls Laboratory (SAFL) on the campus of the University of Minnesota. A 150 kW fan drives the flow into the tunnel test section, which has a length of 16 m and a cross-section of 1.7 m by 1.7 m. Coarse wire mesh and a honeycomb flow straightener condition the flow prior to entering a contraction with an area ratio of 6.6:1 upstream of the main test section. Immediately at the end of the contraction, a trip is placed to promote boundary layer growth. The model turbines were placed roughly 13 m downstream of the trip in the test section where a turbulent boundary layer thickness of  $\delta \approx 0.6$  m was measured under thermally neutral conditions. The neutral cases investigated herein set the air and floor temperature equal, and were held to within  $\pm 0.2^\circ\text{C}$ . Mean and fluctuating flow

statistics of the baseline turbulent boundary layer are provided in [62].

The turbine models tested in the tunnel are approximately 1:750 scale models of the Clipper Liberty 2.5 MW utility-scale wind turbine. The resulting dimensions for the model are a three-blade rotor with a diameter of 0.128 m (GWS/EP-5030x3 rotor). While the original design for this rotor was for use on a model aircraft, the blades were oriented such that the high pressure surface faced the inflow. In addition, the blade airfoil profiles are nearly flat. The hub height is 0.104 m and remains within the lower 25 % of the boundary layer, similar to the full-scale turbine. The tip-speed ratio (TSR),  $\lambda$ , was controlled by selecting the model turbine generator and is on the lower end of the  $\lambda$  operating range of standard turbines and provides a detailed comparison to wind farm experiments completed at the same velocity. The TSR can be defined as  $\lambda := \frac{\omega R}{U_\infty}$ , where  $\omega$  is the rotor speed [rad/s],  $R$  is the radius [m], and  $U_\infty$  is the freestream velocity [m/s]. The free-spinning TSR for the model turbine in the undisturbed boundary layer is  $\lambda \approx 4.5$ , while typical values for utility-scale wind turbines range between  $\lambda \approx 3.5$  and 10.

The model turbines have a small DC generator in which a voltage output can be measured or a voltage input can be applied to control the turbine operating condition. The DC voltage input is restricted to lie within  $\pm 1.25$  V. A zero voltage input corresponds to a free-spinning turbine. In this condition, the turbine operates at a high TSR. Applying a positive voltage places a torque on the motor shaft causing the turbine to operate at a lower TSR. A properly chosen voltage results in the turbine operating near its optimal TSR,  $\lambda = 3$ . Thus the DC voltage input mimics the effect of the generator torque on a utility-scale turbine.

#### 7.4.1 Setup

Similar to the simulation example, the experiments in the wind tunnel used a two-turbine array, see Figure 7.8 (left). The upstream turbine was operated in two different states: (i) rated; (ii) derated. In the rated state, a 1.25 V input was applied from a DC power supply or function generator to the DC generator on the model turbine. This input applies a torque opposing the aerodynamic torque and controls the TSR. In the derated state, the turbine was allowed to operate under no load, i.e. a zero voltage input [63]. The rated case corresponds to a turbine operating at the optimal operating point. The derated case corresponds to a turbine operating at a suboptimal operating point. In these experiments, the voltage generated by each turbine is similar to the power generated by a utility-scale turbine. For more details on turbine voltage production and analysis verification, the reader is directed to [63].

The two turbines were placed in a row with a 5D spacing between the turbines, see Figure 7.8 (left). The experiments were run at a wind speed of 4.5 m/s under a neutral boundary layer

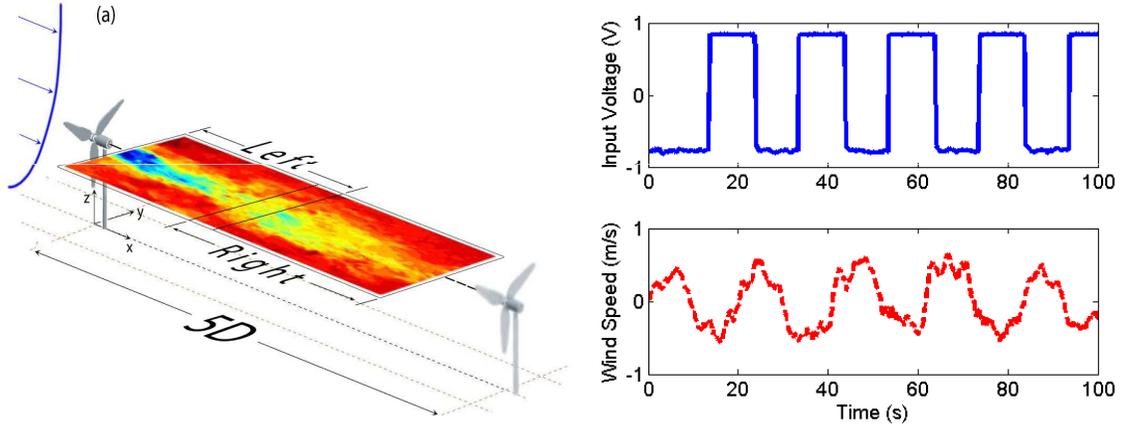


Figure 7.8: (Left) PIV setup in the wind tunnel. (Right) Forced input and corresponding output (streamwise velocity 3D downstream).

with 1.5% turbulence intensity for 100 s. This sample time was selected as it was the limit for the maximum number of samples for the data acquisition system when sampling at 10000 Hz. Wall parallel PIV was used to capture the varying physical characteristics in the wake created by the upstream turbine under rated and derated states with the overall goal of using the results to construct a reduced-order model that captures the dominant dynamics of the system and retains the appropriate input-output behavior. Specifically, the goal of the reduced-order model is to understand the impact of the upstream turbine control on the downstream wake. The use of wall parallel PIV simplified the problem by removing the extra factor of the boundary layer from the wake development. PIV uses a pulsating laser sheet synchronized with high resolution cameras to capture the instantaneous movement of seeding particles in the flow. Olive oil droplets on the order of 5 to 10 microns in diameter are injected into the wind tunnel and tracked by taking snapshots in time and comparing the locations of individual groups of particles to obtain the change in distance between the subsequent frames. The time between snapshots is known and therefore, the velocity vectors of the particles can be computed using two-dimensional spatial cross correlation in the interrogation windows in which the full image is subdivided (TSI PIV software). In this specific case, a fine  $32 \times 32$  pixel<sup>2</sup> interrogation window is used with a 50% overlap, providing a spatial resolution of approximately 1.8 mm in the streamwise and spanwise directions. Each run consisted of 700 snapshots over 100 s.

### 7.4.2 IOROM with the Wind Tunnel

The PIV results were able to provide measurements of the state, or velocity component at every pixel. The voltage input applied to the upstream turbine was recorded as the input. The output was the stream velocity at a point downstream. The output was computed from the PIV data. In contrast to the power output in the SOWFA simulations, this output will be linear in the state. Specifically, a square wave with a 0.05 Hz frequency was applied to the upstream turbine and the streamwise velocity at 3D downstream point was recorded. A square wave was chosen as an input for convenience based on the function generator and power supplies available in the wind tunnel at the time of the experiments.

Figure 7.8 (right) shows the forced input and the corresponding output. Note that the streamwise velocity has been filtered. This is reasonable since a turbine acts as a low pass filter due to the rotor inertia. Therefore, high frequency content in the wind speed can be filtered. An IOROM can be obtained using this input-output data along with the PIV data that measures the velocities (states). One snapshot of the flow field was captured every 0.14 s (7 Hz). The resulting IOROM will have a time step of 0.14 s. As with the IOROM obtained from simulations, this IOROM can be used to examine the flow field and approximate the input-output behavior.

### 7.4.3 Results

The results shown in this section were obtained using the IOROM constructed from experimental data from the wind tunnel. Similar to the simulation results, an IOROM is constructed in the form of (7.2). For these experiments, there is one input, the applied voltage at the upstream turbine  $u := V_1$ , and one output, the streamwise velocity 3D downstream  $y := U_{3D}$ . It should be noted that only the wake of the upstream turbine was captured by the PIV results. In other words, the upstream turbine is approximately at the inflow boundary and the downstream turbine is approximately located at the outflow boundary in the figures in this section. This setup is shown in Figure 7.8 (left).

For this particular example, there are approximately 26,000 grid points and two velocity components for each grid point resulting in approximately 52,000 states. The IOROM was generated using 350 snapshots for a total time of 50 s to compute the POD modes of the system. Figure 7.9 shows the POD modes 1, 2, 10, 20, 50, and 100 of the streamwise velocity component. Figure 7.10 shows the percent of energy contained in the POD modes. The IOROM constructed for these experiments uses 300 modes. The number of modes was chosen based on the model error metric (4.43) as was done in the simulation example.

The full flow field can be reconstructed from the reduced-order model shown in Figure 7.11.

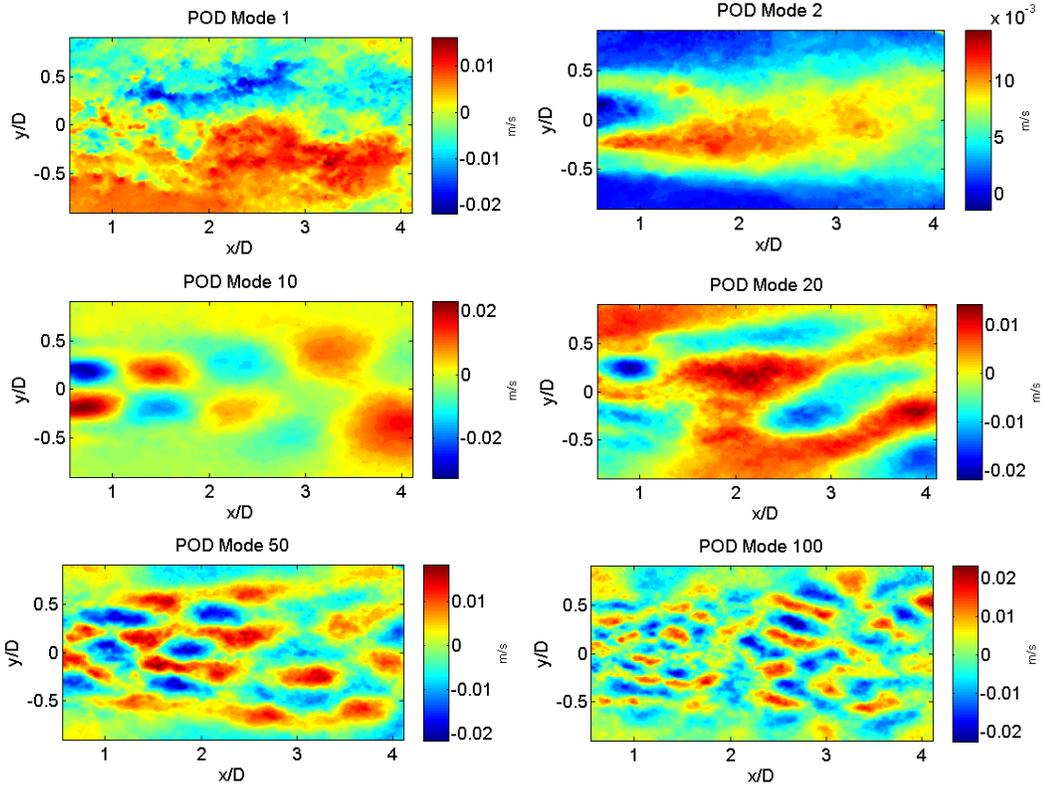


Figure 7.9: POD modes of the PIV wind tunnel experiments.

The output, i.e. the wind speed at 3D downstream, computed by the reduced-order shows good agreement with the PIV data (bottom plot of Figure 7.11). This IOROM model was validated with an additional experimental data set with the same input frequency. Similar to the simulation example, a Kalman filter was implemented to handle the influence of the turbulent characteristics of the flow that the IOROM does not capture. The parameters of the Kalman filter were set to  $P_0^+ = I_r$ ,  $Q_k = 20I_r$ , and  $R_k = 0.1I_{n_y}$ . The wind speed is subject to ambient turbulence and wake meandering (an oscillating effect of the wake), which makes it more difficult to predict. However, by implementing the Kalman filter, the results show good agreement with the PIV experiments. Specifically, the results of the reconstructed flow field and full experimental PIV data are shown in Figure 7.12. In addition, the output of the reduced-order model is compared to the experimental data. As with the simulation example, this indicates that the reduced-order model produces a good representation of the full-order system and can be used for control design in the wind tunnel.

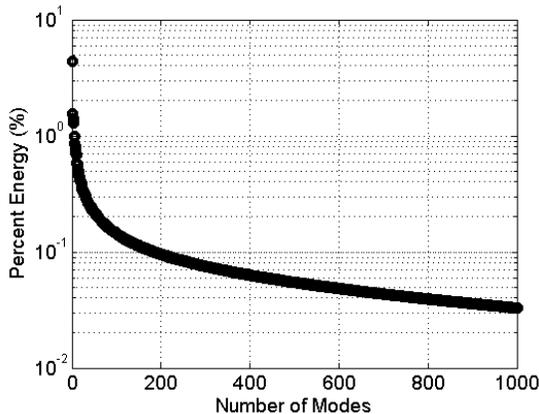


Figure 7.10: Energy of each POD mode from the wind tunnel data.

## 7.5 Conclusions

The IOROM method was applied to a high-fidelity wake model and wind tunnel experiments. Again, this approach takes advantage of characterizing the dominant dynamics in the flow and provides a low-order approximation of the flow. Using this low-order approximation, a reduced-order model can be constructed that retains the input-output behavior seen in the full-order model. In addition, this reduced-order model has a low computational cost and contains the necessary dynamics that are important for problems such as wind farm control. In both the simulation and experimental case, the IOROMs constructed were used on different data sets to demonstrate their predictive capabilities, which could be useful for control design and analysis in wind farms.

## 7.6 Appendix: Tall-Skinny QR Factorization

When using the IOROM method, described in Section 4.8, it is necessary to compute a singular value decomposition of the snapshot matrix,  $X_0$ . This matrix is a tall-skinny matrix where the dimensions are  $(n_x \times n_s)$ . For the SOWFA example in Section 7.3,  $n_x = 3.6$  million and  $n_s = 1000$ . This becomes difficult to do on a desktop computer. However, the MapReduce technique introduced in [118] can be implemented to compute the SVD for tall-skinny matrices. For the SOWFA example, 100 GB of data can be handled to compute the SVD of the snapshot matrix in approximately 1 hour. This process was computed in a serial manner and can be parallelized to speed up the process. The algorithm is briefly summarized below.

Consider  $X_0 \in \mathbb{R}^{n_x \times n_s}$  such that  $n_x \gg n_s$ . The  $X_0$  matrix can be divided into block

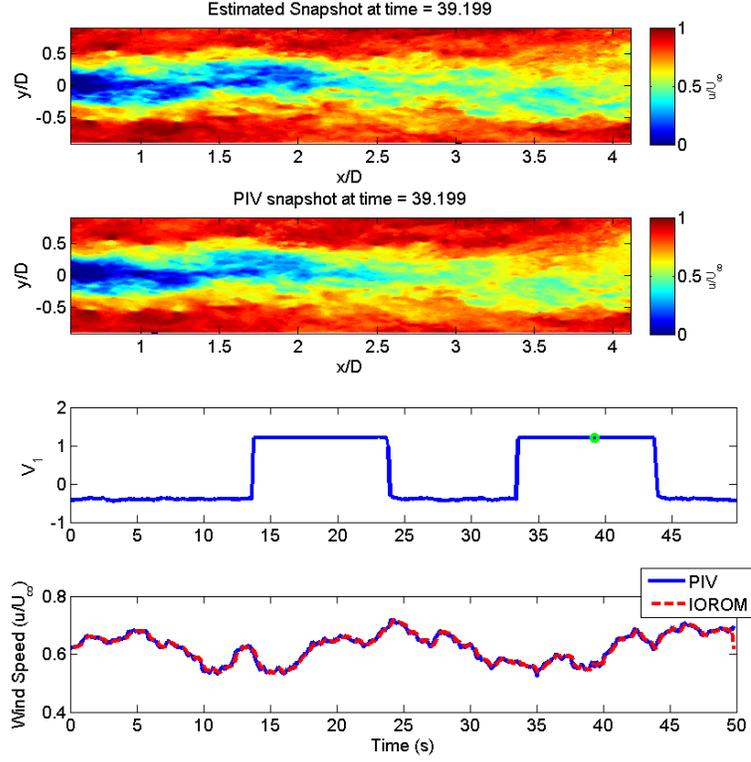


Figure 7.11: Flow reconstructed using the reduced-order model (top) and compared to the PIV results (second plot). The third plot shows the input voltage at the upstream turbine and the bottom plot shows the wind speed output of the reduced-order model compared to the PIV results.

matrices

$$X_0 = \begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_{n_d} \end{bmatrix} \quad (7.3)$$

where  $\bar{X}_i \in \mathbb{R}^{(n_x/n_d) \times n_s}$  and  $n_d$  is the number of divisions of the tall-skinny matrix  $X_0$ . A QR factorization is done for each  $\bar{X}_i$  matrix and arranged as follows:

$$X_0 = \begin{bmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \vdots \\ \bar{X}_{n_d} \end{bmatrix} = \underbrace{\begin{bmatrix} Q_1 & 0 & \dots & 0 \\ 0 & Q_2 & & \vdots \\ \vdots & & \ddots & \\ 0 & \dots & & Q_{n_d} \end{bmatrix}}_{Q \in \mathbb{R}^{n_x \times n_s n_d}} \underbrace{\begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_{n_d} \end{bmatrix}}_{R \in \mathbb{R}^{n_d n_s \times n_s}} \quad (7.4)$$

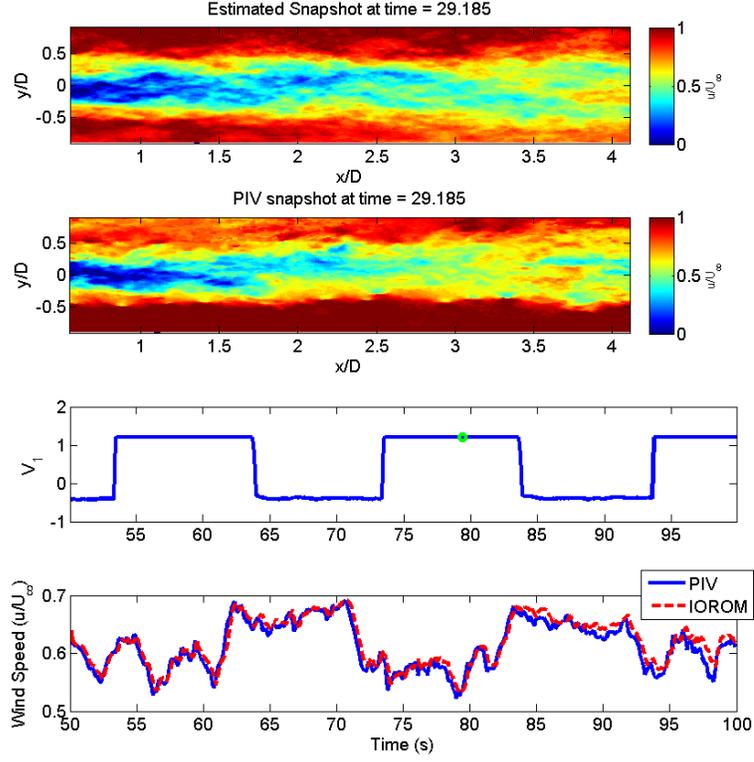


Figure 7.12: The model was validated using a different data set. The corresponding results are shown on the right. Flow reconstructed using the reduced-order model (top) and compared to the PIV results (second plot). The third plot shows the input voltage at the upstream turbine and the bottom plot shows the wind speed output of the reduced-order model compared to the PIV results.

The next step is to take the QR factorization of the  $R$  matrix:

$$\begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_{n_d} \end{bmatrix} = \tilde{Q}\tilde{R}. \quad (7.5)$$

where  $\tilde{R} \in \mathbb{R}^{n_s \times n_s}$ . Now, there is a  $\tilde{R}$  matrix that has dimensions  $(n_s \times n_s)$  rather than  $(n_x \times n_s)$ . The SVD is taken of  $\tilde{R}$

$$\tilde{R} = U_{\tilde{R}}\Sigma_{\tilde{R}}V_{\tilde{R}}^T, \quad (7.6)$$

where  $\Sigma_{\tilde{R}}$  contains the singular values of  $X_0$ . The POD modes, i.e. the  $U$  matrix of  $X_0$

(4.16), can be computed by

$$U = \bar{Q}U_{\tilde{R}}. \quad (7.7)$$

To avoid explicitly forming  $\bar{Q}$ , the POD modes can be computed as:

$$U = \begin{bmatrix} Q_1 U_{\tilde{R}} \\ Q_2 U_{\tilde{R}} \\ \vdots \\ Q_{n_d} U_{\tilde{R}} \end{bmatrix}. \quad (7.8)$$

This is the equivalent to computing the economy SVD of  $X_0$  in MATLAB.

## Chapter 8

# Conclusions

This thesis demonstrates that the coordinated control design problem to improve the performance of wind farms requires accurate wake models with a low computational cost. Low-fidelity models can provide useful insight into wake interactions, but lack the complexity to provide realistic wind farm results. Medium- and high-fidelity models are necessary for constructing an advanced control framework that can be used to optimize turbine placement and control design in a wind farm. In particular, the results seen in high-fidelity models differ significantly from the results obtained using low-fidelity models. The modeling aspect of the wind farm problem is important for developing viable wind farm control strategies. An experimental investigation was conducted to analyze the effect of individual turbine control on wind farm dynamics and the results were used to improve the existing static Park model. Model improvements included the incorporation of turbine dynamics that captured the input/output characteristics of a three-turbine array, which can be used for wind farm control in the wind tunnel. Wall-parallel PIV was used to identify physical wake characteristics based on varying turbine operating conditions. Voltage tests were used to characterize the frequency response of the system.

Alternative techniques for developing reduced-order wake models using data from experiments, as well as simulations, were addressed. Specifically, the data collected from simulations and experiments can be used to extract the dominant characteristics of a wind farm, which can be used for control design and analysis. The main reduced-order modeling technique introduced in this thesis was an extension to DMDC that can construct an input-output reduced-order model (IOROM) that can be used for high-dimensional systems. This IOROM approach takes advantage of characterizing the dominant dynamics in the flow and provides a low-order approximation of the flow. Specifically, this method combines POD with system identification to produce reduced-order models. This type of model reduction

has two main advantages. First, it relies on input-output data from a forced response and does not require adjoints. Second, the reduced-order model is constructed in a way that retains the physical meaning of the states. In addition, this IOROM approach is able to maintain robustness to small amounts of process noise. This IOROM has a low computational cost and contains the necessary dynamics that are important for problems such as wind farm control.

The IOROM method was then extended to construct reduced-order models for high-dimensional nonlinear systems and parameter varying systems. It was assumed that the nonlinear system has a collection of equilibrium operating points. This method has two main components. First, a reduced-order linear system is constructed at each equilibrium point using input/output/state data. Second, a parameter varying linearization is used to connect these linear models. This parameter varying IOROM approach was applied to a medium-fidelity actuator disk example to illustrate how this method is implemented. Lastly, this IOROM method was applied to a large eddy simulation and wind tunnel experiments at a single operating point to further demonstrate the feasibility of this method with large data sets.

Overall, this thesis shows that dominant characteristics of the flow dynamics within the wind farm can be extracted from data sets generated from simulations or experiments. This information can be used to construct useful low-order models that capture the essential dynamics of the system as well as the input/output behavior. These models can be used to design real-time controllers that can improve the overall performance of wind farms.

Additional work in this area of research can be done in future studies. Specifically related to the IOROM approach developed in this thesis, a better understanding of the optimal excitation signal and the sampling time of the data recorded is necessary to identify a model that most accurately captures the dominant dynamics of the system and the input-output behavior. Similar research has been done in the system identification literature and can be extended to this approach. One main source of error in the identified model and the high-dimensional system is due to the presence of process noise and measurement noise. Chapter 4 briefly addresses one way to analyze the impact of process noise. Additional insight is needed to quantify the uncertainty in the identified model.

In addition to modeling, this work shows preliminary applications to controls in Chapter 6. More sophisticated control designs can be done with models generated with this IOROM approach. Uncertainty quantification can be used to design robust controllers for wind farms. The examples used in this thesis for wind farms involve two-turbine arrays. The IOROM approach can generate models for systems with many inputs and many outputs, i.e., wind farms with many turbines. These reduced-order models of larger wind farms can also be used for control design. Lastly, this thesis addresses wind farms as parameter

varying systems and uses wind speed as the varying parameter. Wind direction is also an important parameter to consider when controlling a wind farm. The parameter varying IOROM approach can be used to include wind direction , as well as wind speed. These additional areas of research will aid in producing better representations of high-dimensional systems, which allow for better control designs especially in the area of wind farm control.

# Bibliography

- [1] Wiser, R., “Renewable Portfolio Standards in the United States - A Status Report with Data Through 2007,” LBNL-154E, Lawrence Berkeley National Laboratory, 2008.
- [2] Bitar, E. and Seiler, P., “Coordinated control of a wind turbine array for power maximization,” *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 2898–2904.
- [3] Gebraad, P. and Wingerden, J., “Maximum power-point tracking control for wind farms,” *Wind Energy*, Vol. 18, No. 3, 2015, pp. 429–447.
- [4] Johnson, K. and Fritsch, G., “Assessment of extremum seeking control for wind farm energy production,” *Wind Engineering*, Vol. 36, No. 6, 2012, pp. 701–716.
- [5] Marden, J. R., Ruben, S. D., and Pao, L. Y., “A model-free approach to wind farm control using game theoretic methods,” *Control Systems Technology, IEEE Transactions on*, Vol. 21, No. 4, 2013, pp. 1207–1214.
- [6] Schepers, J. and Van der Pijl, S., “Improved modelling of wake aerodynamics and assessment of new farm control strategies,” *Journal of Physics: Conference Series*, Vol. 75, IOP Publishing, 2007, p. 012039.
- [7] Machiels, L., Barth, S., Bot, E., Hendriks, H., and Schepers, J., “Evaluation of” Heat and Flux” Farm Control. Final Report,” *Wind Energy*, Vol. 2011, 2012, pp. 2010.
- [8] Johnson, K. E. and Thomas, N., “Wind farm control: Addressing the aerodynamic interaction among wind turbines,” *American Control Conference*, 2009, pp. 2104–2109.
- [9] Jensen, N. O., “A note on wind generator interaction,” Tech. Rep. Risø-M-2411, Risø, 1983.
- [10] Ainslie, J. F., “Calculating the flowfield in the wake of wind turbines,” *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 27, No. 1, 1988, pp. 213–224.

- [11] Churchfield, M. and Lee, S., “NWTC design codes-SOWFA,” <https://nwtc.nrel.gov/SOWFA>, 2012.
- [12] Yang, X., Sotiropoulos, F., Conzemius, R. J., Wachtler, J. N., and Strong, M. B., “Large-eddy simulation of turbulent flow past wind turbines/farms: the Virtual Wind Simulator (VWiS),” *Wind Energy*, Vol. 18, No. 12, 2015, pp. 2025–2045.
- [13] Annoni, J., Gebraad, P. M., Scholbrock, A. K., Fleming, P. A., and Wingerden, J.-W. v., “Analysis of axial-induction-based wind plant control using an engineering and a high-order wind plant model,” *Wind Energy*, 2015.
- [14] Goit, J. P. and Meyers, J., “Optimal control of energy extraction in wind-farm boundary layers,” *Journal of Fluid Mechanics*, Vol. 768, 2015, pp. 5–50.
- [15] Zhou, K., Doyle, J. C., Glover, K., et al., *Robust and optimal control*, Vol. 40, Prentice hall New Jersey, 1996.
- [16] Viberg, M., “Subspace-based methods for the identification of linear time-invariant systems,” *Automatica*, Vol. 31, No. 12, 1995, pp. 1835–1851.
- [17] Lall, S., Marsden, J. E., and Glavaški, S., “A subspace approach to balanced truncation for model reduction of nonlinear control systems,” *International journal of robust and nonlinear control*, Vol. 12, No. 6, 2002, pp. 519–535.
- [18] Loève, M., *Probability Theory; Foundations, Random Sequences*, New York: D. Van Nostrand Company, 1955.
- [19] Berkooz, G., Holmes, P., and Lumley, J. L., “The proper orthogonal decomposition in the analysis of turbulent flows,” *Annual review of fluid mechanics*, Vol. 25, No. 1, 1993, pp. 539–575.
- [20] Holmes, P., Lumley, J. L., and Berkooz, G., *Turbulence, coherent structures, dynamical systems and symmetry*, Cambridge university press, 1998.
- [21] Schmid, P. J., “Dynamic mode decomposition of numerical and experimental data,” *Journal of fluid mechanics*, Vol. 656, 2010, pp. 5–28.
- [22] Schmid, P. J., Li, L., Juniper, M., and Pust, O., “Applications of the dynamic mode decomposition,” *Theoretical and Computational Fluid Dynamics*, Vol. 25, No. 1-4, 2011, pp. 249–259.
- [23] Tu, J. H., Rowley, C. W., Luchtenburg, D. M., Brunton, S. L., and Kutz, J. N., “On dynamic mode decomposition: theory and applications,” *arXiv preprint arXiv:1312.0041*, 2013.

- [24] Mezić, I., “Analysis of fluid flows via spectral properties of the Koopman operator,” *Annual Review of Fluid Mechanics*, Vol. 45, 2013, pp. 357–378.
- [25] Proctor, J. L., Brunton, S. L., and Kutz, J. N., “Dynamic mode decomposition with control,” *arXiv preprint arXiv:1409.6358*, 2014.
- [26] Danowsky, B., Lieu, T., and Coderre-Chabot, A., “Control Oriented Aeroservoelastic Modeling of a Small Flexible Aircraft using Computational Fluid Dynamics and Computational Structural Dynamics,” *Proceedings of the AIAA SciTech Conference, San Diego, CA*, 2016.
- [27] Pao, L. Y. and Johnson, K. E., “Control of wind turbines,” *Control Systems, IEEE*, Vol. 31, No. 2, 2011, pp. 44–62.
- [28] Johnson, K. E., Pao, L. Y., Balas, M. J., and Fingersh, L. J., “Control of variable-speed wind turbines: standard and adaptive techniques for maximizing energy capture,” *Control Systems, IEEE*, Vol. 26, No. 3, 2006, pp. 70–81.
- [29] Burton, T., Sharpe, D., Jenkins, N., and Bossanyi, E., *Wind energy handbook*, John Wiley & Sons, 2001.
- [30] Jonkman, J., “NWTC Design Codes - FAST,” <https://nwtc.nrel.gov/FAST>, 2010.
- [31] Van Kuik, G. A., “The Lanchester–Betz–Joukowski limit,” *Wind Energy*, Vol. 10, No. 3, 2007, pp. 289–291.
- [32] Sanderse, B., “Aerodynamics of wind turbine wakes,” *Energy Research Center of the Netherlands (ECN), ECN-E-09-016, Petten, The Netherlands, Tech. Rep.*, 2009.
- [33] Pope, A., *Basic wing and airfoil theory*, McGraw-Hill, 1951.
- [34] Pope, S. B., *Turbulent flows*, Cambridge university press, 2000.
- [35] Howard, K. B., Singh, A., Sotiropoulos, F., and Guala, M., “On the statistics of wind turbine wake meandering: An experimental investigation,” *Physics of Fluids (1994-present)*, Vol. 27, No. 7, 2015, pp. 075103.
- [36] Chamorro, L. P. and Porté-Agel, F., “Turbulent flow inside and above a wind farm: a wind-tunnel study,” *Energies*, Vol. 4, No. 11, 2011, pp. 1916–1936.
- [37] Okulov, V. L. and Sorensen, J. N., “Instability of the far wake behind a wind turbine,” *Abs. 21th ICTAM-2004, Warsaw, Poland (http://ictam04.ippt.gov.pl/)*, 2004.

- [38] Kang, S., Yang, X., and Sotiropoulos, F., “On the onset of wake meandering for an axial flow turbine in a turbulent open channel flow,” *Journal of Fluid Mechanics*, Vol. 744, 2014, pp. 376–403.
- [39] Gebraad, P., Teeuwisse, F., Wingerden, J., Fleming, P., Ruben, S., Marden, J., and Pao, L., “Wind plant power optimization through yaw control using a parametric model for wake effects a CFD simulation study,” *Wind Energy*, Vol. 19, No. 1, 2016, pp. 95–114.
- [40] Fleming, P. A., Gebraad, P. M., Lee, S., van Wingerden, J.-W., Johnson, K., Churchfield, M., Michalakes, J., Spalart, P., and Moriarty, P., “Evaluating techniques for redirecting turbine wakes using SOWFA,” *Renewable Energy*, Vol. 70, 2014, pp. 211–218.
- [41] Sørensen, T., Thøgersen, M. L., Nielsen, P., and Jernesvej, N., “Adapting and calibration of existing wake models to meet the conditions inside offshore wind farms,” *EMD International A/S. Aalborg*, 2008.
- [42] González-Longatt, F., Wall, P., and Terzija, V., “Wake effect in wind farm performance: Steady-state and dynamic behavior,” *Renewable Energy*, Vol. 39, No. 1, 2012, pp. 329–338.
- [43] Larsen, G. C., Madsen Aagaard, H., Bingöl, F., Mann, J., Ott, S., Sørensen, J. N., Okulov, V., Troldborg, N., Nielsen, N. M., Thomsen, K., et al., “Dynamic wake meandering modeling,” Tech. rep., Risø National Laboratory, 2007.
- [44] Hao, Y., Lackner, M., Keck, R.-E., Lee, S., Churchfield, M., and Moriarty, P., “Implementing the dynamic wake meandering model in the NWTC design codes,” 2013.
- [45] Jonkman, B. J., “TurbSim user’s guide: Version 1.50,” 2009.
- [46] Sørensen, J. N. and Myken, A., “Unsteady actuator disc model for horizontal axis wind turbines,” *Journal of Wind Engineering and Industrial Aerodynamics*, Vol. 39, No. 1, 1992, pp. 139–149.
- [47] Sørensen, J. N. and Kock, C. W., “A model for unsteady rotor aerodynamics,” *Journal of wind engineering and industrial aerodynamics*, Vol. 58, No. 3, 1995, pp. 259–275.
- [48] Zikanov, O., *Essential computational fluid dynamics*, John Wiley & Sons, 2010.
- [49] Churchfield, M. J., Lee, S., Michalakes, J., and Moriarty, P. J., “A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics,” *Journal of turbulence*, , No. 13, 2012, pp. N14.

- [50] Fleming, P., Gebraad, P. M., Lee, S., Wingerden, J.-W., Johnson, K., Churchfield, M., Michalakes, J., Spalart, P., and Moriarty, P., “Simulation comparison of wake mitigation control strategies for a two-turbine case,” *Wind Energy*, Vol. 18, No. 12, 2015, pp. 2135–2143.
- [51] National Renewable Energy Laboratory, “NREL’s High-Performance Computing Capabilities,” [http://www.nrel.gov/energysciences/csc/high\\_performance\\_computing\\_capabilities](http://www.nrel.gov/energysciences/csc/high_performance_computing_capabilities), 2014.
- [52] Churchfield, M. J., Lee, S., Moriarty, P. J., Martinez, L. A., Leonardi, S., Vijayakumar, G., and Basseur, J. G., “A large-eddy simulation of wind-plant aerodynamics,” *AIAA paper*, Vol. 537, 2012, pp. 2012.
- [53] Annoni, J., Seiler, P., Johnson, K., Fleming, P., and Gebraad, P., “Evaluating wake models for wind farm control,” *American Control Conference (ACC), 2014*, IEEE, 2014, pp. 2517–2523.
- [54] Jonkman, J. M., Butterfield, S., Musial, W., and Scott, G., “Definition of a 5-MW reference wind turbine for offshore system development,” 2009.
- [55] Buhl, M., “NWTC Design Codes (WT-Perf),” *Last modified*, Vol. 6, 2012.
- [56] Hong, J., Toloui, M., Chamorro, L. P., Guala, M., Howard, K., Riley, S., Tucker, J., and Sotiropoulos, F., “Natural snowfall reveals large-scale flow structures in the wake of a 2.5-MW wind turbine,” *Nature communications*, Vol. 5, 2014.
- [57] Medici, D., “Experimental studies of wind turbine wakes: power optimisation and meandering,” 2005.
- [58] Knudsen, T. and Bak, T., “Data driven modelling of the dynamic wake between two wind turbines,” *Sysid 2012 16th IFAC Symposium on System Identification Brussels*, Vol. 16, 2012, pp. 1677–1682.
- [59] Koch, F., Gresch, M., Shewarega, F., Erlich, I., and Bachmann, U., “Consideration of wind farm wake effect in power system dynamic simulation,” *Power Tech, 2005 IEEE Russia*, IEEE, 2005, pp. 1–7.
- [60] Choi, J. and Shan, M., “Advancement of Jensen (Park) wake model,” *Proceedings of the European Wind Energy Conference and Exhibition*, 2013, pp. 1–8.
- [61] Gebraad, P. M. and Van Wingerden, J., “A control-oriented dynamic model for wakes in wind plants,” *Journal of Physics: Conference Series*, Vol. 524, IOP Publishing, 2014, p. 012186.

- [62] Singh, A., Howard, K. B., and Guala, M., “On the homogenization of turbulent flow structures in the wake of a model wind turbine,” *Physics of Fluids (1994-present)*, Vol. 26, No. 2, 2014, pp. 025103.
- [63] Howard, K., Hu, J., Chamorro, L., and Guala, M., “Characterizing the response of a wind turbine model under complex inflow conditions,” *Wind Energy*, Vol. 18, No. 4, 2015, pp. 729–743.
- [64] Stull, R. B., *An introduction to boundary layer meteorology*, Vol. 13, Springer Science & Business Media, 2012.
- [65] Schlichting, H. and Gersten, K., *Boundary-layer theory*, Springer Science & Business Media, 2003.
- [66] Andersen, S. J., Sørensen, J. N., Ivanell, S., and Mikkelsen, R. F., “Comparison of engineering wake models with CFD simulations,” *Journal of physics: Conference series*, Vol. 524, IOP Publishing, 2014, p. 012161.
- [67] Wang, S. and Seiler, P., “Gain scheduled active power control for wind turbines,” *AIAA Atmospheric Flight Mechanics Conference*, 2014.
- [68] Aho, J., Buckspan, A., Laks, J., Fleming, P., Jeong, Y., Dunne, F., Churchfield, M., Pao, L., and Johnson, K., “A tutorial of wind turbine control for supporting grid frequency through active power control,” *American Control Conference (ACC), 2012*, IEEE, 2012, pp. 3120–3131.
- [69] Katayama, T., *Subspace methods for system identification*, Springer Science & Business Media, 2006.
- [70] Budisic, M., Mohr, R. M., and Mezic, I., “Applied Koopmanism,” *arXiv:1206.3164*, 2012.
- [71] Willcox, K. and Peraire, J., “Balanced model reduction via the proper orthogonal decomposition,” *AIAA journal*, Vol. 40, No. 11, 2002, pp. 2323–2330.
- [72] Rowley, C., “Model reduction for fluids, using balanced proper orthogonal decomposition,” *International Journal of Bifurcation and Chaos*, Vol. 15, No. 03, 2005, pp. 997–1013.
- [73] Jovanović, M. R., Schmid, P. J., and Nichols, J. W., “Sparsity-promoting dynamic mode decomposition,” *Physics of Fluids (1994-present)*, Vol. 26, No. 2, 2014, pp. 024103.
- [74] Wood, G. D. et al., *Control of parameter-dependent mechanical systems*, Ph.D. thesis, Citeseer, 1995.

- [75] Wood, G., Goddard, P., and Glover, K., “Approximation of linear parameter-varying systems,” *IEEE Conference on Decision and Control*, Vol. 1, INSTITUTE OF ELECTRICAL ENGINEERS INC (IEE), 1996, pp. 406–411.
- [76] Van Overschee, P. and De Moor, B., “N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems,” *Automatica*, Vol. 30, No. 1, 1994, pp. 75–93.
- [77] Van Overschee, P. and De Moor, B., *Subspace identification for linear systems: TheoryImplementationApplications*, Springer Science & Business Media, 2012.
- [78] Juang, J.-N. and Pappa, R. S., “An eigensystem realization algorithm for modal parameter identification and model reduction,” *Journal of guidance, control, and dynamics*, Vol. 8, No. 5, 1985, pp. 620–627.
- [79] Ma, Z., Ahuja, S., and Rowley, C. W., “Reduced-order models for control of fluids using the eigensystem realization algorithm,” *Theoretical and Computational Fluid Dynamics*, Vol. 25, No. 1-4, 2011, pp. 233–247.
- [80] Juang, J.-N., Phan, M., Horta, L. G., and Longman, R. W., “Identification of observer/Kalman filter Markov parameters-Theory and experiments,” *Journal of Guidance, Control, and Dynamics*, Vol. 16, No. 2, 1993, pp. 320–329.
- [81] Majji, M., Juang, J.-N., and Junkins, J. L., “Observer/Kalman-filter time-varying system identification,” *Journal of guidance, control, and dynamics*, Vol. 33, No. 3, 2010, pp. 887–900.
- [82] Majji, M., Juang, J.-N., and Junkins, J. L., “Time-varying eigensystem realization algorithm,” *Journal of guidance, control, and dynamics*, Vol. 33, No. 1, 2010, pp. 13–28.
- [83] Majji, M., Juang, J.-N., and Junkins, J. L., “Time-Varying Deadbeat Controller Design,” *Journal of Guidance, Control, and Dynamics*, Vol. 35, No. 1, 2012, pp. 284–295.
- [84] van Wingerden, J.-W. and Verhaegen, M., “Subspace identification of bilinear and LPV systems for open-and closed-loop data,” *Automatica*, Vol. 45, No. 2, 2009, pp. 372–381.
- [85] Felici, F., Van Wingerden, J.-W., and Verhaegen, M., “Subspace identification of MIMO LPV systems using a periodic scheduling sequence,” *Automatica*, Vol. 43, No. 10, 2007, pp. 1684–1697.

- [86] Verdult, V. and Verhaegen, M., “Kernel methods for subspace identification of multi-variable LPV and bilinear systems,” *Automatica*, Vol. 41, No. 9, 2005, pp. 1557–1565.
- [87] Velazquez, A. and Swartz, R. A., “Cyclo-stationary linear parameter time-varying subspace realization method applied for identification of horizontal-axis wind turbines,” *SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring*, International Society for Optics and Photonics, 2013, pp. 86921I–86921I.
- [88] Adrian, R. J. and Westerweel, J., *Particle image velocimetry*, No. 30, Cambridge University Press, 2011.
- [89] Balajewicz, M., *A New Approach to Model Order Reduction of the Navier-Stokes Equations*, Ph.D. thesis, Duke University, 2012.
- [90] Noack, B. R., Schlegel, M., Morzynski, M., and Tadmor, G., *Galerkin method for nonlinear dynamics*, Springer, 2011.
- [91] Boulkroune, B., Kinnaert, M., and Zemouche, A., “POD-based state estimation of simulated moving bed chromatographic processes,” *Control Conference (ECC), 2013 European*, IEEE, 2013, pp. 3408–3414.
- [92] Rathinam, M. and Petzold, L. R., “A new look at proper orthogonal decomposition,” *SIAM Journal on Numerical Analysis*, Vol. 41, No. 5, 2003, pp. 1893–1925.
- [93] Smith, T. R., *Low-dimensional models of plane Couette flow using the proper orthogonal decomposition*, 2003.
- [94] Lanczos, C., Lanczos, C., Physicist, M., and Lanczos, C., *Linear differential operators*, Vol. 393, SIAM, 1961.
- [95] Bartels, R. H. and Stewart, G., “Solution of the matrix equation  $AX + XB = C$  [F4],” *Communications of the ACM*, Vol. 15, No. 9, 1972, pp. 820–826.
- [96] Rowley, C. W., Mezić, I., Bagheri, S., Schlatter, P., and Henningson, D. S., “Spectral analysis of nonlinear flows,” *Journal of fluid mechanics*, Vol. 641, 2009, pp. 115–127.
- [97] Williams, M. O., Kevrekidis, I. G., and Rowley, C. W., “A Data-Driven Approximation of the Koopman Operator: Extending Dynamic Mode Decomposition,” *Journal of Nonlinear Science*, Vol. 25, No. 6, 2015, pp. 1307–1346.
- [98] Dawson, S., Hemati, M. S., Williams, M. O., and Rowley, C. W., “Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition,” *arXiv preprint arXiv:1507.02264*, 2015.

- [99] Iungo, G., Santoni-Ortiz, C., Abkar, M., Porté-Agel, F., Rotea, M., and Leonardi, S., “Data-driven Reduced Order Model for prediction of wind turbine wakes,” *Journal of Physics: Conference Series*, Vol. 625, IOP Publishing, 2015, p. 012009.
- [100] Xu, P., Liu, J., Zeng, W., and Shen, Y., “Effects of errors-in-variables on weighted least squares estimation,” *Journal of Geodesy*, Vol. 88, No. 7, 2014, pp. 705–716.
- [101] Lin, F., Fardad, M., and Jovanovic, M. R., “Design of optimal sparse feedback gains via the alternating direction method of multipliers,” *Automatic Control, IEEE Transactions on*, Vol. 58, No. 9, 2013, pp. 2426–2431.
- [102] Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J., “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine Learning*, Vol. 3, No. 1, 2011, pp. 1–122.
- [103] Jovanovic, M. R. and Bamieh, B., “Componentwise energy amplification in channel flows,” *Journal of Fluid Mechanics*, Vol. 534, 2005, pp. 145–183.
- [104] Demmel, J. W., *Applied numerical linear algebra*, Siam, 1997.
- [105] Hemati, M. S., Williams, M. O., and Rowley, C. W., “Dynamic mode decomposition for large and streaming datasets,” *Physics of Fluids (1994-present)*, Vol. 26, No. 11, 2014, pp. 111701.
- [106] Poussot-Vassal, C. and Roos, C., “Generation of a reduced-order LPV/LFT model from a set of large-scale MIMO LTI flexible aircraft models,” *Control Engineering Practice*, Vol. 20, No. 9, 2012, pp. 919–930.
- [107] Gugercin, S., Antoulas, A. C., and Beattie, C., “H<sub>2</sub> model reduction for large-scale linear dynamical systems,” *SIAM journal on matrix analysis and applications*, Vol. 30, No. 2, 2008, pp. 609–638.
- [108] Panzer, H., Mohring, J., Eid, R., and Lohmann, B., “Parametric model order reduction by matrix interpolation,” *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs, Regelungs-und Informationstechnik*, Vol. 58, No. 8, 2010, pp. 475–484.
- [109] Takarics, B. and Seiler, P., “Gain Scheduling for Nonlinear Systems via Integral Quadratic Constraints,” *accepted to the American Control Conference*, 2015.
- [110] Packard, A., “Gain Scheduling via Linear Fractional Transformations,” *Systems and Control Letters*, Vol. 22, 1994, pp. 79–92.
- [111] Wu, F., “A generalized LPV system analysis and control synthesis framework,” *International Journal of Control*, Vol. 74, 2001, pp. 745–759.

- [112] Scherer, C., *Advances in linear matrix inequality methods in control*, chap. Robust mixed control and linear parameter-varying control with full-block scalings, SIAM, 2000, pp. 187–207.
- [113] Pfifer, H. and Seiler, P., “Robustness Analysis of Linear Parameter Varying Systems Using Integral Quadratic Constraints,” *International Journal of Robust and Nonlinear Control*, 2014.
- [114] Aster, R., Borchers, B., and Thurber, C., *Parameter estimation and inverse problems*, Academic Press, 2011.
- [115] Simley, E., Pao, L. Y., Frehlich, R., Jonkman, B., and Kelley, N., “Analysis of wind speed measurements using continuous wave LIDAR for wind turbine control,” *Proceedings of the 49th AIAA Aerospace Sciences Meeting*, 2011.
- [116] Harris, M., Hand, M., and Wright, A., “Lidar for turbine control,” *National Renewable Energy Laboratory, Golden, CO, Report No. NREL/TP-500-39154*, 2006.
- [117] Skogestad, S. and Postlethwaite, I., *Multivariable feedback control: analysis and design*, Vol. 2, Wiley New York, 2007.
- [118] Dean, J. and Ghemawat, S., “MapReduce: simplified data processing on large clusters,” *Communications of the ACM*, Vol. 51, No. 1, 2008, pp. 107–113.
- [119] Simon, D., *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*, John Wiley & Sons, 2006.
- [120] Kalman, R. E., “A new approach to linear filtering and prediction problems,” *Journal of basic Engineering*, Vol. 82, No. 1, 1960, pp. 35–45.