

Spring 4-8-2019

Gaussian Process Regression applied to Marine Energy Turbulent Source Tuning via Metamodel Machine Learning Optimization

Sterling Olson
University of New Mexico

Follow this and additional works at: https://digitalrepository.unm.edu/me_etds



Part of the [Mechanical Engineering Commons](#)

Recommended Citation

Olson, Sterling. "Gaussian Process Regression applied to Marine Energy Turbulent Source Tuning via Metamodel Machine Learning Optimization." (2019). https://digitalrepository.unm.edu/me_etds/165

This Thesis is brought to you for free and open access by the Engineering ETDs at UNM Digital Repository. It has been accepted for inclusion in Mechanical Engineering ETDs by an authorized administrator of UNM Digital Repository. For more information, please contact amywinter@unm.edu.

Sterling Stewart Olson

Candidate

Mechanical Engineering

Department

This dissertation is approved, and it is acceptable in quality and form for publication: *Approved*
by the Thesis Committee:

Dr. Sang Lee, Chair

Dr. Peter Vorobieff, Co-Chair

Dr. Humberto III Silva, Member

Dr. Jack C.P. Su, Member

Gaussian Process Regression applied to Marine Energy Turbulent Source Tuning via Metamodel Machine Learning Optimization

by

Sterling Stewart Olson

B.S., Nuclear & Radiological Engineering, Georgia Institute of
Technology, 2013

M.S., Chemical Engineering, University of New Mexico, 2015

M.B.A., University of New Mexico, 2018

THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

Mechanical Engineering

The University of New Mexico

Albuquerque, New Mexico

May, 2019

Dedication

To the pursuit of knowledge.

Acknowledgments

I would like to thank my mentor Chris Chartrand for his time and patience over the last four years teaching me to use the superior Linux operating system for numerical analysis. To Jesse Roberts for supporting me in my effort to apply project milestones to an academic pursuit. Thank you to Jack Su for his guidance and selection of the surrogate model investigation detailed within. Thank you to the UNM faculty who have helped me apply my work to a Master's thesis. Finally thanks to Tito Silva for working with me as a co-advisor and technical support in the numerical analysis.

Gaussian Process Regression applied to Marine Energy Turbulent Source Tuning via Metamodel Machine Learning Optimization

by

Sterling Stewart Olson

B.S., Nuclear & Radiological Engineering, Georgia Institute of Technology, 2013

M.S., Chemical Engineering, University of New Mexico, 2015

M.B.A., University of New Mexico, 2018

M.S., Mechanical Engineering, University of New Mexico, 2019

Abstract

Converting energy from the currents found within tidal channels, open ocean, rivers, and canals is a promising yet untapped source of renewable energy. In order to permit current energy converters for installation in the environment, the CECs must be shown to non-negatively impact the environment. While developing these model increased utility may be gained if researchers may optimize mechanical power while constraining environmental effects. Surrogate models have garnered interest as optimization tools because they maximize the utility of expensive information by building predictive models in place of computational or experimentally expensive model runs. Marine hydrokinetic current energy converters require large-domain simulations to estimate array efficiencies and environmental impacts. Meso-scale models typically represent turbines as actuator discs that act as momentum sinks and sources of turbulence. An OpenFOAM model was developed where actuator-disc k - ϵ turbulence was characterized using an approach developed for flows through vegetative canopies.

Turbine-wake data from laboratory flume experiments collected at two influent turbulence intensities were used to calibrate parameters in the turbulence-source terms in the k - ϵ equations. Parameter influences on longitudinal wake profiles were estimated using Gaussian-process regression with subsequent optimization achieving results within 3% of those obtained using the full model representation, but for as low as 27% of the computational cost (far fewer model runs). This framework facilitates more efficient parameterization of the turbulence-source equations using turbine-wake data.

Contents

List of Figures	ix
List of Tables	x
Nomenclature	xi
1 Introduction	1
1.1 Overview	1
2 Problem Background	4
2.1 Problem	7
2.2 Research Question	7
3 Governing Theory & Equations	9
3.1 Reynolds Average Navier-Stokes	9
3.2 k - ϵ Turbulence model	11
3.3 Momentum Theory	13
4 Framework Development	15
4.1 Flume Experiment	15
4.2 OpenFOAM Model Development	16
4.3 OpenFOAM Grid Study	19

Contents

4.4	Turbulence-source Equations	22
4.5	Metrics	23
4.6	Standard k - ϵ Model Results	24
4.7	PEST Calibration/Parameter Estimation	25
5	Surrogate Model Development	29
5.1	Gaussian Process Regression	29
5.2	Surrogate Model Sample Space	40
5.3	Building the Surrogate Model	41
5.4	Surrogate-model Fitting Error	43
5.5	Surrogate Optimization Framework	43
6	Results and Discussion	46
6.1	Surrogate & OpenFOAM Parameter Values Compared to Experiment	49
6.2	Limitations and Future Research	50
7	Conclusions	52
	References	53

List of Figures

2.1	Basic turbine types [15]	5
2.2	Environmental influences and turbine wake [4]	6
4.1	INFERMER Experimental Flume Tank [38]	16
4.2	Experimental Representation [38]	17
4.5	Uniform mesh refinement grid study results starting at $h_0 = 0.1m$ plotted at the spatial points of interest for each quantity of interest namely (a.) U^* at $TI_\infty = 3\%$, (b.) U^* at $TI_\infty = 15\%$, (c.) k at $TI_\infty = 3\%$, and (d.) k at $TI_\infty = 15\%$	21
4.6	Comparison of experimental data (red dots) to the standard $k-\epsilon$ actuator disc results (blue dash-dot curve) for QoIs U^* at (a) $I_{3\%}$ and (b) $I_{15\%}$ and I at (c) $I_{3\%}$ and (d) $I_{15\%}$	28
5.1	Two samples data sets	30
5.2	Shows two random variable examples	32
5.7	Flow diagram for the surrogate-model framework.	45
6.1	Comparison of experimental data (red dots) to optimized results using OpenFOAM (solid curve) and GPR (black-dashed curve) for QoIs U^* at (a) $I_{3\%}$ and (b) $I_{15\%}$ and I at (c) $I_{3\%}$ and (d) $I_{15\%}$. The standard $k-\epsilon$ actuator disc results are presented for reference (blue dash-dot curve).	51

List of Tables

4.1	Boundary condition applied for each variable and patch	19
4.2	Performance metrics.	23
4.3	Actuator disc with standard k - ϵ model errors.	25
4.4	PEST initial guess and tunable parameter bounds	27
5.1	Optimized GPR kernel hyperparameters.	42
5.2	GPR surrogate model performance as compared to "true" OpenFOAM model.	43
6.1	GPR-PEST and OpenFOAM-PEST optimized parameters and corre- sponding performances.	47
6.2	Errors in simulated QoIs using optimized parameters from the GPR- PEST and OpenFOAM-PEST calibrations.	50

Nomenclature

$I_{15\%}$ GPR model predicts wake TI for 15% TI BC.

$I_{3\%}$ GPR model predicts wake TI for 3% TI BC.

L_{∞} Maximum Error.

$MAPE$ Mean Absolute Percentage Error.

MSE Mean Squared Error.

Q Quantity of Interest.

TI Turbulent Intensity.

$U_{15\%}$ GPR model predicts wake velocity for 15% TI BC.

$U_{3\%}$ GPR model predicts wake velocity for 3% TI BC.

ϵ Turbulent kinetic energy dissipation.

k Turbulent kinetic energy.

R Real Random Variable.

exp Experimental Data.

sim OpenFOAM Model.

Nomenclature

srgt Surrogate Model.

tkε Turbulent kinetic energy.

BC Boundary Condition.

BEM Blade element method.

CEC Current Energy Converter.

CFD Computational Fluid Dynamics.

D3D Delft3D-Flow.

GP Gaussian Process.

GPR Gaussian Process Regression.

GPR-PEST PEST optimizations results from GPR model.

HAWT Horizontal Axis Water Turbine.

IC Initial Condition.

MHK Marine and Hydrokinetic.

OpenFOAM-PEST PEST optimization results from OpenFOAM model.

PEST Model independent Parameter Estimation Software.

QoI Quantity of Interest.

RANS Reynolds Averaged Navier-Stokes.

SNL Sandia National Laboratories.

Nomenclature

SWE Shallow Water Equations.

VAWT Vertical Axis Water Turbine.

VOF Volume of Fluid.

Chapter 1

Introduction

1.1 Overview

Electricity generated by marine hydrokinetic (MHK) devices holds promise as a renewable, domestic energy source that can help meet the US National demand; offsetting the need for traditional power sources such as coal and gas that can have detrimental long-term impacts on global climate and support foreign dependencies that put the US at risk. However, research is needed to make MHK energy cost competitive as well as to understand its potential environmental effects. As such, potential industry investors need to reduce costs and time associated with meeting regulatory requirements through an improved understanding of the potential environmental impacts associated with deploying arrays of MHK devices. The work described here is part of a larger project which focuses on developing and applying numerical modeling tools to assess potential changes to the environment caused by the operation of MHK arrays. Application of the tool will support optimal placement of MHK devices by determining optimal array spacing that maximizes array power performance and minimizes potential environmental effects.

Specifically, for the MHK industry to succeed, it is imperative to balance MHK power generation with ecosystem and anthropogenic considerations. This research

Chapter 1. Introduction

at large aims to develop computation tools allowing investigators to understand the interrelationship between the number, size, efficiency, and configuration of MHK devices and any subsequent effects this may have on the deployment environment. Research at Sandia National Laboratories (SNL) and elsewhere [65] has shown that it is unlikely that a small number of devices will create environmental concern due to converting only a small fraction of the available power from a large resource. However, commercial-scale deployments must be analyzed to ensure environmental tipping points are not exceeded. To characterize these potential effects, tools that allow the virtual design of array layouts for optimization are needed. The tools must simulate flows through and around MHK arrays, as well as MHK-specific noise generation and propagation, and then determine the environmental response. Understanding MHK-driven changes to the physical environment facilitate evaluation of how these changes may affect the local ecosystem and aquatic life.

Current Energy Converters (CECs) are a subset of MHK devices capable of converting energy found within tidal channels, open ocean, rivers, and canals. Development of numerical tools for CECs can be hastened by augmenting currently developed open-source oceanic and estuary models, such as Delft-3D (D3D). D3D is a software suite that allows users to calculate three-dimensional hydrodynamics, sediment transport, morphology, and water quality for fluvial, estuary and coastal environments [11]. These capabilities made D3D a natural starting point for the implementation of a CEC module. D3D is capable of solving non-steady hydrodynamic flow and transport equations. SNL Water Power Technologies group then needed to only implement the CEC module into open source FORTRAN source code.

Research at Sandia led to the publication of SNL-Delft3D-CEC which adds a custom CEC module into Delft3D. In addition to the standard tunable parameters such as coefficient of thrust and power, tunable parameters were added to the k - ϵ turbulence model. While ideal for mesoscale modeling Delft3D is not the best tool for modeling of flume size experimental setups. In order to most accurately represent the

Chapter 1. Introduction

flume and make use of available high-performance computing clusters, an OpenFOAM model was created in place of Delft3D for the work herein. As Delft3D solves the nearly identical equations for its environmental modeling the parameters will, in theory, coincide with those found by a parameter tuning in OpenFOAM.

Chapter 2

Problem Background

The magnitude of natural flow disruption due to CEC devices is dependent on local site conditions and device/array characteristics and is an issue that must be addressed and resolved prior to obtaining the necessary permits to deploy and operate in US waters. Unfortunately, little is known about the significance of physical effects of individual or multiple CEC devices or of the cumulative impacts of these devices on aquatic ecosystems over days, years, or decades of operation. Furthermore, tools and assessment techniques to evaluate CEC-derived effects on the physical environment are in the nascent stage of development and application. This lack of knowledge affects the ability of developers and other stakeholders to accurately assess CEC-driven environmental effects and hence affects regulatory-agency actions, opinions of stakeholder groups, and commitments of energy-project developers and investors. Therefore, the ultimate goal of this project seeks is to develop, validate, and enable industry use of 'CEC friendly' hydrodynamic modeling tools to ensure a balance between CEC array power performance and environmental effects.

CECs produce power by extracting kinetic energy from fluid flow and subsequently converting the extracted energy into electricity. CECs have advantages when compared to similar renewables for their ability to generate greater power with smaller turbines, and being visually/ audibly less intrusive to humans [15]. Unique challenges to CEC

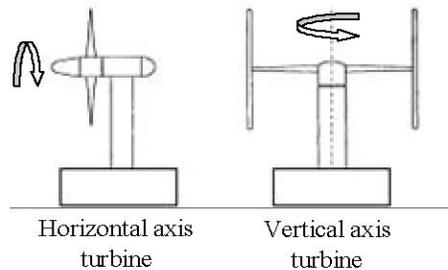


Figure 2.1: Basic turbine types [15]

power production include higher installation cost and maintenance cost associated with the submerged marine turbines, corrosion due to salinity, higher axial stresses, and possibly altering the marine habitat [3, 44].

A number of competing CEC device types exist with basic turbine design types mirroring that of the wind energy industry. The two most common device types are horizontal axis water turbines (HAWTs), and vertical axis water turbines (VAWTs) displayed in Figure 2.1. VAWTs are advantageous for design simplicity, generator coupling, lower noise emissions and performance in a skewed flow. HAWTs are advantageous for the knowledge-base, performance, and control schemes [29]. The research herein focused on HAWT-type devices.

HAWT performance is a function of the deployment site, device properties, and environmental considerations. As displayed in Figure 2.2, a HAWT is installed in a marine environment with current flow normal to the blades. The flow profile approaching the turbine is a function of the wave amplitude and phase, wind direction, velocity profile, and turbulence. These environmental properties determine the numerical quantities of interest (QoI), velocity and turbulence intensity, impinging the CEC rotor.

The act of removing kinetic energy from the flow with a CEC results in the fluid separating from the blades, spinning off and creating a region of rotational flow wider than projected area created by the CEC blades [30]. This separated flow region

Chapter 2. Problem Background

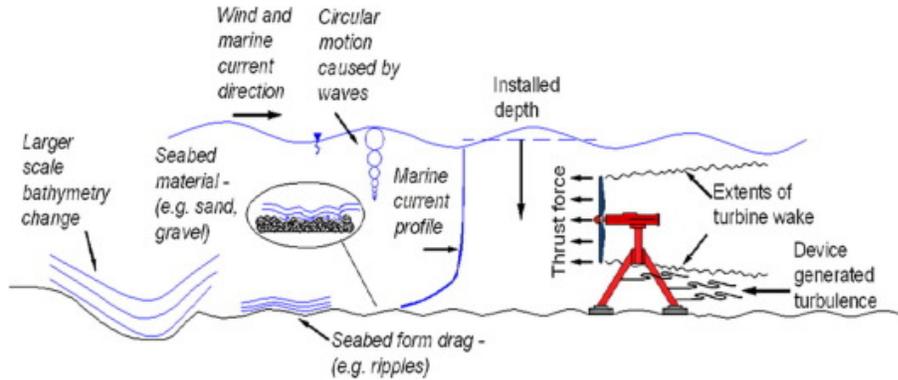


Figure 2.2: Environmental influences and turbine wake [4]

is known as the wake, and accurate computational modeling of the turbine wake is the crux to predicting both turbine-environment interaction and turbine-array interaction. It has been shown that the influent velocity is of secondary importance to turbulent intensity in terms of wake modeling, [38]. Accurate wake prediction pertains to accurate tidal farm power production and array impact on the aquatic environment.

Developing simulation tools for wave and tidal devices has been identified as a top priority for bringing marine renewable energy to market [5, 61]. Data and computational needs have motivated laboratory-scale experiments, which help reduce capital expenditures associated with sea deployments [33, 38, 39, 40, 42, 59]. Data from laboratory flume experiments have been used to calibrate numerical models including blade and blade-design models [36, 32], electrical-performance models [16], blade element method (BEM) models [34], actuator-disc models [21], and turbulence models [23]. However Gaurier et al. [19] questions experimental standards and design while Neary et al. [41] raise concerns about the use of common measurement tools in laboratory and field experiments both of which complicate comparisons between data sets and model calibration efforts. Here, one experimental data set collected at two influent turbulent intensities (I_∞) is used for model calibration [38].

2.1 Problem

The MHK industry is in need of a modeling tool capable of solving both near and far field effects in relatively short simulation times. Furthermore, this model needs to be able to simulate environmental factors, such as waves, wind, solute, and sediment transport. As MHK arrays may occupy tens to hundreds of square kilometers, supercomputers are required for high-spatial-resolution, spinning-turbine simulations [9, 10, 26]. However, reduced-order models (e. g., representing turbines by actuator discs and solving for flow fields using the shallow-water equations) are typically implemented to facilitate timely decision making. Turbines can be analytically and numerically approximated with actuator discs, which although they do not rotate, they remove momentum from the system by converting it into turbulent kinetic energy [6].

Sandia National Laboratories (SNL) implemented a turbine module with two components into an existing open source hydrodynamic software Delft3D, enabling CEC modeling. The first component implements HAWTs or VAWTs using an actuator disc model. Borrowing upon research from the wind-energy community [52, 51], k - ϵ turbulence-source terms originally developed for wind transport through vegetative canopies [27, 46] have been incorporated into the environmental flow codes SNL-Delft3D-CEC [58] and SNL-EFDC [23]. This second component increases turbine wake control by providing the user a parameterized turbulence model; adjusting turbulence production or dissipation at the turbine location. The implementation of this module is known as SNL-Delft3D-CEC [58]. However, these turbulence source parameters require tuning to experimental data.

2.2 Research Question

This research seeks to compare two methods of turbulence parameter tuning. The first method is adapted from the turbulence source tuning performed by James et.

Chapter 2. Problem Background

al. [23]. This method wraps a model-independent optimization software around the developed computational fluid dynamics (CFD) model. While effective this procedure can be quite expensive in terms of computation and time. There the second method under consideration was a machine learning technique where the CFD model was pre-sampled and a meta-model was created. For a given set of turbulence source parameters, the meta-model was capable of predicting wake behavior behind the turbine at the experimentally measured values, and only at those values, saving computation time from solving the entire flume domain as performed by CFD. The less computationally expensive meta-model was capable of being optimized in an identical manner by the model independent optimization software providing turbulence parameter values that minimized predicted and experimental wake error.

Chapter 3

Governing Theory & Equations

Computational Fluid Dynamics (CFD) can be thought of as a branch between fluid experiments and fluid theory [2]. CFD provides numerical approximations to partial differential equations using discretized algebraic equations [20]. The qualitative fluid flow solution to the conservation equations is determined by the domain geometry, fluid properties, initial conditions (IC) and boundary conditions (BC) [30]. CFD results must be scrutinized to ensure that the results provided can be trusted. Sources of error in CFD may be due to discretization, data input, IC, BC, or modeling assumptions. This section will cover the Reynolds Averaged Navier-Stokes (RANS), k - ϵ , and momentum theory equations used in this research to model a hydro-kinetic turbine flume experiment using CFD.

3.1 Reynolds Average Navier-Stokes

The modeling of a fluid was started here by considering the continuity equation which expresses conservation of mass in differential form using index notation:

$$\partial_t \rho + \partial_i \rho u_i = 0 \tag{3.1}$$

Here ∂_t is the operator $\partial/\partial t$, ρ is the fluid density, ∂_i is the partial derivative operator

Chapter 3. Governing Theory & Equations

in each spatial dimension, and u_i is the fluid velocity. Equation 3.1 uses Einstein or index notation [30] where like index sum such that $\partial_i u_i$ is the analog to the divergence operator $\nabla \cdot \mathbf{u}$ in vector notation. Assuming that a fluid particle has a zero rate of density change the fluid can be considered incompressible and Equation 3.1 becomes:

$$\partial_i u_i = 0 \quad (3.2)$$

With the continuity equation in hand, one may then turn to Newton's second law for the conservation of momentum which states that a bodies' net force must be equal to its mass multiplied by its acceleration. Here considering an infinitesimal element on a per unit volume basis the sum of body forces plus surface forces are equal to mass multiplied by acceleration

$$\rho g_i + \partial_j \tau_{ij} = \rho \frac{Du_i}{Dt} \quad (3.3)$$

where g_i is gravity, τ_{ij} is the strain rate tensor, and D/Dt in the material derivative. Modeling fluid friction requires a constitutive equation of stress-deformation. Considering an incompressible isotropic fluid the stress tensor can be described as:

$$\tau_{ij} = -p\delta_{ij} + 2\mu S_{ij} \quad (3.4)$$

Where τ is the strain rate tensor, p is pressure, δ_{ij} is the Kronecker delta, μ is the viscosity, and S_{ij} is the strain rate tensor defined as:

$$S_{ij} = \frac{1}{2} (\partial_j u_i + \partial_i u_j) \quad (3.5)$$

Now substituting the constitutive equation into Cauchy's equation of motion (Equation 3.6) and applying the fact that velocity is a solenoidal field ($\partial u_i / \partial x_i = 0$)

), we then obtain the Navier-Stokes Equation (Equation 3.7)

$$\rho \frac{Du_j}{Dt} = \rho g_j + \frac{\partial}{\partial x_i} (\tau_{ij}) \quad (3.6)$$

$$\rho \frac{Du_j}{Dt} = \rho g_j - \frac{\partial p}{\partial x_i} + \mu \frac{\partial^2 u_j}{\partial x_i \partial x_i} \quad (3.7)$$

The Navier-Stokes Equation, when combined with the conservation of mass equation, provides a complete mathematical description for the flow of incompressible Newtonian Fluids. Unfortunately, the solution of the Navier-Stokes equation becomes intractable when considering turbulence due to the complexity of the Navier-Stokes second order, nonlinear, partial differential equations mixed with the length scales and time scales of turbulent eddies. Fortunately, this level of turbulent detail is rarely needed and the Navier-Stokes equation may be solved through a process of Reynold's decomposition.

This averaging has created a new term $\overline{u_i u_j}$ which will require a new equation or set of equations to describe. The equations which describe this term are known as turbulence closure models.

3.2 k - ϵ Turbulence model

At high Reynolds number Navier-Stokes prediction of turbulent flows are computationally expensive. Since the average quantities of the mean flow equations are what is typically of interest one may average the Navier-Stokes equation. This is achieved by Reynold's decomposition (Equation 3.9) which considers a mean and fluctuating component of each dependent field variable. After performing this decomposition and plugging it into the continuity and NS equations, the resulting equations may be averaged to consider only the mean quantities.

Chapter 3. Governing Theory & Equations

$$\tilde{u}_i = U_i + u_i, \quad (3.8)$$

$$\tilde{p} = P + p, \quad (3.9)$$

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (3.10)$$

$$\frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -g\delta_{i3} - \frac{1}{\rho_0} \frac{\partial P}{\partial x_j} + \frac{\partial}{\partial x_j} \left(\nu \frac{\partial U_i}{\partial U_j} - \overline{u_i u_j} \right) \quad (3.11)$$

which yields Equation 3.9, where capital letters are the averaged quantities and the primed variables are the fluctuating component.

Reynolds decomposition has now doubled the number of dependent-field variables and introduces a closure problem. To close the RANS equations for the turbulent fluxes of momentum a common approach is to use the Boussinesq approximation. This approximation is often applied to oceanic flows. The assumptions underlying this approximation are [30]:

1. Constant density, except where ρ is multiplied by g
2. Constant dynamic viscosity (μ), thermal conductivity(κ), and specific heat (c)
3. Mach number is low

These approximations make a number of assumptions and introduce a concept know as the turbulent viscosity hypothesis, which is equivalent to Fick's law and assumes the gradient based transfer of turbulent diffusion with eddy viscosity (ν_t) used analogously to molecular viscosity.

$$\overline{u'_i u'_j} = \nu_T \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} k \quad (3.12)$$

In equation 3.12, k is known as the turbulent kinetic energy (*tke*) and the introduction of this variable requires a transport model. The most common transport

Chapter 3. Governing Theory & Equations

model is a system order two turbulence model known as $k - \epsilon$. This model solves for the transport of the turbulent kinetic energy and the dissipation rate of energy (ϵ). The length scale is determined by $l_T = k^{(3/2)}/\epsilon$, velocity scale by $u_T = \epsilon^{1/2}$, and turbulent viscosity by $\nu_T = C_\mu k^2/\epsilon$. The transport equations of the $k - \epsilon$ model are given by:

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_j} \left(\frac{\nu_T}{\sigma_k} \frac{\partial k}{\partial x_j} \right) - \epsilon - \overline{u_i u_j} \frac{\partial U_i}{\partial x_j}, \quad (3.13)$$

$$\frac{D\epsilon}{Dt} = \frac{\partial}{\partial x_j} \left(\frac{\nu_T}{\sigma_\epsilon} \frac{\partial \epsilon}{\partial x_j} \right) - C_{\epsilon 1} \left(\frac{\nu_T}{u_i u_j} \frac{\partial U_i}{\partial x_j} \right) \frac{\epsilon}{k} - C_{\epsilon 2} \frac{\epsilon^2}{k} \quad (3.14)$$

Default model constants were developed to conform to experimentally measured velocity profiles, fluctuation profiles, and energy budgets for a variety of simple turbulent flows. These constants are given by:

$$C_\mu = 0.09, \quad C_{\epsilon 1} = 1.44, \quad C_{\epsilon 2} = 1.92, \quad \sigma_k = 1, \quad \sigma_\epsilon = 1.3$$

3.3 Momentum Theory

Momentum theory was first developed for general explanations of propeller action independently by both Rankine [49] and Froude [18]. The production of thrust is explained only by the momentum changes in the fluid and says nothing about the object which caused it. This theory was idealized as an actuator disc which caused an instantaneous pressure change for the fluid passing through it. Momentum theory assumes that the disk imparts a uniform acceleration to the fluid, the flow is frictionless, and that there is an unlimited inflow of water. As the scale of the modeled system is large the time-step associated with the simulation will be large in comparison to the angular speed of the turbine blades [28]. Therefore a useful assumption can be made by treating the turbine's kinetic energy removal as a force

Chapter 3. Governing Theory & Equations

created by a porous disc. The study of the gross flow characteristics has been common practice for representing wind turbines [48]. Momentum theory assumes the fluid is ideal, steady, and incompressible. The flow is uniform through the disc and in the wake, and the disc does not impart any rotational flow. This momentum extraction is given by Equation 3.15:

$$S_{\text{mom}} = -\frac{1}{2}C_x U_d^2, \quad (3.15)$$

where S_{mom} represents a source in the momentum equation, C_x is a function coefficient defined below, and U_d is velocity at the disc. Equation (3.15) has been modified to depend U_d . Typical momentum theory depends on U_∞ but the presence of the device alters the local flow field. For turbine arrays (as the number of devices increases) U_∞ is not known for turbines in the center of the array so the device-specific thrust coefficient, C_t (-) must be amended such that velocity at the disc may be used [56]:

$$C_d = 2 \frac{1 - \sqrt{1 - C_t}}{1 + \sqrt{1 - C_t}}, \quad (3.16)$$

$$C_x = \rho A_d C_d, \quad (3.17)$$

where A_d [m²] is the flow-facing area of the actuator disc and ρ [kg/(m³)] is the water density.

Chapter 4

Framework Development

4.1 Flume Experiment

Experimental data collected by Mycek et al. [38] at the IFREMER flume in France was utilized to tune the actuator disc model. The INFREMER flume was 18m long, 4m wide, by 2m deep as shown in Figure 4.1. The three-bladed horizontal-axis turbine used in this experiment had a diameter of 0.7m (d_t) which created a blockage ratio of 4.81%. The quantities of interest were focused on wake recovery which included measurements of velocity (U) and turbulence intensity (I) collected at 10 locations behind the turbine. These centerline points as measured by normalized distance behind the turbine ($x^* = x/d_t$) were at $x^* = 1.2, 2, 3, 4, 5, 6, 7, 8, 9, 10$. Measurements were taken for turbulent inflow conditions of $I_{3\%}$ and $I_{15\%}$ both at $U_\infty = 0.8$ m/s. A schematic of the experimental setup is shown in Figure 4.2. Mycek measured the QoIs at 26 locations on either side of each centerline point, however that data was not used here as it is known that 1) tuning to centerline data provides a good fit for lateral data and 2) the focus of this research was to determine if a surrogate model could sufficiently represent the OpenFOAM simulation. Moreover, the purpose of this was not to determine the global minima.

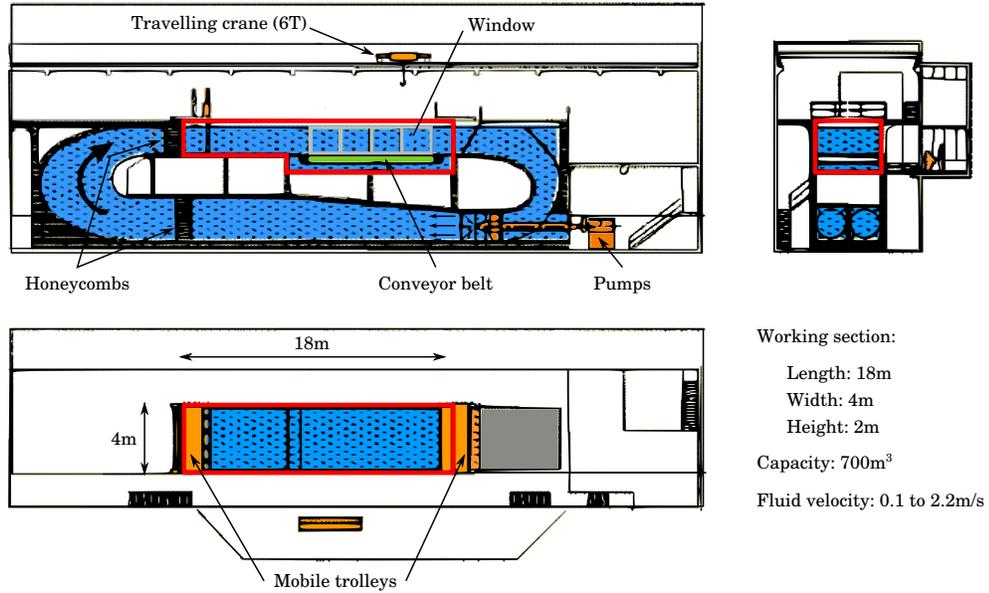


Figure 4.1: INFERMER Experimental Flume Tank [38]

4.2 OpenFOAM Model Development

The IFREMER flume tank used in the Mycek experiment was modeled in the research using open-source OpenFOAM v5.0. The CFD model here used the InterFOAM solver which can model two-phase flow using the volume of fluid (VOF) method. Two-phase flow was chosen here such that a free surface could be represented and future research could best match environmental codes which model free surfaces using the shallow water equations [62]. The InteFOAM solver was used with RANS and the $k-\epsilon$ turbulence model [31]. Only half of the IFREMER flume tank working section was modeled utilizing the longitudinal symmetry plane for computational efficiency gains as shown in Figure 4.4. This computational domain was, therefore, a cuboid of dimensions 18m long \times 2m wide \times 2.5m deep. The extra half a meter in the depth direction was used to model air on top of the flume as shown in Figure 4.4. As the symmetry plane ran down the longitudinal axis of the flume it bisected the

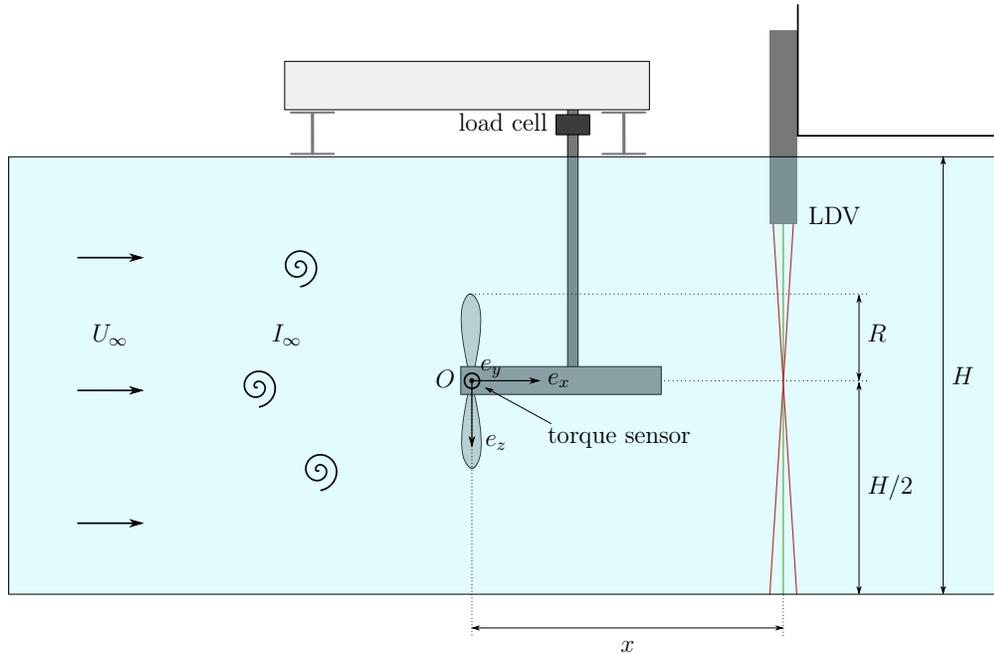


Figure 4.2: Experimental Representation [38]

0.1m thick actuator disc located $x = 6$ m from the inlet.

The cuboid domain was set up with six specific boundaries. In OpenFOAM boundaries are typically split up into a set of patches which may include one or more enclosed areas of the boundary surface. The inlet was broken up into two patches called *In* and *aboveIn*. This allowed the model to send in water into one patch and air into the other. Similarly the outlet was divided into *Out* and *aboveOut*. The top of the domain was set as a patch named *atmosphere*. The floor of the flume, upper and lower symmetry plane and backside of the flume were set as walls. Setting the symmetry plane type patch on the chosen symmetry-plane showed a nominal difference from the wall modeling which will be described.

The boundary values for the native variables are organized in Table 4.1. The variable ν_t is calculated for all boundaries by $\nu_t = C \cdot k^2 / \epsilon$. The boundary values of the sides are set as a slip for all values except α which is set as zero gradient. The

Chapter 4. Framework Development

fluid is driven down the flume by a 0.8 m/s velocity at the inlet and zero pressure at the outlet. The outlet velocity condition applies an *inletOutlet* condition which is a variable boundary condition. If the fluid is going out of the domain the boundary is a zero gradient. However, if flow reverses into the domain it will enter with a velocity of 0 m/s. At the inlet, a zero gradient is continuously applied for the pressure value.

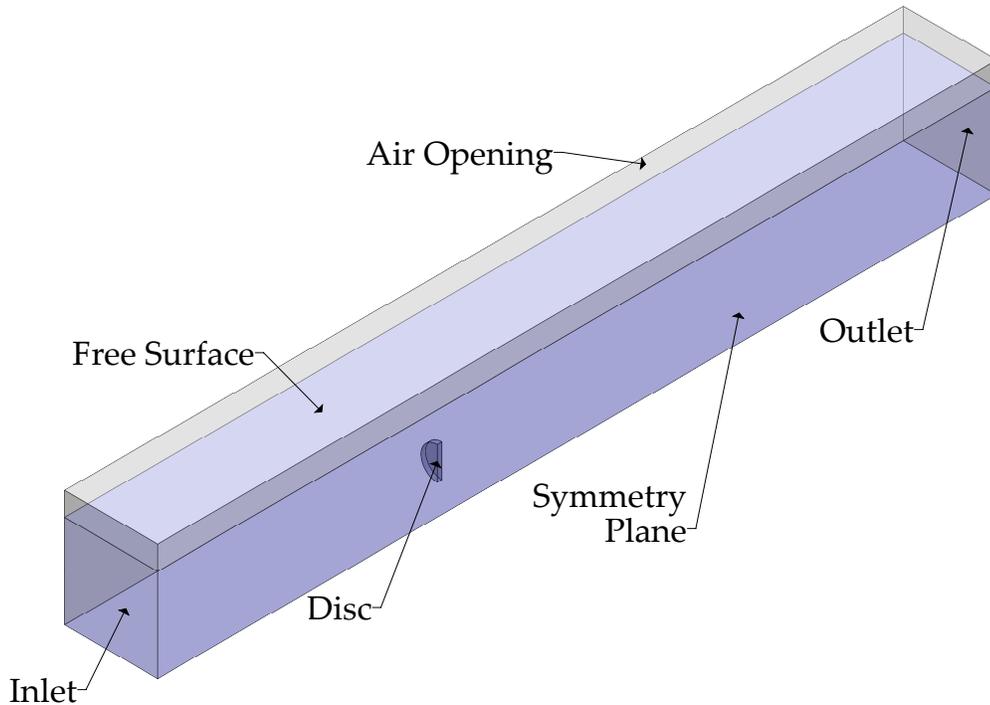


Figure 4.3: Isometric view of model domain

The boundary value for α is specified as a fixed value of 1 (water) at the *In* and 0 (air) for the *aboveIn*. At the exit of the flume, an *inletOutlet* condition is applied such that if flow reverses back into the domain then only water comes in from zero to two meters and air comes in from 2 meters to 2.5 meters. The boundary condition allows the water level to vary at the exit.

Inlet flow was specified uniformly as $U_\infty = 0.8$ m/s. The influent turbulence values were specified as for k and ϵ as $k = 3/2 (I_\infty U_\infty)^2$ and $\epsilon = c_\mu * k^{3/2} / d_t$. At

Table 4.1: Boundary condition applied for each variable and patch

Variable	In/ AboveIn	Out/ AboveOut	Atmosphere
α	fixedValue 1/ 0	inletOutlet In 1/ 0	inletOutlet In 0
ϵ	fixedValue	zeroGradient	zeroGradient
k	fixedValue	zeroGradient	zeroGradient
$P-\rho gh$	zeroGradient	fixedValue 0	zeroGradient
U	fixedValue 0.8	inletOutlet In 0	zeroGradient

the outlet, the pressure was maintained 2 m above the flume floor with 0.5 m of air above the water. As vertical velocity profiles were not measured and boundary-layer theory for flow over a flat plate suggests that the boundary layer was only ~ 0.1 m tall, free-slip conditions were applied along the flume floor and walls. The elevation of the free surface was internally calculated by the VOF technique but was constrained by the specified-pressure outlet boundary condition. The OpenFOAM model was run using mixed-order solution techniques including 1st order for k and ϵ (upwind) and 2nd order for U divergence (linear upwind). The observed orders of convergence for U and k were to be 2.08 and 1.94 respectively for the $I_\infty = 3\%$ case at $x^* = 5$.

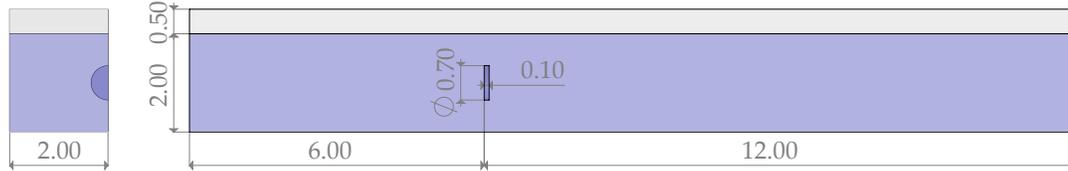


Figure 4.4: Model domain dimensions [m] for the front (left) and side (right) view along the symmetry plane.

4.3 OpenFOAM Grid Study

The focus of this work was to determine if the surrogate model may be utilized for optimization in place of the full simulation. Thereby a consistent trend shown by the quantities of interest will suffice for a fair comparison between the simulation

Chapter 4. Framework Development

and surrogate model. However, a grid convergence study was performed to quantify the solution order of convergence to provide a magnitude of discretization errors and provide confidence in the simulated results. One popular technique is Richardson Extrapolation [53, 54]. Here a sixth ordered extrapolation is taken using three uniform grid refinements. A discrete solution f is assumed to have a series representation in the grid spacing h and multiplied by a continuous function g which is independent of discretization. The extrapolated solution here is given by Equations 4.1, and 4.2:

$$f_{extrap} = f_2 + (f_2 - f_3) \cdot \left[\frac{1}{r} \right]^{p-1} \quad (4.1)$$

$$p = \frac{\ln \left[\frac{f_3 - f_2}{f_2 - f_1} \right]}{\ln(r)} \quad (4.2)$$

Equation 4.1 states that the extrapolated solution is given by the most refined grid QoI (f_3), the second most refined grid QoI (f_2), the refinement rate ($r = 1/2$) and the convergence rate (p). The rate of convergence is given by Equation 4.2 and can be found by taking the natural log of the ratio of the differences of the most refined to second most refined and the second most refined and least refined grid and dividing this quantity by the natural log of the refinement rate.

Equations 4.1 and 4.2 can be applied for each QoI and any point in the grid domain. Here the QoIs were determined by the experimental measurements to be velocity (U) and turbulence intensity (I). As the turbulence closure model is k - ϵ the quantity I is a function of turbulent kinetic energy (k) and velocity where $I = \sqrt{2/3 \cdot k} / U$. To remove interdependence of the U on the I the Richardson extrapolation used k for analysis. Equations 4.1 and 4.2 were applied at each of the 10 points behind the turbine where the QoIs were measured.

The OpenFOAM flume model representing IFREMER was discretized using OpenFOAM's structured meshing tool blockMesh [60]. Uniform mesh refinement ($r=1/2$) was performed on an initial $h_0 = 0.1$ m grid discretization twice resulting

in uniform 0.05 m and 0.025 m discretization for the resulting grids. The grid convergence results are shown in Figure 4.5.

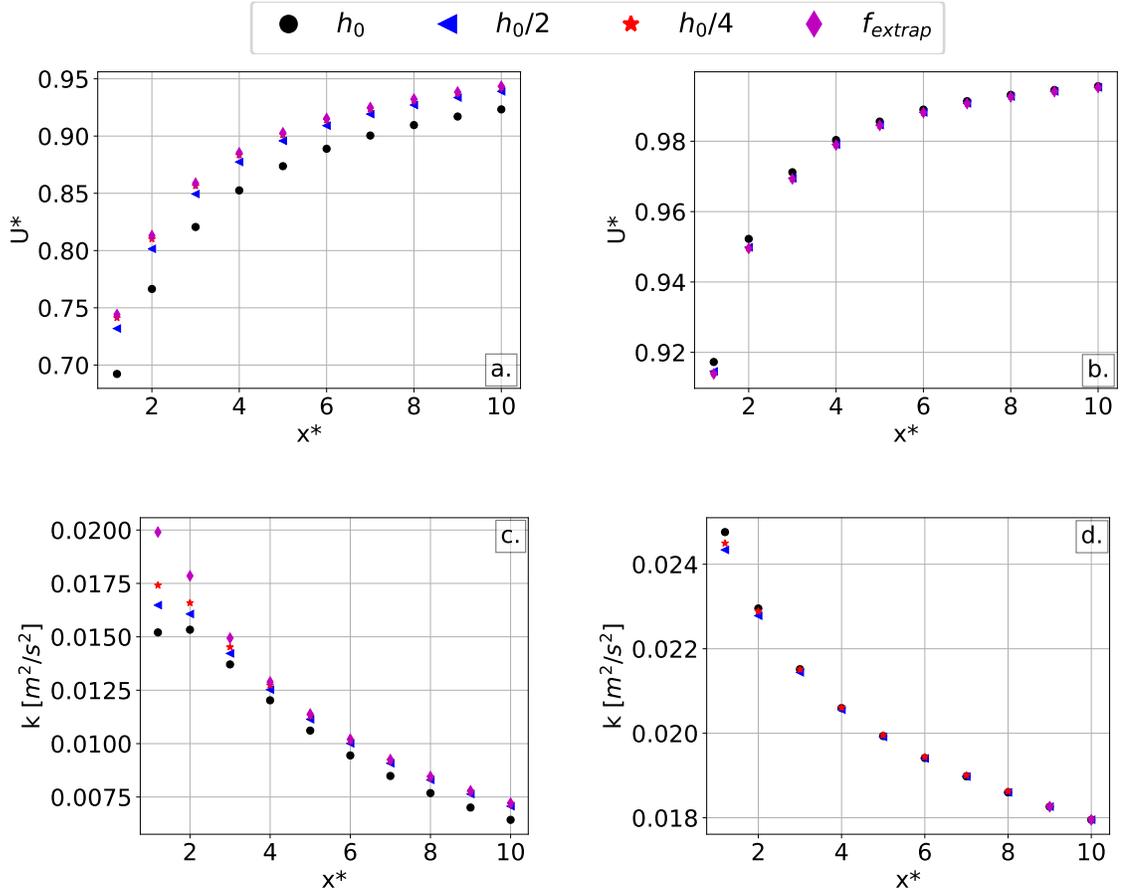


Figure 4.5: Uniform mesh refinement grid study results starting at $h_0 = 0.1m$ plotted at the spatial points of interest for each quantity of interest namely (a.) U^* at $TI_\infty = 3\%$, (b.) U^* at $TI_\infty = 15\%$, (c.) k at $TI_\infty = 3\%$, and (d.) k at $TI_\infty = 15\%$

Overall the grid refinement shows nominal differences to the solution which ensures that any conclusion drawn about the possibility of using a surrogate model in place of the OpenFOAM model will be sufficient. The results for the $I_\infty = 3\%$ (Figures 4.5.a and 4.5.c) case show some convergence as the grid is refined. For $U_{I_\infty=3\%}$ the most refined grid ($h_0/4$) shows nominal difference from the f_{extrap} solution obtained

using Equations 4.1 and 4.2. For $U_{I_\infty=3\%}$ and $k_{I_\infty=3\%}$ the three diameters nearest the turbine ($x^* = 1.2, 2, 3$) there is a non-negligible difference between the h_0 and f_{extrap} solutions. Solution wall clock times are intractable at $h_0/2$ and exceedingly so at $h_0/4$. As the solution follows the correct trend for all points excluding $k_{I_\infty=3\%}$ at $x^* = 1$ this grid was considered sufficient to represent the OpenFOAM model.

The results for the $I_\infty = 15\%$ (Figures 4.5.b and 4.5.d) case show negligible difference and the correct trend for all points. Close examination of k for the $I_\infty = 15\%$ case at $x^* = 1.2$ shows that the solution did not monotonically decrease. A Richardson extrapolation cannot be performed for a non-monotonically decreasing solution as Equation 4.1 must take the logarithm of a negative number. As the change in solution is small it is difficult to discern from Figure 4.5.d but the non-monotonic decrease occurs for all $x^* \leq 8$ and therefore an f_{extrap} solution can not be calculated. Here, k at $x^* = 9, 10$ had a monotonically decreasing solution and the f_{extrap} appears on top of all grid solution values.

4.4 Turbulence-source Equations

Originally developed to more accurately model flow through vegetative canopies [27, 46] the k - ϵ turbulence-equation source terms have since been applied to wind turbines [52, 51] and MHK models [23]. These source terms were added to OpenFOAM and applied at the actuator disc location. Sources of turbulent kinetic energy, k (m^2/s^2), and its dissipation rate, ϵ (m^2/m^3), are:

$$S_k = \frac{1}{2}C_x (\beta_p U^3 - \beta_d U k), \quad (4.3)$$

$$S_\epsilon = \frac{1}{2}C_x \left(\beta_p C_{\epsilon 4} \frac{\epsilon}{k} U^3 - \beta_d C_{\epsilon 5} U \epsilon \right), \quad (4.4)$$

where $C_x (-)$ is the modified thrust coefficient as defined in section 3.3. U (m/s) is the velocity at the local velocity or velocity at the turbine/actuator disc. Adjustable

parameter β_p (-) is the fraction of k converted to wake-generated kinetic energy by drag. Adjustable parameter β_d (-) accounts for k dissipation through conversion to kinetic energy by the turbine blades (or actuator disc), or the short-circuiting of the turbulence cascade where energy transfers from large-scale turbulence to smaller scales. Adjustable parameters $C_{\epsilon 4}$ and $C_{\epsilon 5}$ are closure constants [63]. For vegetative canopies, Katul et al. [27] suggested $\beta_p = 1$, $\beta_d = 5.1$, $C_{\epsilon 4} = 0.9$, and $C_{\epsilon 5} = 0.9$, for wind turbines Réthoré et al. [51] suggested $\beta_p = 0.05$, $\beta_d = 1.6$, $C_{\epsilon 4} = 1.5$, and $C_{\epsilon 5} = 0.2$, and for actuator discs representing current energy converters (CECs) James et al. [23] suggested $\beta_p = 0.96$, $\beta_d = 1.38$, and $\beta_p C_{\epsilon 4} = \beta_d C_{\epsilon 5} = 3.87$.

4.5 Metrics

Table 4.2 defines three metrics applied here to compare the QoIs (Q) between the experimental measurements (exp), OpenFOAM simulations, (sim), and surrogate-model estimates (srgt). As described in section 4 the two QoIs (U , I), were sampled at 10 points in the wake, at two I_∞ (3%, 15%). Therefore there were a total of $N = 40$ experimental values for model comparison, and $N = 10$ for any QoI at a given I_∞ .

Table 4.2: Performance metrics.

Comparisons	Mean absolute percentage error (MAPE)	Mean squared error (MSE)	Maximum error ($ L_\infty $)
sim/exp	$\frac{100}{N} \sum_{i=1}^N \left \frac{Q_{\text{sim}} - Q_{\text{exp}}}{Q_{\text{exp}}} \right _i$	$\frac{1}{N} \sum_{i=1}^N \left(\frac{Q_{\text{sim}} - Q_{\text{exp}}}{Q_{\text{exp}}} \right)_i^2$	$100 \cdot \max \left \frac{Q_{\text{sim}} - Q_{\text{exp}}}{Q_{\text{exp}}} \right _i$
srgt/sim	$\frac{100}{N} \sum_{i=1}^N \left \frac{Q_{\text{srgt}} - Q_{\text{sim}}}{Q_{\text{sim}}} \right _i$	$\frac{1}{N} \sum_{i=1}^N \left(\frac{Q_{\text{srgt}} - Q_{\text{sim}}}{Q_{\text{sim}}} \right)_i^2$	$100 \cdot \max \left \frac{Q_{\text{srgt}} - Q_{\text{sim}}}{Q_{\text{sim}}} \right _i$
srgt/exp	–	$\frac{1}{N} \sum_{i=1}^N \left(\frac{Q_{\text{srgt}} - Q_{\text{exp}}}{Q_{\text{exp}}} \right)_i^2$	–

Metrics listed in Table 4.2 are provided at each appropriate step. However, three metric definitions are used for surrogate model decision making and are provided formal definitions as follows:

- Fitting Error** Indicates the difference between surrogate model predictions and simulation prediction by $MAPE_{srgt/sim}$
- Predicted Error** The difference between surrogate model prediction and physical experiment by $MSE_{srgt/exp}$
- Final Error** At optimal turbulence source parameter values, the difference between simulation predictions and physical experiment

Fitting error determines if the developed surrogate model is of sufficient accuracy to proceed to optimization and if the surrogate model represents the CFD simulator well at the final value obtained by optimization. The fitting error compares the predictions of the surrogate model and the simulation. The predicted error is minimized during optimization and is equivalently the objective function for the surrogate model optimization. As an optimization of the full CFD and the surrogate is provided in this work, the predicted error term allows for simple reference to refer to the surrogate model optimization. Lastly, final error refers to the optimization values which minimize predicted error, run in the full CFD simulator (OpenFOAM) compared to the physical experiment. The term final error is used as it represents the ultimate goal of the surrogate model optimization. The use of these terms in relation to the overall surrogate model development is formally defined in Section 4: Surrogate Model Optimization Framework.

4.6 Standard k - ϵ Model Results

The standard k - ϵ model will serve as a reference for performance improvements and justify the improved simulated results agreement with experimental data. The standard k - ϵ model is equivalent to the turbulent source terms set to zero ($0 = \beta_p = \beta_d = C_{\epsilon 4} = C_{\epsilon 5}$). The error of the standard k - ϵ model as compared to the experiment are presented in Table 4.2. Model error was highest for low turbulence ($I_{3\%}$). The average model error by MAPE was 28.2% and near 20% for individual

QoIs excluding $U_{3\%}$ where the wake recovery was predicted to be significantly faster than the experiment.

Table 4.3: Actuator disc with standard k - ϵ model errors.

Q	MAPE _{sim/exp}	MSE _{sim/exp}	$ L_\infty _{\text{sim/exp}}$
$U_{3\%}$	50.3%	0.287	77.3%
$U_{15\%}$	21.3%	0.113	82.4%
$I_{3\%}$	23.3%	0.070	39.6%
$I_{15\%}$	18.0%	0.058	45.7%
Mean	28.2%	0.132	61.3%

4.7 PEST Calibration/Parameter Estimation

Model error to experiment was improved through the calibration of the turbulent source terms. Calibration was performed using the model-independent, Parameter ESTimation (PEST) software suite [12], which uses the gradient-based Gauss-Marquardt-Levenberg approach to minimize the objective function in a least-square sense. The basis of this optimization technique is formed around the partial derivatives calculation of the observation points with respect to the model input parameter. The derivatives are stored in the normalized Jacobian, then the Marquardt lambda ensures a stable approach toward the minima by slightly offsetting the gradient descent by addition to the Jacobian diagonal. A new parameter set which minimizes the objective function is calculated from iterative matrix inversion. Robust gradient-based optimization is simple to implement but may suffer from becoming trapped in local minima. A highly nonlinear objective-function response surface can be problematic. The PEST optimization method was primarily chosen for comparison to previous work of James et. al. [23]. Once demonstrated, future research will include non-gradient decent based optimization methods such as particle swarm [47] which has shown promising and repeatable results based on the author’s preliminary work.

Chapter 4. Framework Development

PEST was supplied with four adjustable parameters ($\beta_p, \beta_d, C_{\epsilon 4}, C_{\epsilon 5}$) and 40 experimental observations (10 measurements downstream of the turbine for each QoI) to formulate the objective function as:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N w_i \left(\frac{Q_{\text{predicted}} - Q_{\text{exp}}}{Q_{\text{exp}}} \right)_i^2, \quad (4.5)$$

where i is a measured observation, N is the total number of observations, w_i is the weight assigned to each observation, and $Q_{\text{predicted}}$ and Q_{exp} are the OpenFOAM or surrogate model equivalents to the experimental QoIs (turbulence intensity and velocity along the centerline of the wake), respectively. Unit weights were applied to all observations. Moreover, U and TI were normalized to each range from 0 to 1.

Derivatives were incremented using a relative scheme, multiplying the current absolute value of the parameter by the calculated increment value. The initial increment was set to 5E-2 with a lower bound of 1E-5. The derivatives were set to switch between forward and central, using forward initially and switching to second order central differences when the relative objective function reduction is below 0.1. Since the parameter groups are set to a relative max then the change in a parameter group “ b ” would be $|b - b_0|/|b_0| \geq 10^{-5}$. No transformation of the parameters bounds was performed and the change limit of any one iteration was controlled by a factor ($f > 1$) defined for a parameter b as $b_0/f \leq b \leq fb_0$ for positive b_0 and $fb_0 \leq b \leq b_0/f$ for negative b_0 . The initial lambda value was set to ten with an adjustment factor of three. The limiting relative phi reduction between lambdas was set to 0.3 with a maximum of ten lambda trials per iteration. An objective function result less than 10^{-3} was considered convergent with four values obtained in this range. Furthermore, failure to lower the objective function for greater than ten consecutive iterations would end optimization and the total maximum number of optimization iterations was sixty. [64].

PEST was applied to both the OpenFOAM and GPR surrogate models. Parame-

Chapter 4. Framework Development

ters β_p and β_d were initially specified according to the results from James et al. [23]. Due to the use of a different turbulence model, initial guesses of 0.9 were used for C_{ϵ_4} and C_{ϵ_5} as suggested by Katul et al. [27]. All lower bounds were set to 0.1. For surrogate model construction, the sampled space must be representative, but the broader the parameter range, the more sample points required to develop the GPR. The upper bound of β_p is physically limited to 1 because this is the fraction of kinetic energy converted to k . The upper bound of β_d was set to 4. The upper bounds of C_{ϵ_4} and C_{ϵ_5} were commensurately set equal to 1 and 4, respectively. These ranges were sufficient to afford comparisons between OpenFOAM model optimizations and GPR model optimization.

Table 4.4: PEST initial guess and tunable parameter bounds

Parameter Name	Initial Value	Lower bound	Upper bound
x_1	0.90	0.1	1.0
x_2	1.38	0.1	4.0
x_3	0.90	0.1	1.0
x_4	0.90	0.1	4.0

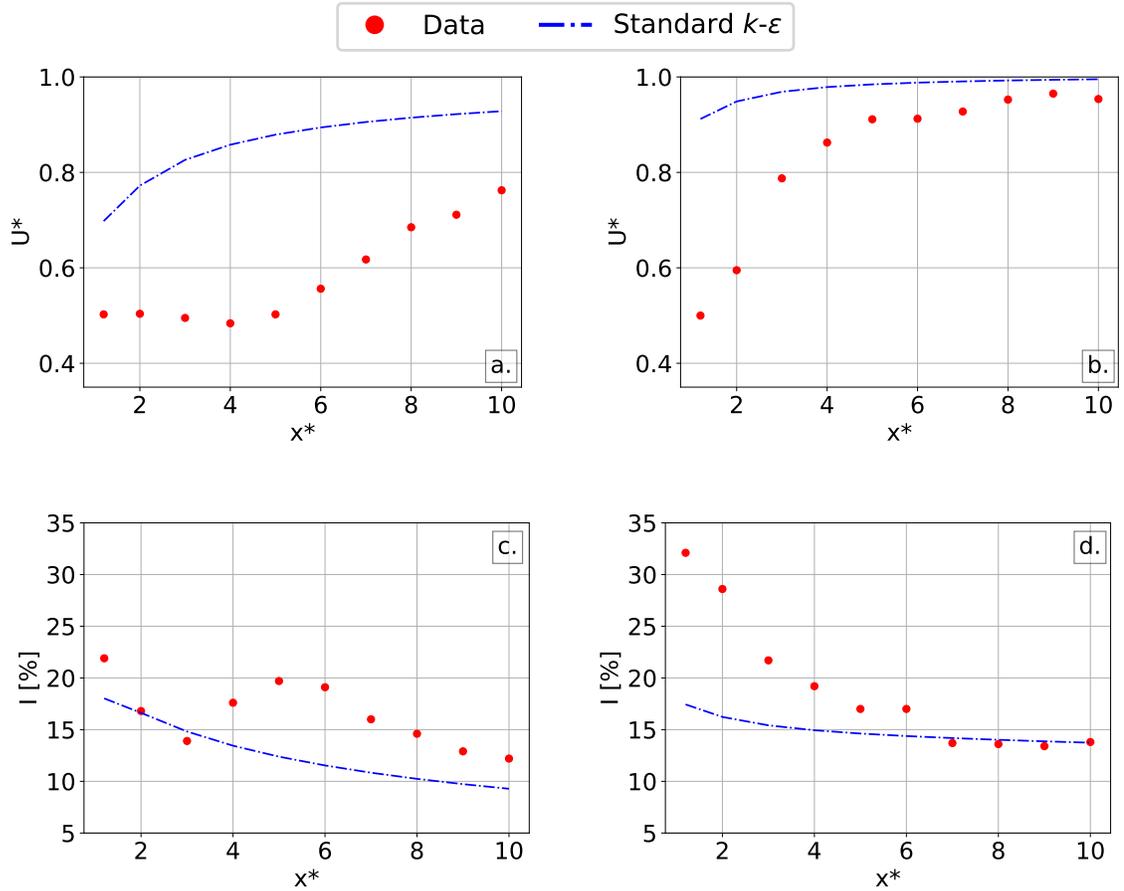


Figure 4.6: Comparison of experimental data (red dots) to the standard $k-\epsilon$ actuator disc results (blue dash-dot curve) for QoIs U^* at (a) $I_{3\%}$ and (b) $I_{15\%}$ and I at (c) $I_{3\%}$ and (d) $I_{15\%}$.

Chapter 5

Surrogate Model Development

For models with long run times exceeding 24 hours, calibration becomes intractable because in addition to the many required model calls, the calibration process itself is iterative; rarely is an acceptable calibration achieved on the first try [13]. Alternately, pre-sampling parameter space and building a surrogate model can speed up the process if the surrogate model sufficiently approximates the computationally expensive model. Nevertheless, care must be taken when constructing the surrogate model to ensure that it is a valid substitute for the full, physics-based model. A Gaussian process regression model was built using sample data from the OpenFOAM model for varying turbulence parameter values.

5.1 Gaussian Process Regression

A Gaussian process (GP) is a distribution over functions [55]. Functions may be represented by an infinitely long vector or more tractably a large number of sample points [17]. Therefore a GP is a collection of real random variables (\mathbb{R}). The function we wish to approximate $f(\mathbf{x})$ is a subset of the \mathbb{R} , any finite number of which have a

joint Gaussian distribution i.e.:

$$f(\mathbf{x}) \sim \mathcal{N}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x})) \quad (5.1)$$

where \mathbf{x} is a vector of any \mathbb{R} , $m(\mathbf{x})$ is the mean value of \mathbf{x} , $K(\mathbf{x}, \mathbf{x})$ is the covariance or measure of similarity between any two points in \mathbf{x} and $f(\mathbf{x})$ is an unknown function represented by a normal distribution (\mathcal{N}) with mean $m(\mathbf{x})$ and covariance $K(\mathbf{x}, \mathbf{x})$. A Gaussian process regression is performed by conditioning the unknown function values at \mathbf{x}_* on the known data $(\mathbf{x}, f(\mathbf{x}))$ to estimate $f(\mathbf{x}_*)$ [17, 50].

Gaussian Basics

First consider two sets of random variables shown in Figure 5.1. The results of this sampling can be seen to be non-deterministic as a given x value has multiple $f(x)$ values associated with it.

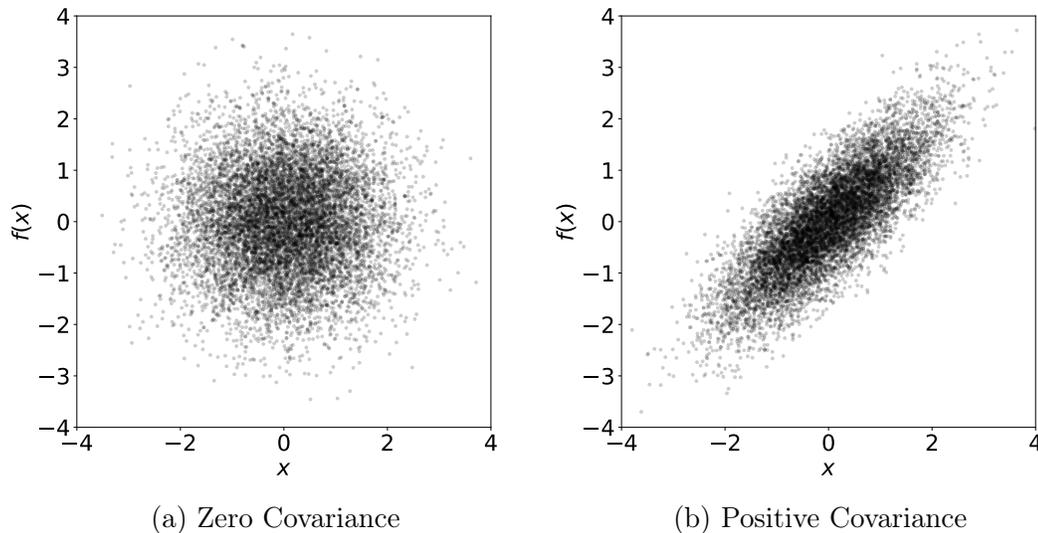


Figure 5.1: Two samples data sets

In Figures 5.1a and 5.1b the random variables \mathbf{x} and $f(\mathbf{x})$ each of length N have a zero mean. In Figure 5.1a information about x does not provide additional

information about the value $f(x)$. For instance if an $f(x)$ value was known for $x = 2$, a better guess could not then be obtained for $f(x = 1)$ as the value is equally likely to be above and below the abscissa. However, in Figure 5.1b the positive covariance helps provide additional information about the value $f(x)$ given x . For instance if given a positive x , $f(x)$ can be assumed to be positive with high probability [17].

To assume that data is Gaussian distributed is to say that any point x and $f(x)$ is simulated by a normal distribution with a mean and covariance calculated from the sample data as the sample mean and a sample variance:

$$\begin{bmatrix} x \\ f(x) \end{bmatrix} = \mathcal{N} \left(\begin{bmatrix} \mu_x \\ \mu_{f(x)} \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \right) \quad (5.2)$$

For the present example, it was assumed the mean is zero. As the mean acts as a scaling factor to the distribution, this is not a limiting assumption. This joint distribution may be visualized in three dimensions as a Gaussian curve over the two-dimensional data in Figure 5.1. The Gaussian curve over these data points (think about a 3D Gaussian bell over the green curve (3σ)) is known as the joint distribution and represents the probability of x and $f(x)$ [$p(x, f(x))$]. Figure 5.2 shows the data with the green 3σ contour shown around the data on the $(x, f(x))$ plane. The blue curve shows the projection of the joint Gaussian and represents the probability of x ($[p(x)]$), and the red curve similarly represents the probability of $f(x)$ [$p(f(x))$]. These projections are known as the conditional distribution and allow a probabilistic computation at a given point [17].

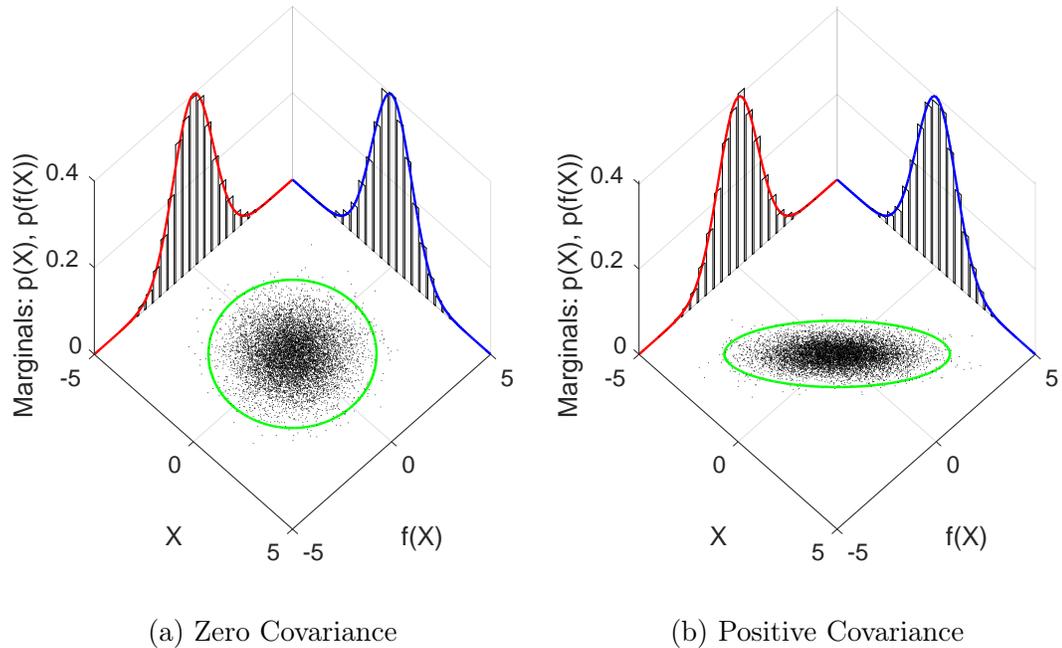


Figure 5.2: Shows two random variable examples

The conditional distribution may be used to obtain a probability distribution for $f(x)$ given x [$p(f(x)|x)$]. This conditional distribution allows one to predict a mean $f(x)$ value and a standard deviation using Theorem 5.1 known as the Schur complement or matrix inversion lemma [7].

Theorem 5.1. *Suppose $x = (x_1, x_2)$ is jointly Gaussian with parameters*

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

Then the marginals are given by

$$p(x_1) = \mathcal{N}(x_1|\mu_1, \Sigma_{11})$$

$$p(x_2) = \mathcal{N}(x_2|\mu_2, \Sigma_{22})$$

Chapter 5. Surrogate Model Development

and the posterior conditional is given by

$$\begin{aligned} p(x_1|x_2) &= \mathcal{N}(x_1|\mu_{1|2}, \Sigma_{1|2}) \\ \mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2) \\ \Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \end{aligned}$$

Now that a joint Gaussian has been defined by the sample, and conditional distributions may be obtained using the Theorem 5.1 a sampling process may be used to create a Gaussian prior [55]. The Gaussian prior will be a sample of functions calculated as a random normal with mean zero and covariance one ($\mathcal{N}(0, 1)$). In order to form the Gaussian posterior, the prior distribution will be conditioned with the sample data. Before conditioning, the prior knowledge (smoothness assumption) is applied using a formula known as the kernel. The most basic kernel is the squared exponential kernel given in Equation 5.3:

$$K_{ij} = e^{-\|x_i - x_j\|^2} = \begin{cases} 0 & \|x_i - x_j\| \rightarrow \infty \\ 1 & x_i = x_j \end{cases} \quad (5.3)$$

The squared exponential kernel given in Equation 5.3 has the properties that as x_i and x_j become further apart the kernel becomes zero and if $x_i = x_j$ the kernel is unity. These properties lead to a semi-positive definite matrix and allow the system of equations to be solved using Cholesky decomposition [1, 25, 57]. This kernel enforces a prior assumption of smoothness where small variations in the x variables produce small variations in the resulting $f(x)$. Now that there is a formal definition of how similar any two points should be, and therefore predictions at new points may be made by modifications to the kernel matrix [7].

Given a data set (\mathcal{D}) with two vectors, \mathbf{x} and \mathbf{f} , of length N where $\mathbf{f} = f(\mathbf{x})$ and $f(\cdot)$ is some unknown function; one may wish to predict the functional outputs \mathbf{f}_* given \mathbf{x}_* where $\mathbf{f}_* = f(\mathbf{x}_*)$ using Gaussian process regression. This is accomplished

by appending the prediction points to the multivariate Gaussian distribution as shown in Equation 5.4:

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} = \mathcal{N} \left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_* \\ \mathbf{K}_*^T & \mathbf{K}_{**} \end{bmatrix} \right) \quad (5.4)$$

where using Equation 5.3 \mathbf{K} is $K(\mathbf{x}, \mathbf{x})$, \mathbf{K}_* is $K(\mathbf{x}, \mathbf{x}_*)$, and \mathbf{K}_{**} is $K(\mathbf{x}_*, \mathbf{x}_*)$ with dimensions $N \times N$, $N \times N_*$, and $N_* \times N_*$ respectively. Also, \mathbf{f} and \mathbf{f}_* are a joint vector which comes from this multivariate Gaussian. Using Theorem 5.1 we may find the conditional as:

$$p(\mathbf{f}_* | \mathbf{x}_*, \mathbf{x}, \mathbf{f}) = \mathcal{N}(\mathbf{f}_* | \mathbf{m}_*, \boldsymbol{\Sigma}_*) \quad (5.5)$$

$$\boldsymbol{\mu}_* = \mu(\mathbf{x}_*) + \mathbf{K}_*^T \mathbf{K}^{-1} (\mathbf{f} - \mu(\mathbf{x})) \quad (5.6)$$

$$\boldsymbol{\Sigma}_* = \mathbf{K}_{**} - \mathbf{K}_*^T \mathbf{K}^{-1} \mathbf{K}_* \quad (5.7)$$

In summary, a Gaussian process is a subset of a stochastic process where an n dimensional real input vector \mathbf{x} and any finite number of determined random variables are specified to have a joint Gaussian distribution. A Gaussian process defines a prior distribution over functions which when supplied data may be used to define a posterior distribution over functions [55]. Here the distribution over functions is accomplished by using the one-to-one mapping between input \mathbf{x} and output $f(\mathbf{x})$ to describe the functional output as normally distributed and this may be defined as a Gaussian process i.e. $f(\mathbf{x}) \sim \mathcal{N}(\mu, \sigma^2 I) \Rightarrow f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ (See Rasmussen et al. [50]). The Gaussian process is fully specified by the mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ where \mathbf{x}, \mathbf{x}' is any point in the input domain \mathbf{x} . The covariance function or kernel may take many forms such that the kernel is semi-positive definite and has the property that as $x - x' \rightarrow \infty$, $k(x, x') \rightarrow 0$; as the distance between any two points increases, the covariance between those two points

tends to zero. GPR is considered local because predictions are weighted averages of measurements in the neighborhood of the point where the system is estimated [45].

Example in Two-Dimensions

An example Gaussian process is demonstrated in two dimensions for the data shown in Figure 5.3. The given plot shows a "True" unknown function, $f(x) = \sin(0.9x)$, in blue. Ten random sample points have been measured shown with black "+" markers and makes up the given data set (\mathcal{D}) with two vectors, \mathbf{x} and \mathbf{f} , of length $N = 10$. The goal was to use Gaussian process regression to predict the function values over the x range negative five to five ($\mathbf{x}_* \in [-5, 5]$). Here the prediction points (\mathbf{x}_*) were chosen using the linear space function in python (`linspace(start, stop, number of points)`) specified as $\mathbf{x}_* = \text{linspace}(-5, 5, 50)$. These \mathbf{x}_* are the points shown by the blue dots in Figure 5.3 for which the \mathbf{x}_* are chosen and the $f(\mathbf{x}_*)$ values are unknown.

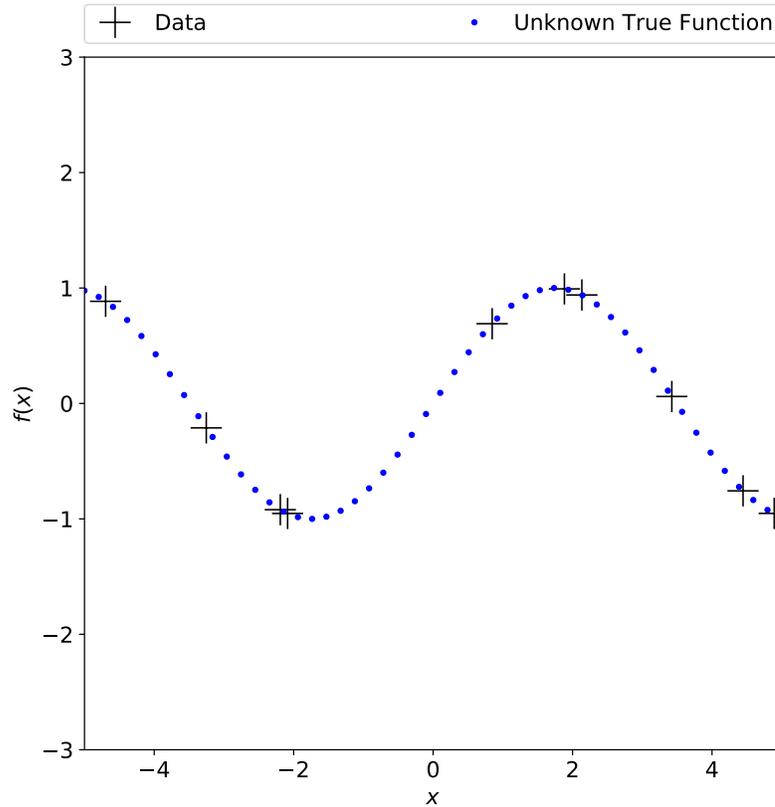


Figure 5.3: Example GPR will use data set (\mathcal{D}) consisting of points '+' to predict the unknown "True Function" locations shown shown as blue dots

To create a Gaussian prior the mean is assumed zero, and the \mathbf{x}_* data is applied to the squared exponential kernel (Equation 5.3) providing a measure of similarity between each of the \mathbf{x}_* points ($K(\mathbf{x}_*, \mathbf{x}_*)$). A Gaussian process is a distribution over functions and here 10 sample functions were chosen. Each of the prior functions were created by taking the dot product of the Cholesky decomposition of $K(\mathbf{x}_*, \mathbf{x}_*)$ and a random normal distribution sampled at \mathbf{x}_* . Repeating this process 10 times and plotting the result gives the Gaussian prior shown in Figure 5.4. Increasing the number of Gaussian process curves in Figure 5.4 would cover the mean value (0) more thickly with lines as each predicted point on the lines is $\mathcal{N}(0, 1)$ distributed.

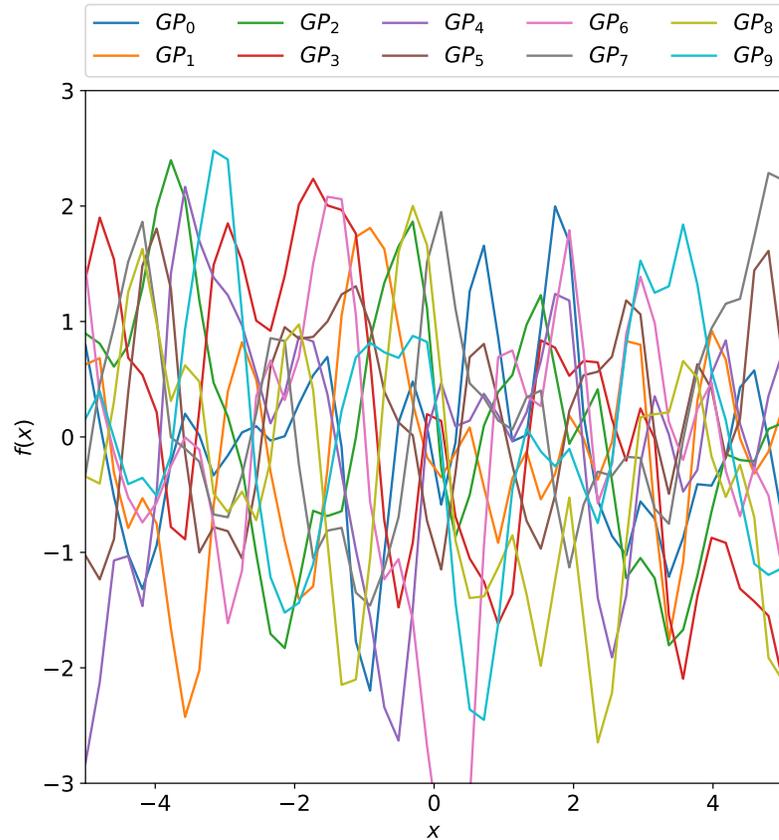


Figure 5.4: Example Gaussian Process prior evaluated at \mathbf{x}_* for 10 samples where each line represents one GP sample

Conditioning the Gaussian prior with the 10 data points (\mathcal{D}) shown in Figure 5.3 results in the Gaussian posterior shown in Figure 5.5. The posterior distribution serves to demonstrate how Gaussian process regression predicts exactly the values for the provided data set \mathcal{D} and is less certain of the functional values further away from \mathcal{D} . Finding the mean and variance of the Gaussian process posterior samples returned the GPR prediction $\boldsymbol{\mu}_*$ and the certainty of that prediction $\boldsymbol{\Sigma}_*$ as plotted in Figure 5.6.

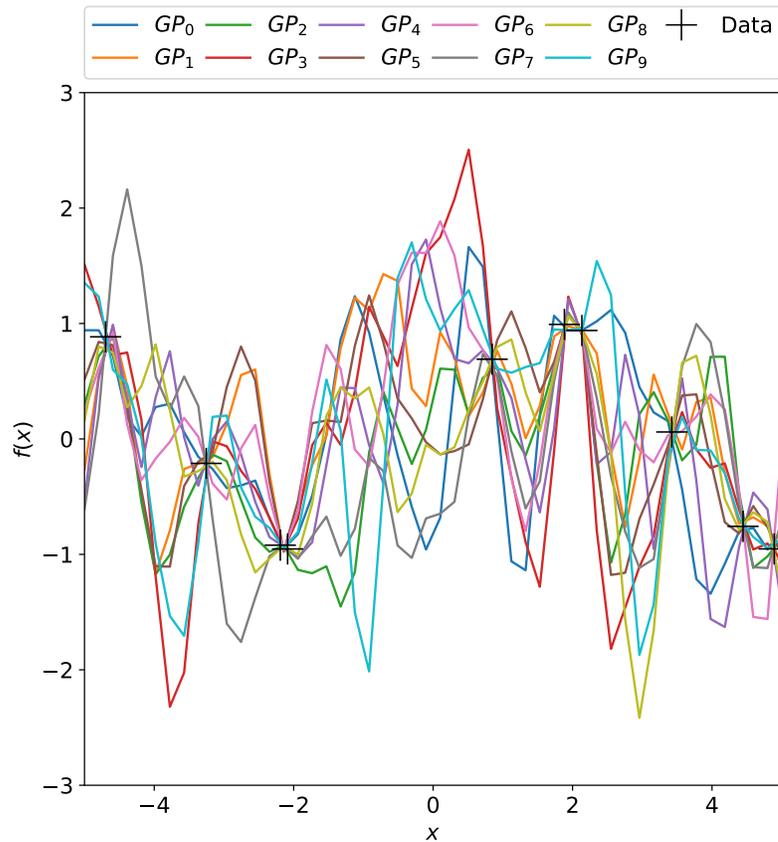


Figure 5.5: Example Gaussian posterior given data shown in Figure 5.3 for each of the 10 GP Samples

In summary Figure, 5.6 shows the mean (μ_*) labeled as "Prediction" as a dashed red line and the certainty of the predictions at three standard deviations as a filled grey area (3σ). The data set (\mathcal{D}) is plotted as red "+". Lastly, the "True" unknown function is shown as a solid blue line. While the sampling of the unknown function provided a rather poor fit this sampling demonstrates a couple of interesting points about GPR. The first is that in the areas where data points \mathcal{D} are very near to each other the certainty of the estimate is high and the mean prediction provides a nominal difference from the true unknown function. Secondly, it can be seen that in areas where the data is far apart the uncertainty is much higher. As in a real scenario the "True Function" remains unknown following the mean prediction, this uncertainty

measurement may guide future sampling measurements.

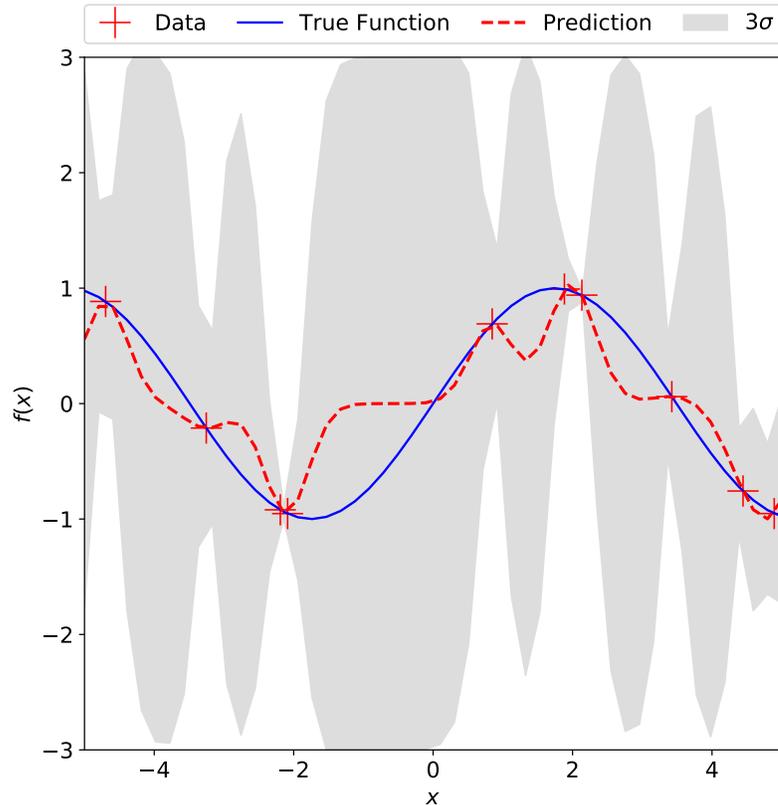


Figure 5.6: Example GPR mean and standard deviation over the GP posterior functions shown in Figure 5.5

Kernel Optimization

In the previous section, the squared exponential kernel was used to describe the similarity between the data. However, there are many kernels which may be specified to describe the similarity between data points. These kernel functions differ in functional form but must meet the same properties specified in Equation 5.3. In addition to changes in the functional form, kernels may have hyperparameters which may be optimized for a given data set. For this work a radial basis function (Equation

5.8) with hyperparameters α , ℓ , and σ was selected as the kernel:

$$k(x, x') = \sigma^2 \left[1 + \frac{(x - x')^2}{2\alpha\ell^2} \right]^{-\alpha}, \quad (5.8)$$

where σ^2 is the kernel scale factor, α is the inverse variance parameters, and ℓ is the length scale defined as the distance between x and x' where the covariance can be considered zero. Similar to variance, the scale factor describes the average squared distance of the kernel function from its mean. Hyperparameter α characterizes the "smoothness" of the function where increasing α leads to smoother responses. Hyperparameters, α , ℓ , and σ , were specified through an optimization process that maximized the log-marginal likelihood defined as.

$$\mathcal{H} = \log p(y_i|x_i, \Theta) = -\frac{1}{2} \log |k(x, x')| - \frac{1}{2} (y_i - \mu)^\top k(x, x')^{-1} (y_i) - \frac{n}{2} \log(2\pi), \quad (5.9)$$

where y_i is a vector of observed target values (QoIs), the mean function μ is assumed zero, n is the number of components in x_i and Θ represents the hyperparameters in the kernel function. For the optimization, hyperparameter bounds were specified as $\sigma^2 \in [0.1, 10]$, $\alpha \in [10^{-4}, 2]$, and $\ell \in [0.1, 2]$. As suitable hyperparameters values were not known prior to data fitting bounds for σ^2 and α were adjusted such that the bound were not reached during the hyperparameters optimization. However, ℓ was limited to 2 to prevent overfitting [37].

5.2 Surrogate Model Sample Space

For the present case a multivariate Gaussian was created using data over a pseudo-random five-point linear sample across parameter bounds $\beta_p, C_{\epsilon 4} \in [0.1, 1]$ and $\beta_d, C_{\epsilon 5} \in [0.1, 4]$. The sampling lower limits were set to 0.1 to prevent one element of the pair from obviating the other as shown in Equation 4.4. The five points were

$\beta_p, C_{\epsilon 4} \in [\underline{0.1}, 0.325, \underline{0.55}, 0.775, \underline{1.0}]$ and $\beta_d, C_{\epsilon 5} \in [\underline{0.1}, 1.075, \underline{2.05}, 3.025, \underline{4.0}]$. The underlined values were used to construct the surrogate model resulting in 81 samples (3^4) and the non-underlined values were used for verification totaling 16 samples (2^4). For a single set of parameters $(\beta_p, \beta_d, C_{\epsilon 4}, C_{\epsilon 5})$ a “model call” required two OpenFOAM model runs (one each for the $I_{3\%}$ and $I_{15\%}$ boundary conditions). These 97 sample points (81+16) were run in OpenFOAM to create the data set \mathcal{D} to build a Gaussian process regression model and validate the QoI response surface as a function of the adjustable turbulence parameters. The \mathbf{x}_* testing points for this case was variable and depended on the optimization routine.

5.3 Building the Surrogate Model

From each OpenFOAM model run, the simulated QoIs at the 10 points downstream of the turbine, yielded 810 data points to train each GPR model and 160 verification points. Using the training data from the OpenFOAM model the data were separated into four groups by QoI. For a given set of turbulence source parameters $(\beta_p, \beta_d, C_{\epsilon 4}, C_{\epsilon 5})$ at a given boundary condition ($I_{3\%}, I_{15\%}$) the centerline wake response (U and I) at 10 distances behind the turbine were separated out for Gaussian process regression. By this separation four GPR models were constructed ($U_{I=3\%}, U_{I=15\%}, I_{I=3\%}, I_{I=15\%}$). Each model will intake the four turbulence source parameters and then output the predicted wake response at 10 points downstream for a given QoI (specified by the model name) at the subscript turbulent intensity BC. As a shorthand, these four models will be referred to as $U_{3\%}, U_{15\%}, I_{3\%}, I_{15\%}$. The GPR surrogate models were constructed using Python version 2.7.12 with Numpy version 1.13.1. Data storage and handling were completed using Scipy 0.19.1 and Pandas version 0.20.3. The GPR itself was trained using Scikit-Learn 0.19.2. Plots were made using Matplotlib version 1.5.3. [22, 24, 35, 43].

Prior to creating the Gaussian posterior the kernel was tuned to the data set \mathcal{D} by

maximizing the marginal probability of \mathbf{f} given \mathbf{x} by adjusting the hyperparameters $(\sigma, \ell, \alpha \in \Theta)$ ($p(\mathbf{f}|\mathbf{x}, \Theta)$, Equation 5.9). The kernel asserts prior knowledge of smoothness into the model and therefore this tuning of the kernel using hyperparameters determines the smoothness of the provided data set \mathcal{D} . The GRP model was trained to maximize the log-marginal likelihood by adjusting the kernel hyperparameters using the SciPy optimization routine, `fmin_l_bfgs_b`, which is a quasi-Newton algorithm for constrained optimization [8, 24]. Five restarts (the number of times an optimization was initiated using random hyperparameters) were specified; increases did not further improve the kernel optimizations shown in Table 5.1.

Table 5.1: Optimized GPR kernel hyperparameters.

Q	σ	ℓ	α	Log-marginal likelihood
$U_{3\%}$	0.35	2.0	0.024	1,211
$U_{15\%}$	0.16	2.0	0.044	1,831
$I_{3\%}$	0.17	2.0	0.082	1,813
$I_{15\%}$	0.11	2.0	0.055	2,143

Table 5.1 shows each GPR model (Q) as the QoI it represents and the I_∞ of the case specified as the subscript (Section 5.3). The hyperparameters σ, ℓ, α are the final values used in the kernel as specified by Equation 5.8. The log-marginal likelihood was the objective function (Equation 5.9) to maximize. As the maximum value is dependent on the data there is no target set for a log-marginal likelihood but the relative values between each Q provide some insight into the kernels ability to predict the smoothness of the data set \mathcal{D} . Most notably in Table 5.1 all ℓ hit the maximum at 2.0, which is due to the tendency of GPR to overfit the data [37]. Not limiting ℓ resulted in $\ell = 4.8, 10, 5.5,$ and 5.8 for $U_{3\%}, U_{15\%}, I_{3\%},$ and $I_{15\%}$, respectively, while only raising the log-marginal likelihood on the order of 2%. The small value of α suggests that the GPR are less smooth than a radial basis function while σ is reasonably consistent across the models. The optimized kernel values in

Table 5.1 may demonstrate the need for a separate model for each QoI and ambient TI . Looking to the I model while the variance was similar to the ability to adjust α allowed the optimization scheme to increase the log-marginal likelihood to 2143.7 (an 18% increase).

5.4 Surrogate-model Fitting Error

A test of model performance at the 16 turbulence source parameter sets excluded from GPR training checked how well the surrogate model represented the OpenFOAM model. As discussed in Section 5.2 these 16 points were in between and furthest from the 81 data points used to train the GPR. The results are listed in Table 5.2 and show that all $\text{MAPE}_{\text{srgt}/\text{sim}}$ results were $< 10\%$ for each QoI (Q). The better performance of the $I_{15\%}$ is not guaranteed but potentially expected due to the higher log-marginal likelihood noted in Table 5.1.

Table 5.2: GPR surrogate model performance as compared to "true" OpenFOAM model.

Q	$\text{MAPE}_{\text{srgt}/\text{sim}}$	$\text{MSE}_{\text{srgt}/\text{sim}}$	$ L_{\infty} _{\text{srgt}/\text{sim}}$
$U_{3\%}$	4.8%	0.7	14.9%
$U_{15\%}$	1.5%	0.2	9.7%
$I_{3\%}$	9.8%	17.8	49.7%
$I_{15\%}$	3.4%	3.5	21.1%
Mean	4.9%	5.5	23.9%

5.5 Surrogate Optimization Framework

The process to build, verify, and optimize the surrogate model is given in Figure 5.7. Up to this point the variables have been identified (Section 4.4), the data collected through the OpenFOAM parameter space (Section 5.2), four GPR surrogate models were constructed (Section 5.3), and confirmed to each have less than 10% error by

Chapter 5. Surrogate Model Development

MAPE (Section 5.4). Now the optimization of the turbulence parameters is performed by wrapping PEST around the four GPR model to minimize the average predicted error compared to experiment. This optimization model is now labeled GPR-PEST. Following the GPR-PEST calibration, the optimal parameters were supplied to the OpenFOAM model to verify that the fitting error at the GPR-PEST optimized point was $<10\%$ on average. If the fitting error was greater than 10% then this process (build GPR and minimize predicted error) was repeated, iteratively, updating the GPR model training data using the optimized point until the fitting error at the GPR-PEST optimized point was $<10\%$ on average. Finally, the GPR-PEST parameters were evaluated in OpenFOAM simulations according to the three performance metrics listed in Table 4.2 and the final error was reported.

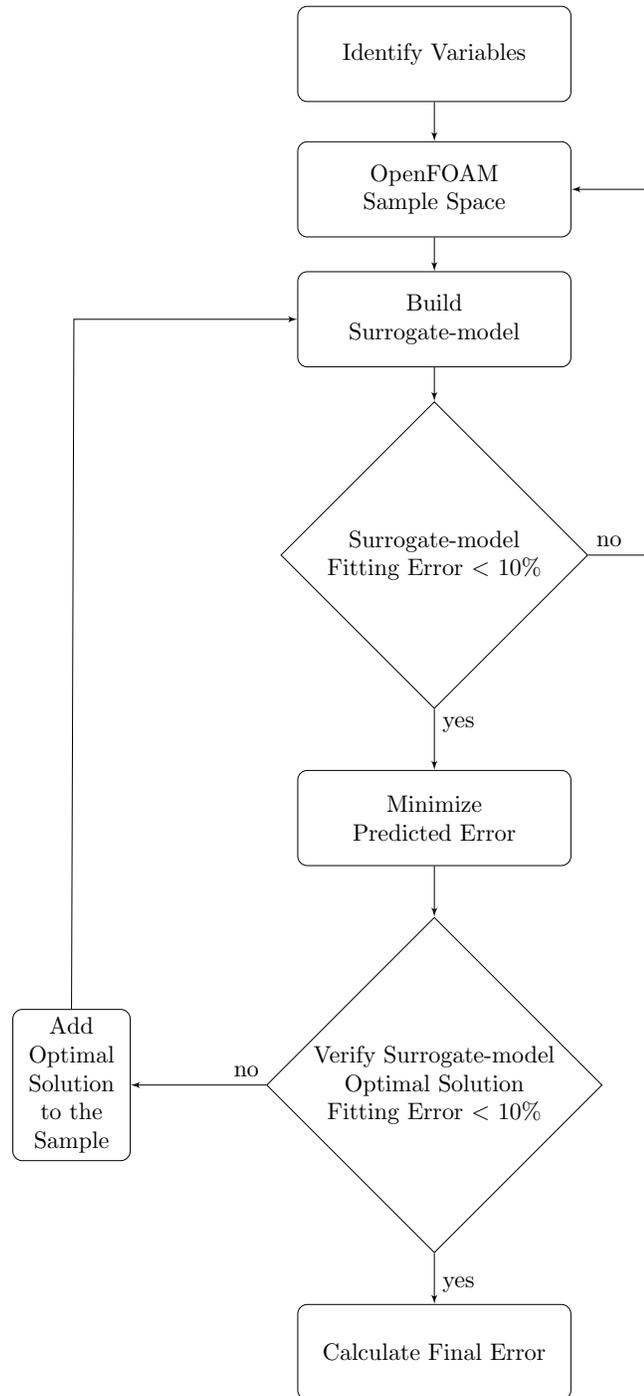


Figure 5.7: Flow diagram for the surrogate-model framework.

Chapter 6

Results and Discussion

The optimized parameter values for the GPR-PEST are organized in Table 6.1. Additionally, an optimization called OpenFOAM-PEST has been introduced which represents a traditional optimization of the full model (in contrast to a surrogate model). By comparing the GPR-PEST results to the OpenFOAM-PEST results one may draw conclusions about the efficiency and accuracy of the GPR-PEST optimization approach introduced here albeit the optimization method is not global. The GPR column has two final results (Final₁, Final₂) as two surrogate model iterations were required to achieve a complete optimization (Section 5.5). Two iterations were required because while the first GPR-PEST optimization (Final₁) minimized predicted error ($\text{MSE}_{\text{srgt/exp}}$) to 0.084, the fitting error ($\text{MAPE}_{\text{srgt/sim}}$) at the optimal solution was equal to 22.4% exceeding the 10% acceptance criterion (Figure 5.7). Therefore, the OpenFOAM results from the Final₁ case ($\beta_p, \beta_d, C_{\epsilon 4}, C_{\epsilon 5} = 1.0, 1.05, 1.0, 0.1$) were added to the original 81 OpenFOAM sample points and the GPR model was rebuilt now using 82 training points. A GPR-PEST optimization identical to the first was repeated on the new GPR surrogate model resulting in the optimal solution Final₂. This optimization (Final₂) decreased the predicted error to 0.043 with an acceptable fitting error of 3.6%. This concluded the GPR-PEST optimization with a total of 99 model calls (81 training, 16 verification, 2 final). The

Table 6.1: GPR-PEST and OpenFOAM-PEST optimized parameters and corresponding performances.

Parameter	GPR		OpenFOAM
	Final ₁	Final ₂	Final
β_p	1.00	1.00	1.00
β_d	1.05	0.80	1.84
$C_{\epsilon 4}$	1.00	1.00	0.77
$C_{\epsilon 5}$	0.10	0.16	0.13
Predicted Error	0.084	0.043	–
Fitting Error	22.4%	3.6%	–
Final Error	–	0.0358	0.0347

GPR-PEST final error ($MSE_{sim/exp}$) was 0.0358, lower than the GPR predicted error (0.043).

The success of the GPR-PEST optimization was determined by comparison to an equivalent optimization of the OpenFOAM simulation (OpenFOAM-PEST) shown in the column "OpenFOAM" of Table 6.1. The OpenFOAM-PEST further minimized the final error (0.0347) but the GPR-PEST achieve a result within 3.1% of the OpenFOAM-PEST result. GPR-PEST and OpenFOAM-PEST turbulence parameter values differed primarily for parameters β_d and $C_{\epsilon 4}$. OpenFOAM-PEST parameters plugged into the GPR model returned high predicted error and therefore without expanding the training data the GPR model would not achieve the same parameter solution as OpenFOAM-PEST. While definitive discussion of the turbulence parameter differences (β_d and $C_{\epsilon 4}$) cannot be made because k and ϵ are both coupled and nonlinear, the direction of the parameter changes makes intuitive sense following from Equations 4.3 and 4.4. It is possible that the GPR-PEST solution counterbalances the decrease of β_d (increasing k) with the increase of $C_{\epsilon 4}$ (increasing ϵ / dissipation of k) which results in a nominally similar solution to the OpenFOAM-PEST solution.

The fraction of tke converted to wake-generated kinetic energy by drag represent by β_p was determined to be 1.0 by both the GPR-PEST and the OpenFOAM-

PEST. Converting 100% *tke* to wake-generated kinetic energy by drag is physically unreasonable. Given this result and the drastic improvement from the standard $k-\epsilon$ results it is not surprising that these turbulence source terms are accounting for model form error beyond the turbulence model. Further, the GPR-PEST *final*₂ solution also hits the optimization upper-bound of $C_{\epsilon 4}$. This ostensibly suggests that if permitted the GPR-PEST optimization may have increased $C_{\epsilon 4}$ beyond 1.0. In order to extend the surrogate model training data at the boundary and retain identical information as the other boundary points would require 27 (3^3) additional model runs. However the user may implement many simplifications to keep additional model calls to a minimum. It can be seen from the OpenFOAM-PEST result that within the given range of turbulence parameters a lower objective function value ($\text{MSE}_{\text{sim/exp}} = 0.0347$) existed. This demonstrated both that extending the boundary using an additional 27 model calls may have not been the best use of expensive computational resources and the susceptibility of the GPR surrogate model to find local minima.

The parameters used in a gradient-based search optimization tool such as PEST are often adjusted following an iterative procedure [64]. The OpenFOAM-PEST results reported in Table 6.1 took on the order of thousands of OpenFOAM model calls to self tune the optimization software PEST and included the input of outside PEST experts. This makes the surrogate model optimization on the thousands of percent more efficient. However, to provide a lower bound for comparison for the total number of model calls between the surrogate procedure and the OpenFOAM-PEST optimization a new PEST control file was created following the guidance of *Getting the Most out of PEST* [14]. This singular PEST run took a total of 134 model calls but returned a higher final error (0.0354) than the reported OpenFOAM-PEST results in Table 6.1. In order to achieve a final error as low as the one reported in Table 6.1 this PEST control file would need to be iteratively adjusted. However, this PEST optimization serves as a conservatively lower bound for the total number of model calls. The GPR-PEST model used a total of 99 OpenFOAM model calls thereby

enacting only 74% of the computational expense of this single OpenFOAM-PEST optimization (134 model calls). As the GPR-PEST wall clock time is on the order of minutes it was neglected from the time comparison. This decrease in model calls and the ability to utilize all information from expensive computations makes surrogate models of practical interest to researchers and practitioners alike.

6.1 Surrogate & OpenFOAM Parameter Values Compared to Experiment

The final error reported in Table 6.1 can be seen for the GPR-PEST and OpenFOAM-PEST out for each QoI (Q) in Table 6.2 in the "Mean" row under the $MSE_{sim/exp}$ column. For each solution the primary contributor to $MSE_{sim/exp}$ is $I_{3\%}$. This was due to the experimentally measured I increasing at four-diameters downstream as detailed in Mycek et al. [38]. Here it was not expected that the actuator disc model could capture the delayed shear layer mixing caused by the low ambient turbulence ($I_\infty = 3\%$) as it did not preferentially add low k in the center of the turbine and high k at the edges of the disc. A good model should be valid over a large range of turbulence values and therefore the model was tuned to the 3% ambient turbulence but it should be noted that ambient turbulence of 3% is unlikely in turbine deployment conditions. It is encouraging that the least contributor to $MSE_{sim/exp}$ is $I_\infty = 15\%$ for both U and I as I_∞ near 15% is typical in turbine deployment sites [38]. Comparing GPR-PEST and OpenFOAM-PEST the average model fitness was 14.5% and 13.6% by $MAPE_{sim/exp}$ respectively. The average maximum error ($|L_\infty|$) was nominally equivalent between the GPR-PEST and OpenFOAM-PEST solutions.

The optimization results are graphically compared to experiment and the standard $k-\epsilon$ model in in Figure 6.1. The experiment data was shown as red dots, the GPR-PEST result is a dashed curve and the OpenFOAM-PEST solution is a solid curve. Further the standard $k-\epsilon$ actuator disc results (Table 4.3) were plotted as a blue

Table 6.2: Errors in simulated QoIs using optimized parameters from the GPR-PEST and OpenFOAM-PEST calibrations.

Q	GPR			OpenFOAM		
	$\text{MAPE}_{\text{sim/exp}}$	$\text{MSE}_{\text{sim/exp}}$	$ L_{\infty} _{\text{sim/exp}}$	$\text{MAPE}_{\text{sim/exp}}$	$\text{MSE}_{\text{sim/exp}}$	$ L_{\infty} _{\text{sim/exp}}$
$U_{3\%}$	11.8%	0.0185	21.6%	11.9%	0.0193	22.1%
$U_{15\%}$	9.5%	0.0227	38.6%	8.8%	0.0256	41.2%
$I_{3\%}$	28.2%	0.0917	44.1%	27.1%	0.0880	43.7%
$I_{15\%}$	8.5%	0.0102	16.2%	6.4%	0.0058	11.5%
Mean	14.5%	0.0358	30.1%	13.6%	0.0347	30.2%

dash-dot curve for reference to how the solution changed with the canopy turbulence parameters. Both the GPR-PEST and OpenFOAM-PEST optimized results were comparable as noted in Table 6.2. Excluding $I_{15\%}$ neither turbulence parameter solution precisely matched the experimental data indicating structural model error. The plots show graphically that for each of the quantitative metrics in Table 6.2 the near field was the primary contributor to error for the QoI U . Correct representation of the near field velocity using an actuator disc was a known issue [6]. Further, the plots definitively show the actuator disc model’s failure to model the second bump in I at $I_{\infty} = 3\%$ for diameters four to seven.

6.2 Limitations and Future Research

The research presented here was limited by simplifications used in sampling, bounding, building, and optimizing the surrogate model. Future research will demonstrate effective ways to visualize the quality of the input data for a model of this type. In addition, attempts to bound this four-dimensional space is an ongoing research area, which has proven to be computationally expensive. The GPR kernel function used here could have been improved through the use of kernel-optimization techniques. Use of a non-gradient based optimization, like particle swarm, that performs well for highly nonlinear spaces may prove valuable. Lastly, additional surrogate models are under development including optimized GPR kernels and deep learning models.

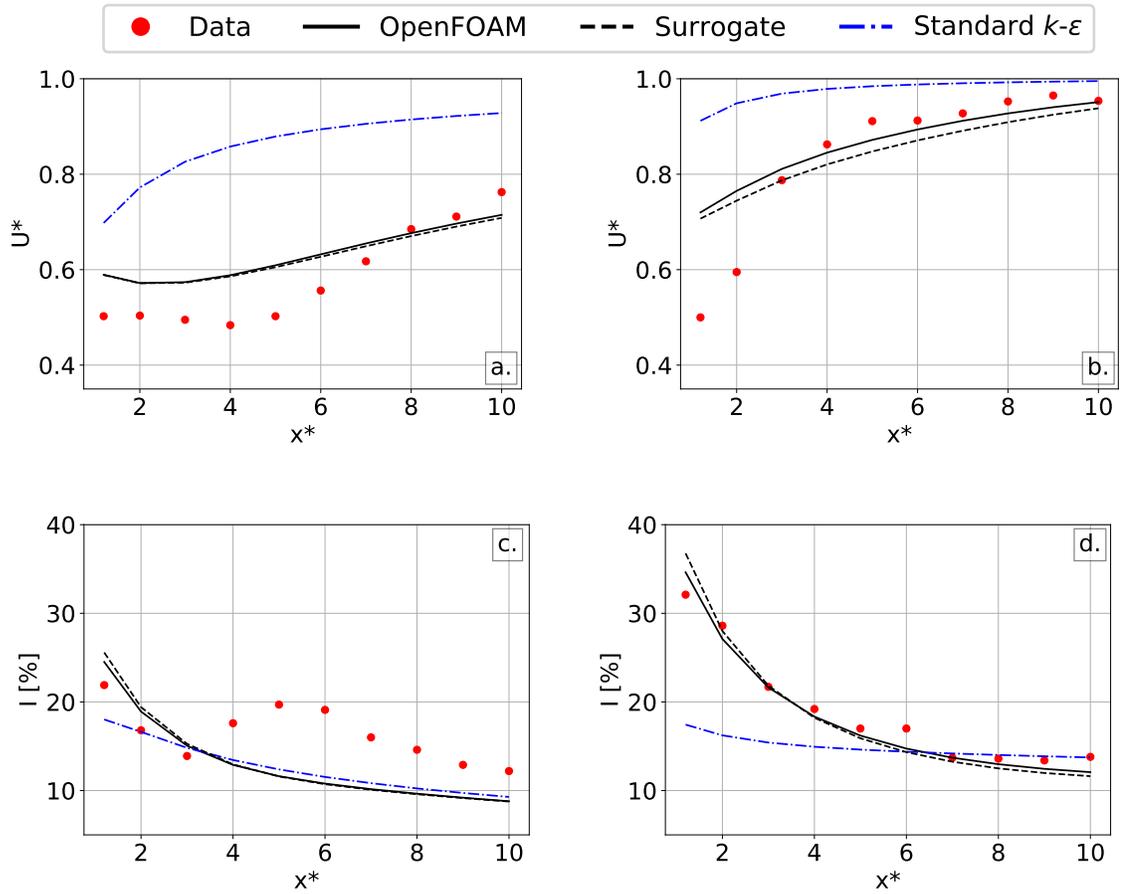


Figure 6.1: Comparison of experimental data (red dots) to optimized results using OpenFOAM (solid curve) and GPR (black-dashed curve) for QoIs U^* at (a) $I_{3\%}$ and (b) $I_{15\%}$ and I at (c) $I_{3\%}$ and (d) $I_{15\%}$. The standard $k-\epsilon$ actuator disc results are presented for reference (blue dash-dot curve).

Chapter 7

Conclusions

Licensing of MHK sites will require demonstration of minimal environmental impacts. Tuning the sites hydrodynamic power using these models is a natural use of these model. In order to build confidence the calibration of actuator disc models to experimental data is a necessary. This research presented a generalized approach to the tuning of an actuator disc model using a metamodel approach. A GPR surrogate of an OpenFOAM model was developed which represented the centerline wake velocity and turbulence intensity for a 3% and 15% ambient turbulence intensity. This GPR metamodel was shown to accurately represent the OpenFOAM model within 5% error by MAPE. Upon optimization the GPR-PEST model optimized turbulence parameters final error was within 3.1% of equivalently tuned OpenFOAM-PEST turbulence parameters for a lower bound of only 74% of the computational costs. The success of the metamodel in this application demonstrates the ability to utilize all computationally expensive model calls for this domain specific calibration effort. This research further reaches beyond the current application as the procedure laid out here may further be applied to any long running simulation or experimental measurements. Lastly, the success of this relatively simple surrogate model suggests that advanced surrogate models investigations are worthy of future research.

References

- [1] ALTMAN, N. S. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* 46, 3 (1992), 175–185.
- [2] ANDERSON, J. *Computational fluid dynamics : the basics with applications*. McGraw-Hill, New York, 1995.
- [3] BAHAJ, A., AND MYERS, L. Fundamentals applicable to the utilisation of marine current turbines for energy production. *Renewable energy* 28, 14 (2003), 2205–2211.
- [4] BAHAJ, A., MYERS, L., THOMSON, M., AND JORGE, N. Characterising the wake of horizontal axis marine current turbines, 2007.
- [5] BAHAJ, A., AND MYERS, L. E. Fundamentals applicable to the utilisation of marine current turbines for energy production. *Renewable energy* 28, 14 (2003), 2205–2211.
- [6] BATTEN, W. M., HARRISON, M., AND BAHAJ, A. Accuracy of the actuator disc-RANS approach for predicting the performance and wake of tidal turbines. *Phil. Trans. R. Soc. A* 371, 1985 (2013), 20120293.
- [7] BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.
- [8] BYRD, R. H., LU, P., NOCEDAL, J., AND ZHU, C. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* 16, 5 (1995), 1190–1208.
- [9] CHAMORRO, L. P., LEE, S.-J., OLSEN, D., MILLIREN, C., MARR, J., ARNDT, R., AND SOTIROPOULOS, F. Turbulence effects on a full-scale 2.5 MW horizontal-axis wind turbine under neutrally stratified conditions. *Wind Energy* 18, 2 (2015), 339–349.

References

- [10] CHAMORRO, L. P., TROOLIN, D. R., LEE, S.-J., ARNDT, R., AND SOTIROPOULOS, F. Three-dimensional flow visualization in the wake of a miniature axial-flow hydrokinetic turbine. *Experiments in Fluids* 54, 2 (2013), 1459.
- [11] DELTARES. About delft3d. <https://oss.deltares.nl/web/delft3d/about>, 2017.
- [12] DOHERTY, J. Model-independent Parameter Estimation User Manual Part I: PEST, SENSAN and Global Optimisers. *Watermark Numerical Computing* (2016), pp. 390.
- [13] DOHERTY, J., AND CHRISTENSEN, S. Use of paired simple and complex models to reduce predictive bias and quantify uncertainty. *Water Resources Research* 47, 12 (2011).
- [14] DOHERTY, JOHN. Getting the Most out of PEST. http://www.pesthomepage.org/getfiles.php?file=pest_settings.pdf, 2013.
- [15] ELGHALI, S. B., BENBOUZID, M., AND CHARPENTIER, J. F. Marine tidal current electric power generation technology: State of the art and current status. In *Electric Machines & Drives Conference, 2007. IEMDC'07. IEEE International* (2007), vol. 2, IEEE, pp. 1407–1412.
- [16] ELGHALI, S. E. B., BALME, R., LE SAUX, K., BENBOUZID, M. E. H., CHARPENTIER, J. F., AND HAUVILLE, F. A simulation model for the evaluation of the electrical power potential harnessed by a marine current turbine. *IEEE Journal of Oceanic Engineering* 32, 4 (2007), 786–797.
- [17] FREITAS, N. D. Machine learning - introduction to gaussian processes.
- [18] FROUDE, R. E. On the part played in propulsion by differences of fluid pressure. *Trans. Inst. Naval Architects* 30 (1889), 390.
- [19] GAURIER, B., GERMAIN, G., FACQ, J.-V., JOHNSTONE, C., GRANT, A., DAY, A., NIXON, E., DI FELICE, F., AND COSTANZO, M. Tidal energy “Round Robin” tests comparisons between towing tank and circulating tank results. *International Journal of Marine Energy* 12 (2015), 87–109.
- [20] GERHART, P. M., GERHART, A. L., AND HOCHSTEIN, J. I. *Munson, Young and Okishi's Fundamentals of Fluid Mechanics*. Wiley, 2015.
- [21] HARRISON, M., BATTEN, W., MYERS, L., AND BAHAJ, A. Comparison between CFD simulations and experiments for predicting the far wake of horizontal axis tidal turbines. *IET Renewable Power Generation* 4, 6 (2010), 613–627.

References

- [22] HUNTER, J. D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9, 3 (2007), 90–95.
- [23] JAMES, S. C., JOHNSON, E. L., BARCO, J., AND ROBERTS, J. D. Simulating current-energy converters: SNL-EFDC model development, verification, and parameter estimation. *Renewable Energy* (2017).
- [24] JONES, E., OLIPHANT, T., PETERSON, P., ET AL. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 11/17/2017].
- [25] JONES, M. C., MARRON, J. S., AND SHEATHER, S. J. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association* 91, 433 (03 1996), 401.
- [26] KANG, S., BORAZJANI, I., COLBY, J. A., AND SOTIROPOULOS, F. Numerical simulation of 3D flow past a real-life marine hydrokinetic turbine. *Advances in Water Resources* 39 (2012), 33–43.
- [27] KATUL, G. G., MAHRT, L., POGGI, D., AND SANZ, C. One-and two-equation models for canopy turbulence. *Boundary-Layer Meteorology* 113, 1 (2004), 81–109.
- [28] KERWIN, J., AND HADLER, J. *The Principles of Naval Architecture Series: Propulsion*. Society of Naval Architecture and Marine Engineers (SNAME), 2010.
- [29] KHAN, M., BHUYAN, G., IQBAL, M., AND QUACOUE, J. Hydrokinetic energy conversion systems and assessment of horizontal and vertical axis turbines for river and tidal applications: A technology status review. *Applied energy* 86, 10 (2009), 1823–1835.
- [30] KUNDU, P., COHEN, I., AND DOWLING, D. *Fluid mechanics*. Academic Press, Waltham, MA, 2012.
- [31] LAUNDER, B. E., AND SPALDING, D. B. The numerical computation of turbulent flows. In *Numerical Prediction of Flow, Heat Transfer, Turbulence and Combustion*. Elsevier, 1983, pp. 96–116.
- [32] LEE, J. H., PARK, S., KIM, D. H., RHEE, S. H., AND KIM, M.-C. Computational methods for performance analysis of horizontal axis tidal stream turbines. *Applied Energy* 98 (2012), 512–523.
- [33] MAGANGA, F., GERMAIN, G., KING, J., PINON, G., AND RIVOALEN, E. Experimental characterisation of flow effects on marine current turbine behaviour

References

- and on its wake properties. *IET Renewable Power Generation* 4, 6 (2010), 498–509.
- [34] MALKI, R., WILLIAMS, A., CROFT, T., TOGNERI, M., AND MASTERS, I. A coupled blade element momentum–computational fluid dynamics model for evaluating tidal stream turbine performance. *Applied Mathematical Modelling* 37, 5 (2013), 3006–3020.
- [35] MCKINNEY, W. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (2010), S. van der Walt and J. Millman, Eds., pp. 51–56.
- [36] MICHELEN, C., MURRAY, J. C., NEARY, V. S., BARONE, M., ET AL. CACTUS open-source code for hydrokinetic turbine design and analysis: Model performance evaluation and public dissemination as open source design tool. Tech. rep., Sandia National Laboratories, 2014.
- [37] MOHAMMED, R. O., AND CAWLEY, G. C. Over-fitting in model selection with gaussian process regression. In *International Conference on Machine Learning and Data Mining in Pattern Recognition* (2017), Springer, pp. 192–205.
- [38] MYCEK, P., GAURIER, B., GERMAIN, G., PINON, G., AND RIVOALEN, E. Experimental study of the turbulence intensity effects on marine current turbines behaviour. part i: One single turbine. *Renewable Energy* 66 (2014), 729–746.
- [39] MYERS, L., AND BAHAJ, A. Experimental analysis of the flow field around horizontal axis tidal turbines by use of scale mesh disk rotor simulators. *Ocean Engineering* 37, 2-3 (2010), 218–227.
- [40] MYERS, L., AND BAHAJ, A. An experimental investigation simulating flow effects in first generation marine current energy converter arrays. *Renewable Energy* 37, 1 (2012), 28–36.
- [41] NEARY, V. S., GUNAWAN, B., HILL, C., AND CHAMORRO, L. P. Near and far field flow disturbances induced by model hydrokinetic turbine: ADV and ADP comparison. *Renewable Energy* 60 (2013), 1–6.
- [42] O’DOHERTY, T., MASON-JONES, A., O’DOHERTY, D., BYRNE, C., OWEN, I., AND WANG, Y. Experimental and computational analysis of a model horizontal axis tidal turbine. In *8th European Wave and Tidal Energy Conference (EWTEC)* (Uppsala, Sweden, 2009).
- [43] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., BLONDEL, M., PRETTENHOFER, P., WEISS, R., DUBOURG,

References

- V., VANDERPLAS, J., PASSOS, A., COURNAPEAU, D., BRUCHER, M., PERROT, M., AND DUCHESNAY, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
- [44] PELC, R., AND FUJITA, R. M. Renewable energy from the ocean. *Marine Policy* 26, 6 (2002), 471–479.
- [45] PERNER, P. *Machine Learning and Data Mining in Pattern Recognition: 9th International Conference, MLDM 2013, New York, NY, USA, July 19-25, 2013, Proceedings*, vol. 7988. Springer, 2013.
- [46] POGGI, D., PORPORATO, A., RIDOLFI, L., ALBERTSON, J., AND KATUL, G. The effect of vegetation density on canopy sub-layer turbulence. *Boundary-Layer Meteorology* 111, 3 (2004), 565–587.
- [47] POLI, R. Analysis of the publications on the applications of particle swarm optimisation. *Journal of Artificial Evolution and Applications* 2008 (2008).
- [48] PRADO, R. A. Reformulation of the momentum theory applied to wind turbines. *Journal of wind engineering and industrial aerodynamics* 58, 3 (1995), 277–292.
- [49] RANKINE, W. J. M. On the mechanical principles of the action of propellers. *Transactions of the Institution of Naval Architects* 6 (1865).
- [50] RASMUSSEN, C. Cki williams gaussian processes for machine learning mit press. *Cambridge, MA* (2006).
- [51] RÉTHORÉ, P.-E. M. *Wind turbine wake in atmospheric turbulence*. PhD thesis, Risø National Laboratory for Sustainable Energy, 2009.
- [52] RETHORE, P.-E. M., SØRENSEN, N. N., BECHMANN, A., AND ZAHLE, F. Study of the atmospheric wake turbulence of a cfd actuator disc model. In *2009 European Wind Energy Conference and Exhibition* (2009).
- [53] ROACHE, P. J. *Verification and Validation in Computational Science and Engineering*. Hermosa Pub, 1998.
- [54] ROACHE, P. J. Code verification by the method of manufactured solutions. *Journal of Fluids Engineering* 124, 1 (2002), 4–10.
- [55] ROBERT, C. Machine learning, a probabilistic perspective, 2014.
- [56] ROC, T., CONLEY, D. C., AND GREAVES, D. Methodology for tidal turbine representation in ocean circulation model. *Renewable Energy* 51 (2013), 448–464.

References

- [57] RUPPERT, D., SHEATHER, S. J., AND WAND, M. P. An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association* 90, 432 (1995), 1257–1270.
- [58] SANDIA ENERGY. SNL-Delft3D-CEC. <http://energy.sandia.gov/energy/renewable-energy/water-power/market-acceleration-deployment/snl-delft3d-cec/>, 2017.
- [59] STALLARD, T., COLLINGS, R., FENG, T., AND WHELAN, J. Interactions between tidal turbine wakes: Experimental study of a group of three-bladed rotors. *Phil. Trans. R. Soc. A* 371, 1985 (2013), 20120159.
- [60] THE OPENFOAM FOUNDATION LTD. OpenFOAM. <https://openfoam.org/version/3-0-1/>, 2017.
- [61] TOPPER, M. B., AND INGRAM, D. M. Identifying the frontier of knowledge for marine renewable energy research. In *9th European Wave and Tidal Energy Conference* (2011).
- [62] VREUGDENHIL, C. B. *Numerical methods for shallow-water flow*, vol. 13. Springer Science & Business Media, 2013.
- [63] WARNER, J. C., SHERWOOD, C. R., ARANGO, H. G., AND SIGNELL, R. P. Performance of four turbulence closure models implemented using a generic length scale method. *Ocean Modelling* 8, 1 (2005), 81 – 113.
- [64] WATERMARK NUMERICAL COMPUTING. PEST Model-Independent Parameter Estimation User Manual Part I: PEST, SENSAN and Global Optimisers . <http://www.pesthomepage.org/Downloads.php>, 2016.
- [65] WILLIAMSON, B. J., BLONDEL, P., ARMSTRONG, E., BELL, P. S., HALL, C., WAGGITT, J. J., AND SCOTT, B. E. A self-contained subsea platform for acoustic monitoring of the environment around marine renewable energy devices—field deployments at wave and tidal energy sites in orkney, scotland. *IEEE Journal of Oceanic Engineering* 41, 1 (2016), 67–81.