



## **SOWFA + Super Controller User's Manual**

Paul Fleming, Pieter Gebraad, Matt Churchfield,  
Sang Lee, Kathryn Johnson,  
John Michalakes, Jan-Willem van Wingerden,  
and Patrick Moriarty

**NREL is a national laboratory of the U.S. Department of Energy  
Office of Energy Efficiency & Renewable Energy  
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy  
Laboratory (NREL) at [www.nrel.gov/publications](http://www.nrel.gov/publications).

**Technical Report**  
NREL/TP-5000-59197  
August 2013

Contract No. DE-AC36-08GO28308

# **SOWFA + Super Controller User's Manual**

Paul Fleming, Pieter Gebraad, Matt Churchfield, Sang Lee, Kathryn Johnson, John Michalakes, Jan-Willem van Wingerden, and Patrick Moriarty

Prepared under Task No(s). WE115075

**NREL is a national laboratory of the U.S. Department of Energy  
Office of Energy Efficiency & Renewable Energy  
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at [www.nrel.gov/publications](http://www.nrel.gov/publications).

## NOTICE

This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at [www.nrel.gov/publications](http://www.nrel.gov/publications).

Available electronically at <http://www.osti.gov/bridge>

Available for a processing fee to U.S. Department of Energy and its contractors, in paper, from:

U.S. Department of Energy  
Office of Scientific and Technical Information  
P.O. Box 62  
Oak Ridge, TN 37831-0062  
phone: 865.576.8401  
fax: 865.576.5728  
email: <mailto:reports@adonis.osti.gov>

Available for sale to the public, in paper, from:

U.S. Department of Commerce  
National Technical Information Service  
5285 Port Royal Road  
Springfield, VA 22161  
phone: 800.553.6847  
fax: 703.605.6900  
email: [orders@ntis.fedworld.gov](mailto:orders@ntis.fedworld.gov)  
online ordering: <http://www.ntis.gov/help/ordermethods.aspx>

*Cover Photos: (left to right) photo by Pat Corkery, NREL 16416, photo from SunEdison, NREL 17423, photo by Pat Corkery, NREL 16560, photo by Dennis Schroeder, NREL 17613, photo by Dean Armstrong, NREL 17436, photo by Pat Corkery, NREL 17721.*



Printed on paper containing at least 50% wastepaper, including 10% post consumer waste.

## Acknowledgements

This SOWFA + Super Controller project has been developed by engineers at NREL and TU Delft under the CRADA collaboration CRD-11-440.

This work was supported by the U.S. Department of Energy under Contract No. DE-AC36-08GO28308 with the National Renewable Energy Laboratory and by the NWO Veni Grant no. 11930 “Reconfigurable floating wind farms.”

## Abstract

SOWFA + Super Controller is a modification of the NREL's SOWFA tool that allows for a user to apply multi-turbine or centralized wind plant control algorithms within the high-fidelity SOWFA simulation environment. The tool is currently a branch of the main SOWFA program, but will one day be merged into a single version.

This manual introduces the tool and provides examples such that a user can implement their own super controller and set up and run simulations. The manual provides a summary of SOWFA to allow for the customization of controllers and running of simulations. The details of SOWFA are reported elsewhere (Churchfield and Lee 2013, and Churchfield et al. 2012).

SOWFA + Super Controller, and this manual, are in early-release (alpha) mode. For questions not addressed in this manual, please post to the National Wind Technology Center's Computer-Aided Engineering Software Tools forum:

<https://wind.nrel.gov/forum/wind/viewforum?f=4>

# Contents

<b>Acknowledgements</b> .....	<b>iii</b>
<b>Abstract</b> .....	<b>iv</b>
<b>1 Introduction</b> .....	<b>1</b>
1.1 Overview .....	1
1.2 System Requirements.....	2
<b>2 Archive Contents</b> .....	<b>3</b>
2.1 Subdirectories.....	3
2.2 Scripts.....	4
<b>3 Programming and Compiling</b> .....	<b>5</b>
3.1 Controller Programming.....	5
3.2 Compiling.....	6
<b>4 Setting Up a Simulation and Running It</b> .....	<b>8</b>
4.1 Overview .....	8
4.2 Defining a Case.....	8
4.3 Running a Case .....	8
4.4 Results and Post-Processing.....	9
<b>5 Case Studies and Example</b> .....	<b>11</b>
5.1 Simulation Domain and Meshing.....	11
5.2 The NREL-DELFT (ND) Controller .....	11
5.2.1 Individual Controller .....	11
5.2.2 Super Controller .....	11
5.2.3 The ND_INPUT.txt File .....	11
5.3 Simulation Results .....	12
<b>6 References</b> .....	<b>14</b>

## List of Figures

Figure 1.1 Controller Interface.....	1
Figure 3.1 Controller Files.....	7
Figure 4.1 Epsilon Calculations.....	9
Figure 5.1 Domain size and mesh properties of the example YawLaminarClean case.....	13

## List of Tables

Table 5.1 ND_INPUT.txt of the <i>exampleYawLaminarClean</i> case.....	11
-----------------------------------------------------------------------	----

# 1 Introduction

## 1.1 Overview

SOWFA (Simulator for Off/Onshore Wind Farm Applications) is a computational fluid dynamics solver based on OpenFOAM coupled with FAST, which allows users to investigate wind turbine performance under variable atmospheric conditions (see Churchfield et al. 2012). The code is available online (see Churchfield and Lee 2013).

Recently, SOWFA has been modified to allow for the implementation of multi-turbine coordinated control of wind turbines. In effect, individual turbines implemented within SOWFA are now capable of communicating with a central controller, or with each other via a central router. This modification has been called “SOWFA + Super Controller” with the “super” controller being the central controller or router. (It was already possible to implement identical individual controllers.)

This overall architecture is shown in Figure 1.1. Note that the wind data, plant reference, and forecasting inputs to the master controller are currently a conceptual inclusion; at present, the focus is on the turbines only. The intention of the architecture is to be flexible enough to simulate any multi-turbine controller within the high-fidelity SOWFA environment.

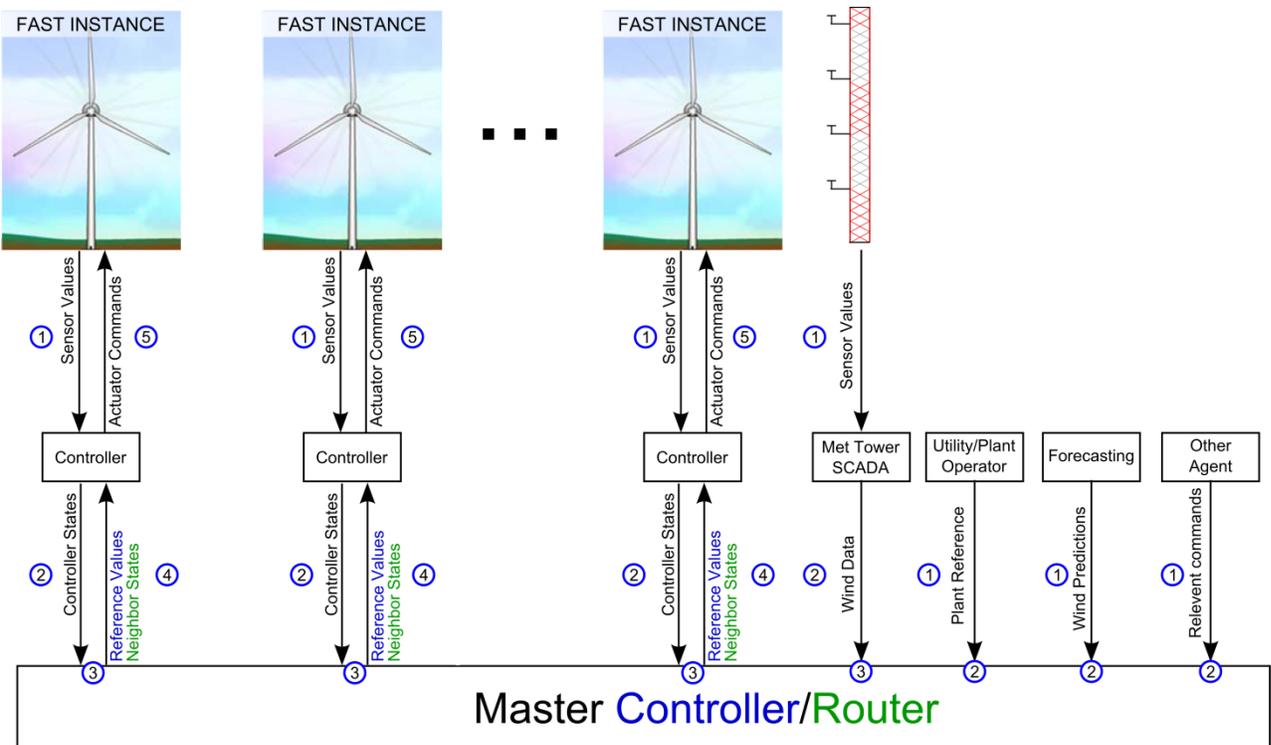


Figure 1.1 Controller Interface.

In this manual, the basics of using SOWFA + Super Controller (S+SC) are reviewed and some examples are presented to get the reader started. The code is very much in an “alpha” stage, but having been used for some internal studies it appears to be reliable enough for a cautious pre-release.

## 1.2 System Requirements

SOWFA and SOWFA+SC may be run over multiple processes on a single PC or across nodes of a cluster using the Message Passing Interface (MPI) for inter-process communication. A cluster is generally required to satisfy memory and computational requirements for a typically sized SOWFA domain. A specially modified version of OpenFOAM allows the super controller process to be assigned its own MPI task. The minimum number of MPI tasks is the number of FAST instances (one per simulated wind turbine) plus one additional task for the super controller process.

Finally, users will need to have compilers capable of compiling both the Fortran and C files, which are used for different components. In our own work we use standard Linux compilers (gcc/g++ and gfortran).

## 2 Archive Contents

S+SC is delivered as an archive. In this chapter, the contents of the archive are briefly reviewed. Note that this archive alone is not sufficient to compile and run S+SC. Additionally, the user must obtain and install:

**Custom OpenFOAM** Along with this archive, you should obtain a custom version of OpenFOAM, called OpenFOAM-2.1.sowfa (with 2.1 denoting the version 2.1.x) and gcc (version 4.3.4 or above). Make sure to include gfortran in the GNU compiler (Free Software Foundation, Inc. 2013).

### 2.1 Subdirectories

The root directory of the archive contains several subdirectories.

**src** This contains the source code for the S+SC project.

**src/openfast** Root directory of source that contains FAST and super controller code.

**src/openfast/FAST/Source/** Location of the FAST source code and location of individual and super controller source code (see section 3.1 for more details.)

**src/openfast/bin** Location of an .objmake file, that will compile just the FAST version and super controller. It is useful to use this during initial compilations to check for bugs, because the full build takes longer..

**src/fastturb** Actuator line code coupled to FAST.

**src/finiteVolume** Member functions to compute the buoyant pressure force.

**run** Directory in which to place the input files of your run cases (each case in a separate subdirectory). The output files of the runs will also be placed in this directory. The directory has the typical structure of an OpenFOAM case, having constant and system subdirectories (See the OpenFOAM User Guide (The OpenFOAM Foundation 2013) for more details). Further details of the contents of run case directory for a typical SOWFA case are provided in 4. In the S+SC archive, the run directory comes with an example case exampleYawLaminarClean. Specific details for the provided example are given in chapter 5.

**applications** This contains the solvers and a utility to initialize the atmospheric boundary layer flow. The executables are located in the bin directory.

**lib** This contains the run-time libraries generated by codes in the src directory.

## 2.2 Scripts

Additionally, there are also several scripts located in the root, which are used in compiling S+SC:

**sourceme** Configure and source this script to establish the location of the customized OpenFOAM version on your computer

**cleanBuildScript** Run prior to building (or before sending out) to clean up compilation directories, and to build S+SC and link the different parts of the OpenFOAM atmospheric boundary layer simulator to FAST.

See Section 3.2 for more instructions for compiling S+SC.

## 3 Programming and Compiling

### 3.1 Controller Programming

Programming a super controller into SOWFA focuses on editing several key files. All of these files are located within the folder `src/openfast/FAST/Source/directory`.

**turbineControllers/TC\_***suffix.c* Individual turbine controller implemented for each turbine.

**turbineControllers/SC\_***suffix.c* The super controller.

**turbineControllers/TC\_HDR\_***suffix.h* Place here any “include” statements needed.

**SOWFAController.h** Use this file to designate which controller files will be built and set the size of the buffer used to pass messages between Turbine Controller and Super Controller.

Note that the files that define the individual turbine controller and super controller (`TC_`*suffix.c*, `SC_`*suffix.c*, and `TC_HDR_`*suffix.c*) are all located within the subdirectory `turbineControllers` and each file contains a matching suffix. The concept is that you can have multiple controllers within this subdirectory that you can label by replacing the suffix in the filenames, and then use `SOWFAController.h` to select which controller to use at compile-time. `SOWFAController.h` is divided into code that is meant to remain unedited, and code that is used to perform the controller selection. For the remainder of this section we drop the suffix notation for cleanliness and refer only to `TC.C`, `SC.C` and `TC_HDR.h`.

The architecture of the control software has been set up to avoid burdening the control designer with re-implementing message passing between controllers and the super controller through the MPI (message passing interface) system. The previously described files, which the user edits, interface with fixed files which are intended to be left as is which implement the details of MPI. Figure 3.1 illustrates the file architecture, with the files to be edited shown in gray.

Figure 3.1 also illustrates the MPI system. The individual turbine control algorithm `TC.c` exchanges its input signals (sensor values) and output signals (actuator commands) with the FAST turbine simulation program by passing back and forth an array variable called `AVR_SWAP` (see Jonkman 2012 and Bossanyi 2009 for a specification of the structure of this array). Additionally, turbine data and command values are exchanged with the super controller through a predefined C structure called `TurbineStruct`, which is passed to the super controller through the MPI. Likewise, a variable `SC_STRUCT` is defined in the super controller code `SC.c`. These structures enable passing turbine data to the super controller and passing command data back to individual turbine controllers. The software architecture can be classified as a client/server setup in which the individual turbine controllers make requests to the super controller program to update and pass the `SC_STRUCT` variable. The `SC_STRUCT` structure passed to the `SC.c` code includes a turbine ID attribute (`turbineID`) that can be used to identify which turbine is calling the super controller.

Note that there is a single controller file `TC.c` that is run on all the turbines. The turbine ID variable that is passed to the Super Controller can be used to specify different control behaviors for each turbine. See the example `YawLaminarClean` case for an example.

Fig. 3.1 illustrates that the super controller routine has access to a previously written function called writeTurbineData.c. This function allows all individual and super controller data to be logged into a single file superCONOUT.csv.

Usage should be clear from the example case exampleYawLaminarClean. In our own work to date, most analysis is based on the file output from this function.

## 3.2 Compiling

After editing the source files, take the following steps to compile S+SC:

1. Go to root directory and type the command “source sourceme”. This will load the needed libraries and define the corresponding paths. By default, the sourceme script is prepared for the Peregrine High Performance Computing system at NREL. Make sure the job definitions in the runscript file will match the utilities and specifications of your cluster.
2. Run the cleanBuildScript. This will compile both FAST and the flow solver code written in OpenFOAM.
3. To compile FAST with the super controller only (and not FAST and OpenFOAM), run .objmake in src/openfast/bin/ to quickly debug the customized controller routines.

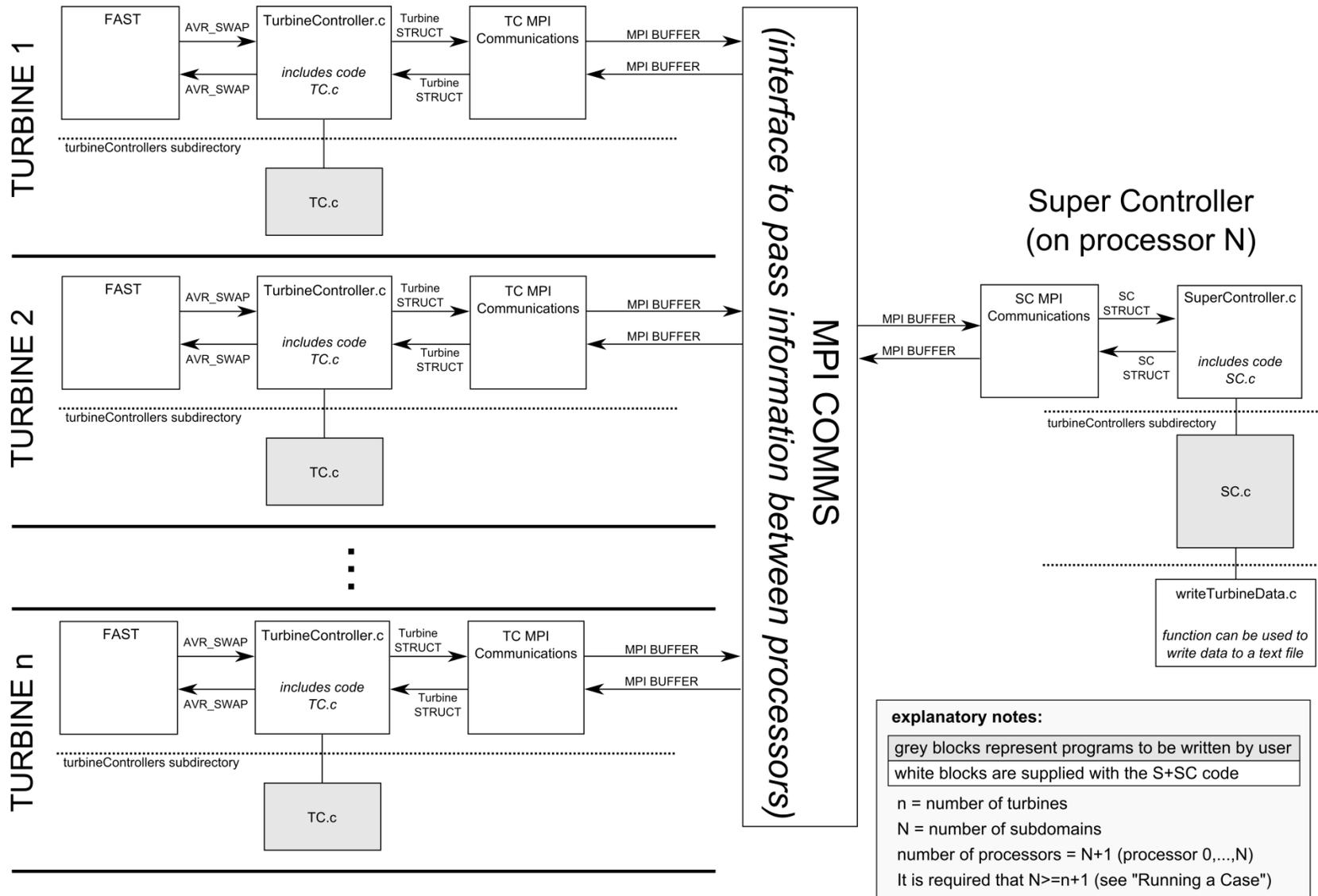


Figure 3.1 Controller Files.

## 4 Setting Up a Simulation and Running It

### 4.1 Overview

Once S+SC is compiled, this chapter will take you through the steps of setting up a simulation case and running it. As a reminder, the specific details on the workings of SOWFA are covered elsewhere (Churchfield and Lee 2013, and Churchfield et al. 2012).

### 4.2 Defining a Case

In addition to the dictionaries needed by OpenFOAM to define your case (such as `constant/transportProperties` and `constant/ABLProperties` (see The OpenFOAM Foundation 2013), SOWFA needs the following definition files :

- `constant/turbineArrayPropertiesFAST`, defining:
  - The locations of the wind turbine rotors
  - The parameters of the actuator line method that are shown in Fig. 4.1. These actuator line parameters define the search area for source terms in the Navier-Stokes equations, in the form of blade forces. The figure provides some rule-of-thumb guides in selecting these parameters. Choice of epsilon should be determined experimentally, but the figure provides a good starting selection.
- The FAST input files `primary_000.fst`, `primary_001.fst`, etc., defining properties and initial conditions of each of the turbines, (see the FAST User's Guide, Jonkman 2012). In the example case, the NREL 5MW reference turbine is used (Jonkman et al. 2009).

Note that some of the turbine properties are defined in both `constant/turbineArrayPropertiesFAST` and the FAST input files, and that these need to match.

### 4.3 Running a Case

Running a case involves steps 1-4 listed below.

In the example case `exampleYawLaminarClean` steps 2-4 are performed by running the runscript in the respective case folder `run/exampleYawLaminarClean` on a cluster. The example runscript is prepared for the MOAB job scheduling system on the Peregrine HPC system at NREL. Make sure the job definitions in the runscript file match the utilities and specifications of your cluster.

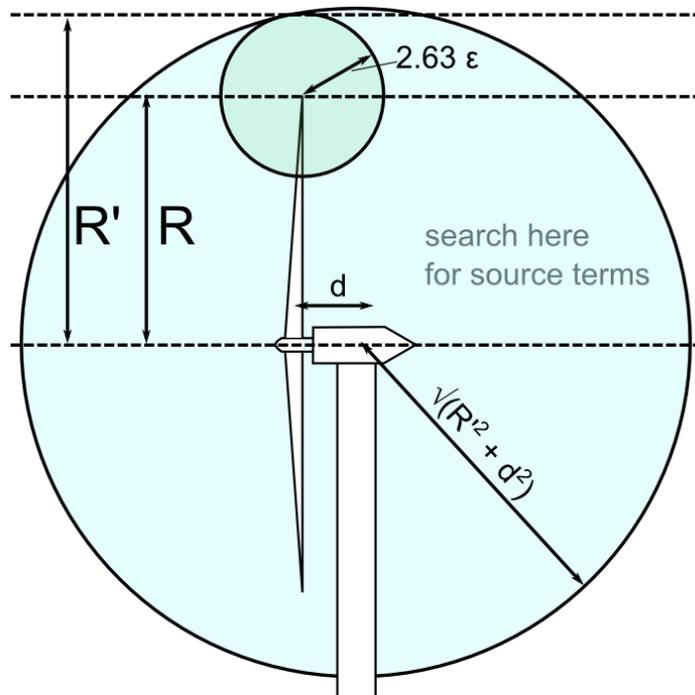
1. **Defining paths of needed libraries and loading OpenFOAM module.** This is done by going to the root directory and executing the command: `source sourceme`
2. **Preparing the mesh.** This is done by running the first part of the runscript, which prepares the mesh using the `blockMesh` OpenFOAM utility according to the definitions in the `constant/polymesh/blockmeshDict` file, then refines it according to the definitions in the `system/refineHexMesh.*` and `topoSet.*` files.
3. **Decomposing the mesh** onto the allocated processors of the cluster, according to the definitions in the `system/decomposeParDict` file. The number of processors used to run a case is the number of mesh subdomains plus one, because the super controller is running on a separate processor, while the FAST instances will be assigned to processors also

used for OpenFOAM calculations. This implies the number of subdomains should be at least the number of turbines, but is typically much larger.

4. **Running the windPlantPisoSolverFAST solver**, according to the specifications (e.g., time length of simulation) in the system/controlDict file.

On the MOAB job scheduling system on the NREL Peregrine HPC system, the runscript is submitted by executing the command: msub runscript.

For further details on the settings in blockmeshDict, decomposeParDict, refineHexMesh, topoSet and controlDict, refer to the OpenFOAM User Guide (The OpenFOAM Foundation 2013).



$$\begin{aligned} \epsilon &\approx 2 * \text{Smallest grid cell dimension} \\ \text{SmearRadius} &= 2.63 * \epsilon \text{ (Gaussian width)} \\ R' &= R + 2.63 * \epsilon \\ \text{EffectiveRadiusFactor} &= \sqrt{(R'^2 + d^2)} / R \end{aligned}$$

Figure 4.1 Epsilon Calculations.

## 4.4 Results and Post-Processing

Generally a SOWFA run will produce the following data:

- The CONOUT folder, containing CONOUT.out.\* and CONOUT.error.\* files, in which the messages and possible errors from each processor are written. The CONOUT.out.0000 and CONOUT.error.0000 files contain the general OpenFOAM outputs. The CONOUT.out.N and CONOUT.error.N files correspond to the super controller (where N is the largest processor number). Furthermore, if the writeTurbineData.c function is used, this function will log output signals of each turbine

in the CONOUT/superCONOUT.csv file. In the alpha distribution, we provide the importSuperCONOUT.m MATLAB script that can be used to read the superCONOUT.csv file into MATLAB.

- Outputs of the OpenFOAM sample utility, in the form of \*.vtk, \*.raw or \*.stl files. (See The Open- FOAM Foundation 2013). In the example cases, \*.vtk files containing horizontal and vertical cut-through plots of the flow are written to the sliceData folder. These files can be read using the ParaView visualization program (Kitware, Inc. 2013), for example.
- primary.out.\* files, which are the FAST output files of each turbine. (See the FAST user guide, Jonkman 2012).

## 5 Case Studies and Example

Included in the alpha distribution are an example super controller and an example simulation, and playing with these is probably the best way to get used to S + SC.

### 5.1 Simulation Domain and Meshing

For the *exampleYawLaminarClean* example case, the domain size and mesh properties are specified in Figure 5.1.

### 5.2 The NREL-DELFT (ND) Controller

The NREL-DELFT (ND) Controller is an example controller which has been used in several published studies (see for example Fleming et al. 2013c). This controller is also implemented in the *exampleYawLaminarClean* example case. It includes a basic implementation of an individual controller and a super controller.

#### 5.2.1 Individual Controller

The individual controller is essentially a direct implementation of the “baseline” NREL 5MW individual controller (fully defined in Jonkman et al. 2009). The individual controller also includes the necessary structure to receive commands from the super controller and to send back turbine data to the super controller.

#### 5.2.2 Super Controller

The super controller includes the code to communicate with individual turbines. It also includes a signature component of the ND controller; specifically, it expects a formatted input file. Currently the ND controller is “open-loop”, in that the super controller commands turbines based on a static set of instructions read from its input file `ND_INPUT.txt`, that is found in the particular case directory in the run directory.

The `SC_ND.c` super controller also uses the `writeTurbineData.c` function to write turbine sensor values received from each turbine to a file `CONOUT/superCONOUT.csv`.

#### 5.2.3 The `ND_INPUT.txt` File

In the implementation of the ND controller provided in the alpha release, the `ND_INPUT.txt` provides yaw reference settings. The first line in the file specifies the number of instruction lines. Each next line in the `ND_INPUT.txt` file specifies a change in the yaw reference setting of one of the turbines present in the simulation. Each of these lines has the following format:

*"time at which setting changes" "turbine of which setting changes" "yaw control set-point"*

lines:	meaning:
3	<i>3 more lines in this file</i>
0 1 0	<i>turbine 1 has yaw reference angle zero at time zero</i>
0 2 0	<i>turbine 2 has yaw reference angle zero at time zero</i>
10 1 40	<i>after 10s, yaw reference angle of turbine 1 is set to 40°</i>

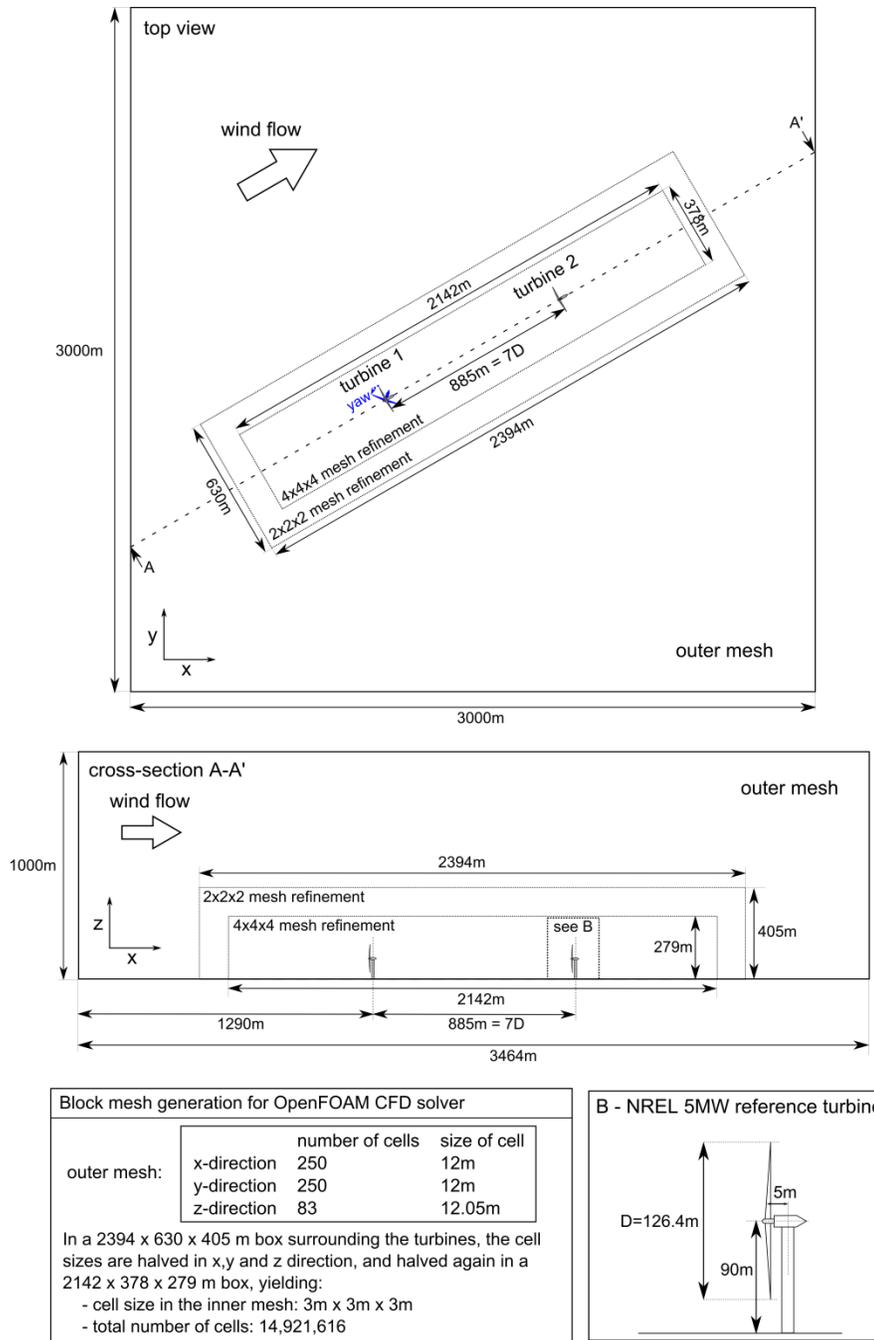
**Table 5.1** `ND_INPUT.txt` of the *exampleYawLaminarClean* case

In the exampleYawLaminarClean case for example, the ND\_INPUT.txt reads as in Table 5.1.

It is fairly straightforward to extend the implementation of the alpha release of the ND controller to other settings than the yaw setting.

### **5.3 Simulation Results**

For examples of simulation results acquired with S+SC, refer to Fleming et al. 2013a, b, and c.



**Figure 5.1 Domain size and mesh properties of the example YawLaminarClean case.**

## 6 References

- Bossanyi, E. (2009). “GH Bladed user manual.” Garrad Hassan and Partners Ltd .
- Churchfield, M.; Lee, S. (2013). NWTC Design Codes (SOWFA). <http://wind.nrel.gov/designcodes/simulators/SOWFA>.
- Churchfield, M.; Lee, S.; Michalakes, J.; Moriarty, P. (2012). “A Numerical Study of the Effects of Atmospheric and Wake Turbulence on Wind Turbine Dynamics.” *Journal of Turbulence* 13(14); pp. 1–32.
- Fleming, P.; Gebraad, P.; Lee, S.; van Wingerden, J.; Johnson, K.; Churchfield, M.; Michalakes, J.; Spalart, P.; Moriarty, P. (2013a). “Evaluating techniques for redirecting turbine wake using SOWFA.” *Proceedings of ICOWES 2013*.
- (2013b). “High-fidelity simulation comparison of wake mitigation control strategies for a two-turbine case.” *Proceedings of ICOWES 2013*.
- Fleming, P.; Gebraad, P.; van Wingerden, J.; Lee, S.; Churchfield, M.; Scholbrock, A.; Michalakes, J.; Johnson, K.; Moriarty, P. (2013c). “The SOWFA Super-Controller: A High-Fidelity Tool for Evaluating Wind Plant Control Approaches.” *Proceedings of EWEA*.
- Free Software Foundation, Inc. (2013). GCC, the GNU Compiler Collection. <http://gcc.gnu.org/>.
- Jonkman, J. (2012). NWTC Design Codes (FAST). <http://wind.nrel.gov/designcodes/simulators/fast>.
- Jonkman, J.; Butterfield, S.; Musial, W.; Scott, G. (2009). *Definition of a 5-MW Reference Wind Turbine for Offshore System Development*. NREL/TP-500-38060.
- Kitware, Inc. (2013). ParaView open-source data analysis and visualization application. <http://www.paraview.org/>.
- The OpenFOAM Foundation (2013). The OpenFOAM user manual. <http://www.openfoam.org/docs/user/>.