A surrogate-model assisted approach for optimising the size of tidal turbine arrays

D.M. Culley [a*], S.W. Funke [b], S.C. Kramer [a] & M.D. Piggott [a,c]

[a] *Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, London, UK*

[b] *Center for Biomedical Computing, Simula Research Laboratory, Oslo, Norway*

[c] *Grantham Institute, Imperial College London, London, UK*

[*] *Corresponding author, dmc13@imperial.ac.uk*

Abstract

The new and costly nature of tidal stream energy extraction technologies can lead to narrow margins of success for a project. The design process is thus a delicate balancing act – to maximise the energy energy extracted, while minimising cost and risk. Scenario specific factors, such as site characteristics, technological constraints and practical engineering considerations greatly impact upon both the appropriate number of turbines to include within a tidal current turbine array (array size), and the individual locations of those turbines (turbine micro-siting). Both have been shown to significantly impact upon the energy yield and profitability of an array.

The micro-siting arrangement for a given number of turbines can significantly influence the power extraction of a tidal farm. Until the layout has been optimised (a process which may incorporate turbine parameters, local bathymetry and a host of other practical, physical, legal, financial or environmental constraints) an accurate forecast of the yield of that array cannot be determined. This process can be thought of as 'tuning' an array to the proposed site to maximise desirable outcomes and mitigate undesirable effects.

The influence of micro-siting on the farm performance means that determining the optimal array size needs to be coupled to the micro-siting process. In particular, the micro-siting needs to be repeated for any new trial array size in order to be able to compare the performance of the different farm sizes. Considering the large number of design variables in the micro-siting problem (which includes at least the positions of each turbine) it becomes clear that algorithmic optimisation is a key tool to rigorously determine the optimal array size and layout.

This paper proposes a nested optimisation approach for solving the array size and layout problem. The core of this approach consists of two nested optimisation procedures. The 'outer' optimisation determines the array size. At each 'outer' iteration the power extracted by N turbines is found via a separate optimisation of their micro-siting on the proposed site. The 'inner' optimisation is treated as a computationally expensive black-box solver, mapping array size to power (and additionally returning the optimal micro-siting design). This forms the basis of a practical approach to the

array sizing problem based on Bayesian optimisation, in which a surrogate model is built and used to maximise the utility of each evaluation made by the solver.

This paper introduces and reports on the implementation of this novel surrogate-assisted array design approach which, coupled with a simple financial model is demonstrated through optimisation of array size for two test scenarios to maximise the financial return on the array for the developer.

Keywords:
marine renewable energy, tidal turbines, gradient-based optimisation, adjoint method, shallow water equations, array layout

---

1 Introduction

Tidal current turbines (TCTs) use the momentum of tidal currents to generate electricity. Characteristics such as the high power density of the resource and the reliability with which it can be forecast make tidal current power conversion an attractive technology for development. It has been demonstrated by Funke et al. [9], Malki et al. [17], Vennell et al. [27] and others, that the energy yield of an array of a given number of turbines can be heavily contingent upon the micro-siting design of the array; that is, the spatial arrangement and placement of the individual turbines. This is because turbine wakes and complex bathymetry can cause a high spatial variation in flow speed and the power extracted by a turbine has a cubic dependency upon that flow speed.

Bryden and Couch [2] demonstrated that harvesting tidal energy has a sufficiently significant effect upon that flow that it is imperative to design for the *resultant* flow. That is to say, when modelling an array, the flow field must be computed with the turbines in place. This is particularly true with large arrays, when blockage effects begin to dominate – potentially causing bulk flow redirection. Consequently, in the process of a layout optimisation, each iteration in which the micro-siting design is adjusted requires the flow field to be recalculated. The more detail in which the physical processes governing the flow are modelled, the more confidence may be placed in the solution but, of course, the more computationally expensive the solve.

OpenTidalFarm is an open source package for tidal farm layout optimisation, which uses gradient-based optimisation to minimise the number of iterations required to converge. Requiring far fewer iterations than global optimisation approaches (such as genetic algorithms) means that each iteration can be more computationally expensive, which facilitates the physical processes being modelled to a higher level of fidelity than would otherwise be achievable within the same computational budget. For completeness, *OpenTidalFarm* is presented in detail in section 3.

While the micro-siting optimisation problem has received some limited attention [9, 21, 23, 25], the more fundamental question of how many turbines there should be in the first place, has been given very little practical consideration. Resource assessment methodologies often make unrealistic assumptions as to how the resource may be harnessed – for example assuming that turbine barrages stretch across tidal straights from shore to shore, not leaving space for shipping or acknowledging that nearer shore there is insufficient depth to install turbines. Meanwhile, theoretical work – such as Vennell [26] – has framed the question in terms of blockage ratio and turbine density – a useful abstraction which facilitates general analysis of sites of various size (for example, two turbines in a narrow channel may have the same blockage as one hundred turbines

in a wider channel). These models serve a useful purpose in identifying the upper bound of power available in a given region, and in helping us to understand how the hydrodynamic characteristics of the site/array change as the array is tuned. However, for a developer – who has a clearly defined turbine deployment area leased from the land-owner (in the UK this is the Crown Estate) – these approaches are less helpful, and a more practical methodology is called for.

Since the flow dynamics are coupled to the locations of the turbines, and the power produced by those turbines is highly sensitive to the flow velocity, accurately forecasting the potential power output of an array of a given number of turbines, N, is both site and array-design specific and requires a micro-siting optimisation. Therefore in the course of determining the array size for a given site through iterative optimisation, significant computational effort is expended each time a different N is sampled. Consequently the problem is a computationally expensive integer optimisation whereby each time the performance of a given array size is evaluated an 'inner' micro-siting optimisation for that array is required.

The novel contribution of this study is the proposal of a practical methodology to study this relationship. The goal is that this approach may be adopted to scope projects on potentially complex sites and to specify the optimal design (both in terms of turbine numbers and their positions) of potentially large arrays. In this work, use of developer's return upon investment is proposed as the metric of array design quality, and a simple financial model is implemented to demonstrate the utility of this approach.

In the following section, the array sizing optimisation problem is formulated mathematically and split into an 'inner' continuous and 'outer' integer optimisation problem. The 'inner' micro-siting design optimisation approach is briefly reviewed in section 3 and the implementation of a surrogate model for the 'outer' optimisation is explored in section 4. The combined model is detailed in section 5 and applied to a simple, idealised, test case in section 6. In section 7 the approach is demonstrated on a more realistic case based upon the initial deployment plans for a project which is currently under development in the Inner Sound of the Pentland Firth.

2 Problem Formulation

The power of a tidal current turbine array is computed as the sum of the power extracted from the flow due to the presence of the turbines. If we encode the Cartesian coordinates of each turbine in a vector, $\mathbf{m}$, where $\mathbf{m} = ((x_i, y_i) \mid i = 1, \ldots, N)^T$ the power extracted by the turbine may be expressed as

$$\text{Power}(\mathbf{m}) = \int_\Omega \rho c_t(\mathbf{m}, \mathbf{u}) ||\mathbf{u}||^3 \, \mathrm{d}x. \tag{1}$$

Here, turbines are modelled as areas of increased friction [16], described by c_t (and defined in detail below), ρ is the density of the water and $\mathbf{u} : \Omega \times [0, T) \to \mathbb{R}^2$ is the depth-averaged flow velocity which is the solution to the shallow water equations

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nu \nabla^2 \mathbf{u} + g \nabla \eta + \frac{c_b + c_t(\mathbf{m}, \mathbf{u})}{H} \|\mathbf{u}\| \mathbf{u} = 0, \tag{2a}$$

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (H \mathbf{u}) = 0. \tag{2b}$$

The viscosity coefficient, acceleration due to gravity, and quadratic bottom friction are represented by ν, g, and c_b respectively. The total water depth, $H : \Omega \to \mathbb{R}$ is given by $H = h + \eta$ where $h : \Omega \to \mathbb{R}$ is the bottom depth and $\eta : \Omega \to \mathbb{R}$ is the free surface displacement. (2a) and (2b) are discretised using Taylor-Hood ($\mathcal{P}_2\mathcal{P}_1$) element pairs [24] and solved using the finite element method over an irregular triangular mesh [10].

The thrust exerted on the flow by a turbine is parametrised by the thrust coefficient, $C_t \equiv C_t(\mathbf{u})$, which is a function of the flow speed as defined by a 'thrust curve'. These curves are device-dependent and as such are highly proprietorial and generally unavailable for academic publication. Support for user-defined turbine thrust parametrisation in *OpenTidalFarm* was developed for this paper, enabling users to more accurately represent the characteristics of the device of interest in the optimisation. Given the thrust curve can have a strong effect on the hydrodynamics local to the turbine, it will also have a strong effect on the interactions between turbines and therefore the optimal micro-siting design of the array.

From the Morison equation, the thrust applied to the flow by a turbine is

$$F_{3\mathrm{D}}(\mathbf{u}) = \frac{1}{2}\rho A_T C_t \|\mathbf{u}\|\mathbf{u}, \tag{3}$$

where A_T is the area of the disk (i.e. the swept area of the turbine rotor blades) and ρ is the fluid density. Recall that (2) is a depth-averaged model, in which the thrust is exerted by enhanced bottom drag, c_t,

$$F_{2\mathrm{D}}(\mathbf{u}) = \rho \int_\Omega c_t \|\mathbf{u}\|\mathbf{u} \, \mathrm{d}x. \tag{4}$$

So by depth-averaging (3) the correct amount of thrust as calculated in three-dimensions can be applied in two-dimensions by equating (3) and (4), yielding

$$\frac{1}{2} A_T C_t = \int_\Omega c_t \, \mathrm{d}x. \tag{5}$$

To satisfy (5), we define

$$\Xi(\mathbf{p}, \mathbf{x}) = \xi(\mathbf{p}_x, x)\xi(\mathbf{p}_y, y), \tag{6}$$

where $\mathbf{p} = (p_x, p_y)^T$ is the location vector describing the coordinates of the centre of the turbine, ξ is a bump function

$$\xi(p, x) = \begin{cases} \exp\left(1 - \dfrac{1}{1 - \left\|\dfrac{2(x-p)}{d}\right\|^2}\right) & \text{for } \left\|\dfrac{2(x-p)}{d}\right\| < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

and d is the plan-view diameter of the two-dimensional representation of the turbine. The turbine function is square in plan-form shape to support future work in which the

orientation of the turbine in relation to the direction of flow will be considered. In the meantime, since the grid-scale in the turbine area is (at minimum) in the order of $D/10$, a square as compared to a circular turbine representation is similar once discretised. Defining a scaling constant,

$$\Lambda = \frac{A_t}{2\int \Xi \, \mathrm{d}x} \ , \tag{8}$$

which can be computed once the turbine dimensions are known, (5) is now

$$c_t = \Lambda C_t(\mathbf{u})\Xi(\mathbf{p}, \mathbf{x}). \tag{9}$$

Since C_T is device dependent and typically proprietorial, an idealised curve is used here, which is adapted from Martin-Short et al. [18] and defined by

$$C_t(\mathbf{u}) = \begin{cases} c_t^{\text{rated}} \left(\dfrac{\tanh(10(\mathbf{u} - \mathbf{u}_{\text{in}})) + 1}{2} \right), & \text{if } \mathbf{u} \leq \mathbf{u}_{\text{in}} \\ c_t^{\text{rated}}, & \text{if } \mathbf{u}_{\text{in}} \leq \mathbf{u} \leq \mathbf{u}_{\text{rated}} \\ c_t^{\text{rated}} \dfrac{\mathbf{u}_{\text{rated}}^3}{\mathbf{u}^3}, & \text{if } \mathbf{u} \geq \mathbf{u}_{\text{rated}}. \end{cases} \tag{10}$$

This has been implemented as a default in *OpenTidalFarm*. Here, $\mathbf{u}_{\text{in}}$ and $\mathbf{u}_{\text{rated}}$ are the cut in and rated flow speeds respectively and c_t^{rated} is $C_t(\mathbf{u}_{\text{rated}})$. The tanh function is used to smooth the thrust cut in, given that the gradient-based optimisation algorithm being used can handle non-smoothness the underlying functional as long as it is not extreme [22]. Figures 1 and 2 show this thrust curve and the resulting power curve (computed using (1)) respectively.

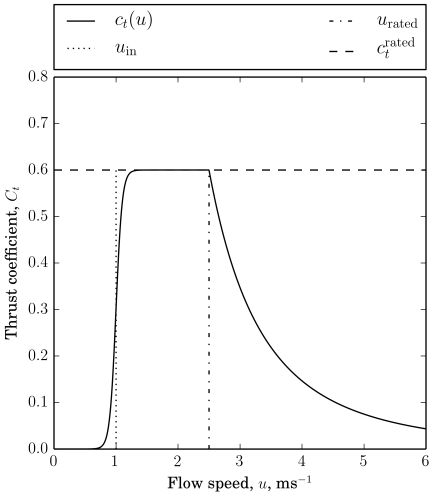


Figure 1: Thrust curve for idealised tidal turbine with a cut-in speed of $u_{\text{in}} = 1.0\,\text{ms}^{-1}$, and a rated speed of $u_{\text{rated}} = 2.5\ \text{ms}^{-1}$.

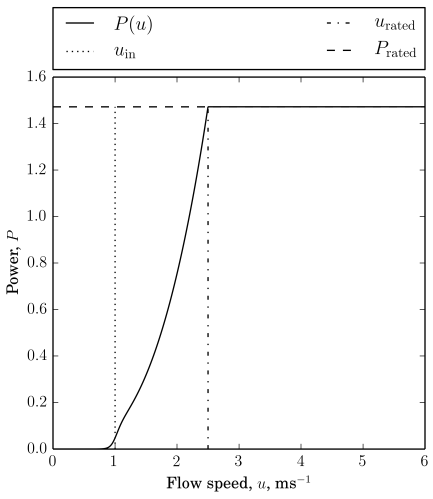


Figure 2: Power curve for idealised 20 m diameter turbine in MW, computed using (1).

Finally, an array of N turbines comprises the sum of the individual c_t functions;

$$c_t = \sum_{i=1}^{N} c_t^i. \tag{11}$$

Assuming the number of turbines, N, is fixed, then the maximum power that may be extracted from the flow is found to be

$$\begin{aligned} \tilde{P}(N) = \max_{\mathbf{m}} \quad & \text{Power}(\mathbf{m}) \\ \text{subject to} \quad & b_l \leq \mathbf{m} \leq b_u \\ & ||\mathbf{p}_i - \mathbf{p}_j|| \geq S_{\min} \quad \forall\ i \neq j. \end{aligned} \tag{12}$$

The first constraint in the above optimisation problem definition prevents turbines from leaving the leased turbine area. If the leased turbine area is rectangular then this may be represented as in (12) where b_l and b_u represent the lower and upper boundaries respectively. However, more general area constraints could be incorporated by adding a general inequality constraint of the form $g(\mathbf{m}) \leq 0$. The last constraint, $||\mathbf{p}_i - \mathbf{p}_j|| \geq S_{\min}\ \forall\ i \neq j$ – where $S_{\min}$ is the minimum spacing – prevents turbines being situated too closely to one another.

This is a *continuous* optimisation problem; given $N \in \mathbb{Z}^+$, the turbine locations which provide the maximum power, $\text{Power}(\mathbf{m}) \in \mathbb{R}$, are returned as the vector $\mathbf{m} \in \mathbb{R}^{2N}$. Since N resides in the integer space, finding the optimum number of turbines in an array is found as an *integer* optimisation problem

$$\max_{N} \quad \tilde{P}(N) \quad N \in \mathbb{Z}^+ \mid N \leq N_{\max}, \tag{13}$$

where $N_{\max}$ is the maximum number of turbines that the site is capable of supporting, which can be calculated from the minimum turbine distance constraints and the turbine lease area (though in reality may instead be limited by practical factors such as the number of turbines the developer can afford).

Using the notation $|_N$ to indicate the length of the preceding vector, the combined problem may be described as

$$\max_{N,\mathbf{m}} \quad P(\mathbf{m}|_N). \tag{14}$$

3 Optimisation of micro-siting using the adjoint

An approach to determining the optimal locations of turbines within an array is packaged in the code *OpenTidalFarm* [9]. *OpenTidalFarm* formulates the turbine layout problem in the form of (12); as an optimisation problem constrained by the shallow water equations. It uses a gradient based method to maximise the power output of the farm by controlling the locations of the individual turbines. The optimisation algroithm handles the spacing and boundary constraints.

The model is initialised with N turbines placed in an initial-guess layout within the turbine area. In future work, a hierarchy of lower fidelity models may be implemented to supply a 'better guess' for this starting layout [3]. The shallow water equations

(2) are solved and the power extracted from the flow by the array is calculated. This process maps $\mathbf{m} \in \mathbb{R}^{2N} \rightarrow \text{Power} \in \mathbb{R}$. The gradient of the power with respect to the turbine positions, $\mathrm{dPower} \,/\, \mathrm{d}\mathbf{m}$, is then computed by solving the adjoint shallow-water equations. This gradient, alongside the evaluation of the power functional are then used to improve the layout of the turbines for the next iteration.

The adjoint approach is particularly suited to determining the gradient of a functional (a function mapping to a scalar) as it does so at a computational cost which is essentially independent of the number of input parameters (in this case the two dimensional Cartesian coordinates of each turbine). For a full discussion on the computation and use of the adjoint in this context, the interested reader is directed towards Pinnau and Ulbrich [19], Gunzburger [13] and Funke and Farrell [8]. Computationally cheap access to gradient data means that gradient-based optimisation methods may be used to improve the turbine layout. Such first-order optimisation enables optimisation convergence in orders of magnitude fewer iterations than alternate zero-order, for example genetic, algorithms [1]. The benefit of this is that rather than using a low fidelity model to solve a large number of iterations each of which is necessarily very cheap, the same computational resource may instead be expended upon a higher fidelity (and hence more realistic) solver. One limitation of a gradient-based approach is that it is a *local* method – meaning that the local optimum will be returned which is dependent upon the initial guess of $\mathbf{m}$.

4 Surrogate modelling

The concept of surrogate modelling is to replace the (complex or expensive) model based on the *physical processes* being simulated – the *simulation* model – with a (simpler or cheaper) model which largely disregards the physical processes and is rather a mathematically constructed input-output map based on the observations made so far. For example, the shallow-water equations, (2), are based upon mathematical representation of conservation of properties. We may run an experiment twice with differing parameters using the shallow-water model, and then fit a function (which is the surrogate model) to the resulting data-set. The output at a different set of inputs can then be found by either re-running the experiment (requiring a shallow water solve), or else by evaluating the surrogate model at that point. The surrogate model is constructed via a fundamentally bottom-up approach, starting with data, using this to mathematically formulate an input-output mapping and then to evaluate it at unknown points rather than finding those values via the analytical model.

In this work, we use *OpenTidalFarm* to evaluate the maximum power that may be extracted from the site by a given number of turbines, which it does by optimising their locations – the array's micro-siting design. Taking a step back, we can treat *OpenTidalFarm* as a 'black-box' – an unknown, non-linear solver – which, given N, solves (12) and returns $\tilde{P}$. While *OpenTidalFarm* is efficient at the performing this task, it is still computationally expensive in the context of running a great number of separate times, evaluating different N's. This motivates the use of surrogate modelling to help minimise the number of calls to *OpenTidalFarm* in the course of optimising N.

Using surrogate models in this way – to replace an expensive black-box functional – is very common in computational approaches to aerodynamic design [15, 14]. The surrogate chosen for this application is a stochastic process, and so the overall approach is known as Bayesian optimisation. Stochastic processes and Bayesian optimisation are two large intersecting areas of a group of techniques often known as 'machine learning'

and therefore represent large subject areas in their own right. A brief overview of the theory of being employed will be presented in the remainder of this section. Once again, the interested reader is directed to more complete resources such as Rasmussen [20], Zhou et al. [28], Keane and Nair [15] for more details.

4.1 Gaussian process regression

Gaussian process regression (GPR) was chosen to construct the surrogate model because it accounts for and quantifies uncertainty in the predictions it produces, supports multi-fidelity sampling (which will be explained in due course) and is fairly well established since it is one of the earliest developed stochastic process regressions. GPR was developed in the 1950's for geo-statistical applications – namely to estimate the level below the surface of strata formed by minerals of interest, based on borehole measurements taken elsewhere. GPR is part of the field of statistical learning theory which has its own distinct notation – consequently we have made the choice to keep discussion to a high level.

The key idea is that variables are no longer treated deterministically but are instead treated as *random variables*. This means the variable can carry several values each with an associated probability – i.e. can be represented by a probability density function (pdf). We begin by making assumptions about our underlying problem by defining, based on judgement, what this pdf may look like, this is called a *prior*. When samples are generated by running the analysis code, the model is 'trained' and combining the prior with these observations the *posterior* distribution is obtained. It is this which we use as a surrogate model. These observations are called the *training set*, $\mathcal{D} = \{(\mathbf{x}_i, f(\mathbf{x}_i))|i = 1, \ldots, n\}$.

The major strength of using a Gaussian process regression to construct the surrogate, rather than a parametric model, is that the former much better reflects what is actually known based on the observations and incorporates uncertainty away from those observations. For example, a parametric model may be fitted to some arbitrary function of interest, $y(x)$, based on some samples of x taken in the range $0 < x < 100$. If this is used as a surrogate to evaluate $y(\alpha)$ for $\alpha >> 100$ a *value* of y will be directly returned – with no indication that the model has been extrapolated. Conversely if GPR was used to construct the surrogate, evaluating $y(\alpha)$ would return a *distribution* with a large covariance (due to $(x - \alpha)^2$ in (**??**) being large) reflecting low confidence in the expected value.

A second major strength of using GPR to construct a surrogate model is that the observations which comprise the training set are also random variables. This means that observations can be taken using variable fidelity analytical models, with observations from high fidelity models being passed in with a pdf closer to a delta function (i.e. very low covariance) and observations from lower fidelity models being passed in with a larger covariance to reflect the uncertainty attached to use of a less accurate model. This is very useful because the fidelity of the *OpenTidalFarm* 'inner' optimisation can be controlled by adjusting the convergence criteria of the gradient-based optimisation algorithm. To explain; a feature of gradient-based (or 'first-order') optimisation routines is that they operate greedily – i.e. an iteration only concludes if the functional has improved. In fact a graph of functional by iteration generally takes a form similar to logarithmic growth, so while a fully converged micro-siting optimisation may require 80 iterations (treating *OpenTidalFarm* as a black-box solver this would represent a *high* fidelity solution) if the optimisation were only allowed 10 iterations the functional

value will be lower than the fully converged value, but will be in the vicinity of it. This unconverged value represents the *low* fidelity solution.

4.2 Bayesian optimisation

The general procedure of Bayesian optimisation is that each time the experiment is run, a new data-point is sampled and this is added to the training set. The posterior is recomputed based upon the larger training set and the next data-point to sample is chosen based upon an *acquisition function* constructed from this surrogate model. The acquisition function chosen for this application is a function of the posterior mean and variance

$$\mathrm{acq}(N) = \boldsymbol{\mu}(N) + \kappa\boldsymbol{\sigma}(N). \tag{15}$$

where κ is a tuning parameter which can be chosen to either 'explore' or 'exploit' N. 'Exploring' means taking samples away from observations that have already been made – i.e. where the variance is largest – hence larger values of κ will induce this behaviour in the optimisation. 'Exploitation' means testing around the area of the expected optimum in greater detail and thus identifying it with tighter tolerance – in this mode $\kappa = 0$.

The strategy for this array design application is to start the optimisation in an exploration mode so as to identify the general shape of $\tilde{P}(N)$ – this will be done using low fidelity samples (i.e. limiting *OpenTidalFarm* to fewer iterations than required for full convergence). Once the the region of N that is likely to contain the optimum has been identified, κ is reduced and new sample points will be chosen so as to exploit this region and close in on the optimal solution.

One additional benefit of having access to the confidence bounds on estimated values is that a 'probabilistic branch and bound' method – as outlined in de Freitas et al. [6] – can be used to quickly identify feasible regions for the optimal solution. Recall that the purpose of the surrogate is not to recreate $\tilde{P}(N)$ accurately over the whole range N, but rather to represent $\tilde{P}(N)$ sufficiently well that the region containing the optimal solution is correctly identified (through the exploration phase) and then in this region ensure the representation of $\tilde{P}(N)$ is accurate. Using confidence bounds provided by the variance enables infeasible regions of the function to be quickly eliminated: Figure 5 has the 95 % confidence bounds plotted (as computed by 1.96σ from the mean). These can be used to discard regions in which the upper confidence bound is less than the maximum of the lower confidence bound, $\max(\mu - 1.96\sigma)$, as these regions are highly unlikely to contain the optimum [6].

5 Combined Model

To make it easier to refer to, the Bayesian optimisation routine as implemented is called *TidalArraySizer*. *TidalArraySizer* interfaces with *OpenTidalFarm* to optimise array size through a Bayesian optimisation approach with variable fidelity sampling and a probabilistic branch and bound method which identifies feasible regions for the optimal solution.

The optimisation begins in exploration mode ($\kappa = 1.96$) taking low fidelity samples at the bounds and midpoint of N. The geometry of the turbine area and the spacing constraints (for example minimum distance constraints) on the turbines determine the maximum number of turbines the site can support. The micro-siting is optimised for $N_{\max}$ to give an evaluation of the power output of the most populous array possible for the site, likewise the $N = 0,\ P = 0$ data point fully evaluates the training set bounds. The next data point is taken at $N_{\max}/2$ which completes initialisation of $\mathcal{D}$, the training set for the GPR.

A surrogate is built up based upon this initial training set, with low fidelity samples. Each time an additional N is sampled, the data is added to the training set and the surrogate is rebuilt and the next point for testing identified using the acquisition function, (4.2). Efficiency in the methodology is achieved by using low fidelity samples to identify the general trend of the function over the range of $[0, N_{\max}]$ during the initial exploration phase.

As the size of the training set increases, and the design space becomes more populated with (low fidelity) samples, there reaches a point where the range of array sizes for which the upper confidence bound exceeds the maximum lower confidence bound ceases to shrink with additional sampling. The rate of this shrinkage is used as a criterion for switching to exploitation mode, with high fidelity sampling.

When this stage is reached, *TidalArraySizer* switches to collecting high-fidelity samples - i.e. tightening the convergence criteria of the 'inner' *OpenTidalFarm* optimisation. If necessary, low fidelity samples can be restarted and allowed to converge and then are updated to high fidelity samples in $\mathbb{D}$ This is the 'exploitation' phase wherein the function has been sufficiently well explored to be certain the maximum is in the region being tested at high fidelity. At this point, the acquisition function κ weighting is set to zero and samples are taken at high fidelity.

At the conclusion of the optimisation, the optimum number of turbines is returned, along with their optimal micro-siting design on the site and a plot of the variable-fidelity surrogate of $\tilde{P}(N)$. Various detailed diagnostic and prognostic output data is also available (for example the flow velocity plots at each timestep or power output per turbine / per timestep, among many others). A flow chart of the combined model is shown in figure 5.

5.1 Financial model

In realistic turbine areas, limited by constraints such as allowing shipping lanes, channel blockage can often not reach a point where a clear maximum of $\tilde{P}(N)$ is found – rather the function tends to plateau as $N_{\max}$ is approached. In general, tidal array developers are interested in maximising the value of their investment, and diminishing returns on investment in additional turbines for an array runs contrary to this goal. While numerous measures of project finance exist (measures such as levelised cost of energy, LCOE, and net present value, NPV, are commonly used) the data upon which these models are based is generally proprietorial and the assumptions required to model a project over, say, a 25 year lifespan give the models a complexity which is beyond the scope of this work. Instead, a simplified financial model has been implemented which demonstrates the utility of the methodology. The modular, object-based design of the *TidalArraySizer/OpenTidalFarm* code allows users to substitute their own financial models to run their own analyses. A similar idealised approach to financial modelling was used in Culley et al. [5] to demonstrate optimisation of micro-siting design to

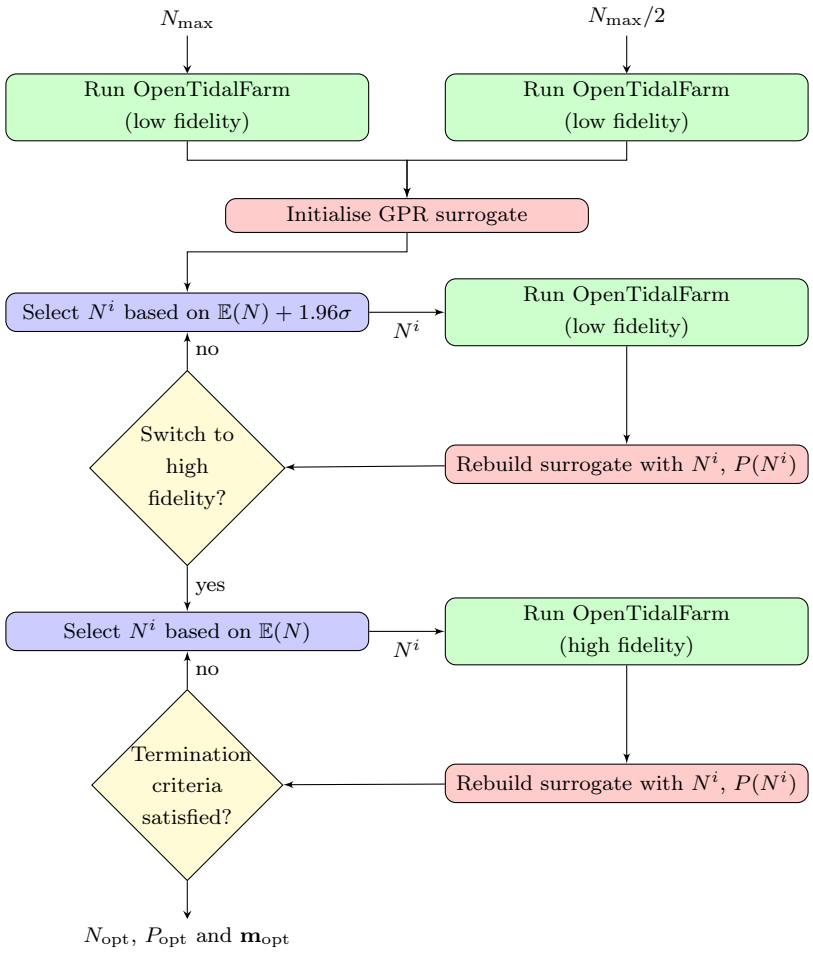


Figure 3: Flow chart showing the ensemble *TidalArraySizer* model.

maximise profit as computed by the income from the power extracted by the turbines minus the cost of sub-sea electrical cabling. For that work, detailed financial data was likewise unavailable – so simplifying assumptions were made to demonstrate the utility of the methodology.

For this work, a notional measure of profit will used which is computed by subtracting the lifetime cost of the array from the lifetime income generated by the power production of the array. It is assumed that the cost of the array is solely a function of the number of turbines multiplied by a constant cost per turbine. A cost per turbine, of £4 million has been chosen to reflect the capital expenditure on the turbine itself. In addition to a per-turbine contribution to support-infrastructure (cabling, electrical conditioning infrastructure etc.), installation and maintenance of an additional £4 million has been added giving a total cost of £8 million per turbine.

Modelling the income generated from power production over the array lifetime is much more difficult as it depends not only on the performance of the array but also on the energy market and governmental support schemes which can vary enormously over 25 years. Consequently the income will be computed as the idealised income per unit of

energy multiplied by the power extracted by the array over the lifetime of the project. The idealised income per unit extracted, $I^{\text{idealised}}_{\text{per unit}}$ is computed from the price per unit of energy that is necessary for an ideal turbine to break even over its life. For this, we measure the energy extracted by a single 1 MW turbine in a wide simple channel with a sinusoidally head-driven inflow which varies between $\pm$ 3 ms^{-1} over a 12.5 hour tidal cycle. Multiplying by the design life, $N^{\text{design}}_{\text{tidal cycles}} = 43,800$, gives the energy extracted by that turbine over its life and dividing the cost of the turbine $C_{\text{turbine}} = £8$ million by this value gives the idealised income per unit of energy, $I^{\text{idealised}}_{\text{per unit}}$. The energy extracted by the array over it's lifetime is found by choosing a representative 12.5 hour cycle in which the average power extraction corresponds to the average over the longer period (running the optimisation based on a longer model time would become too computationally expensive). By measuring the energy extracted over this period and multiplying by the number of tidal cycles in the array lifetime and the idealised income per unit energy we arrive at the lifetime income of the turbine array.

$$\text{Profit}(N) = \text{Income} - \text{Cost}, \tag{16}$$

where

$$\text{Cost} = N \times C_{\text{turbine}}, \tag{17}$$

$$\text{Income} = I^{\text{idealised}}_{\text{per unit}} \times E^{\text{cycle}}_{\text{array}} \times N^{\text{design}}_{\text{tidal cycles}}, \tag{18}$$

and

$$I^{\text{idealised}}_{\text{per unit}} = \frac{C_{\text{turbine}}}{E^{\text{ideal cycle}}_{\text{turbine}} \times N^{\text{design}}_{\text{tidal cycles}}}. \tag{19}$$

Clearly this a heavily simplified financial model whose suitability only extends to proof of concept. When realistic, publishable data is available to improve upon this, future work will look at the impact of a less idealised model. In addition, greater accuracy could be achieved by modelling for longer than a single tidal cycle. Recall that this model is designed to produce an optimal array size and micro-siting design and an idea of the power output for resource assessment purposes. Once an optimised design has been returned more accurate, higher fidelity hydrodynamic and financial models may be used to produce more reliable power and financial return estimates for the array, modelled over a longer period of time.

One of the key challenges of this work is that it lies at the intersection of engineering design and economics. A traditional, non-renewable, power plant's capacity will be sized as a result of an economic analysis in which the relationship between plant cost and power output are well understood from industrial experience. This means that the economic analysis and engineering design can proceed without strongly interacting. Contrastingly, in tidal array design the economics of the installation is strongly dependent upon it's detailed design – for example the turbine micrositing. This means that the economic and technical analyses of a proposed project are highly coupled. This work represents a pragmatic first step toward that coupling – compromises in the fidelity of technical and economic modelling have been made because optimisation is an inherently expensive exercise and engineers have a finite computational budget. However, improvements in the technical and economic modelling of coupled approaches are a subject of future work. Such improvements would likely affect the optimisation approaches that have been employed in this work – for example discontinuities in cost models could drastically alter the profit versus number of turbines response surface.

6 Idealised example

The framework will first be demonstrated on a highly idealised example problem. The domain is a 2560 by 1280 m simple channel with a circular island, 320 metres in diameter located at its centre. The turbine area measures 640 by 320 metres and is centred longitudinally in the channel and in the middle of the sub-channel to the south of the island. The element size for the (unstructured) finite-element mesh varies from 5 metres in the turbine area to 25 metres furthest from it (figure 4). The depth is a constant 50 metres throughout the domain.

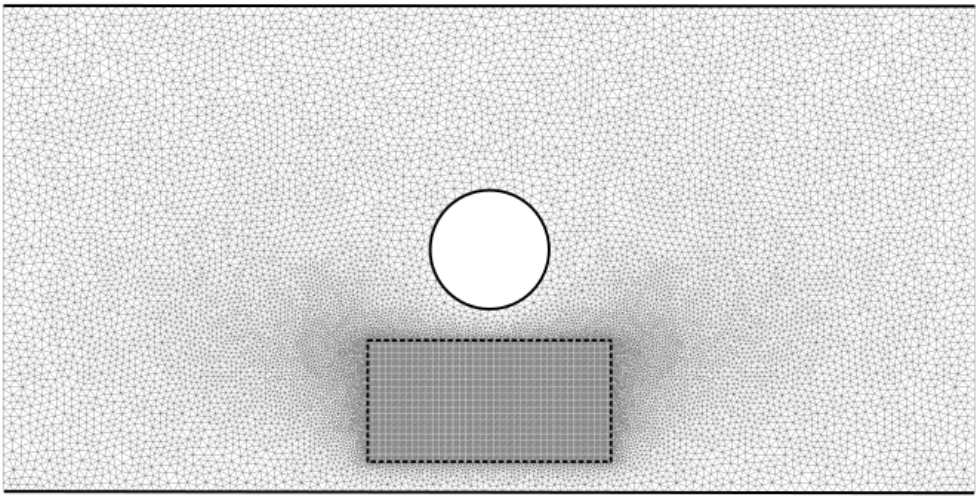


Figure 4: Unstructured computational mesh of the simple channel with circular island test example

Flow is driven by a sinusoidal velocity-forcing ranging from ± 2 ms^{-1} at the inflow boundary over a period of 12.5 hours, divided into 15 time-steps. The viscosity was set to 2 m^{2}s^{-1} and the turbines were parametrised as per (10) and figure 1.

The surrogate is initialised with $N = 0$, $P = 0$ and low fidelity optimisation runs of $N_{\max}$ and $N_{\max}/2$, where $N_{\max}$ is the maximum numbers of turbines that can fit on the site (assuming a minimum centre-to-centre spacing of 30 metres $N_{\max} = 21 \times 10 = 210$). For this example, the low fidelity standard was set by an optimisation tolerance of 1% – which for most samples represented an optimisation requiring circa 15 iterations (that is 15 adjoint solves and typically a similar number of shallow water solves). The initialised surrogate model is shown in figure 5.

In the low fidelity phase of the optimisation, the acquisition function is defined as (4.2) with $\kappa = 1.96$. Low fidelity samples continued to be taken until the range of the region in which the upper confidence bound is greater than the maximum lower confidence bound (shown where the dashed upper confidence bound exceeds the dotted maximum lower confidence bound line in figures 5 and 6) stopped decreasing. Figure 6 shows the surrogate model of the objective function once the algorithm has completed collecting low fidelity samples.

Now the $\kappa = 0$ acquisition function selects sample points based on the maximum of the surrogate prediction. Low fidelity runs can be restarted and allowed to converge to high fidelity. Low fidelity sample points can be clearly identified in the finished

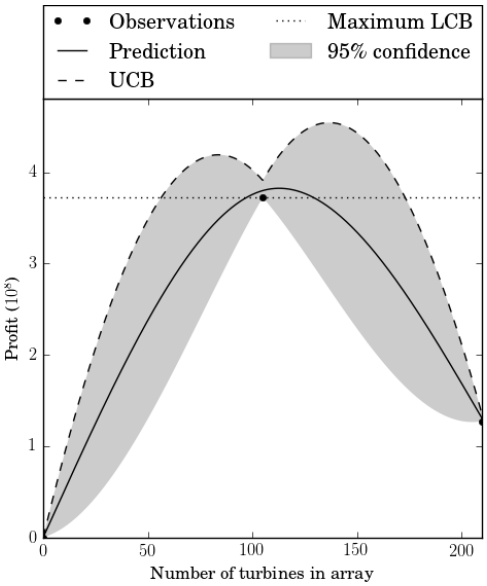


Figure 5: Plot of the Profit(N) surrogate initialised with training set data points at Profit($N = 0$), Profit($N_{\max}$), Profit($N_{\max}/2$).

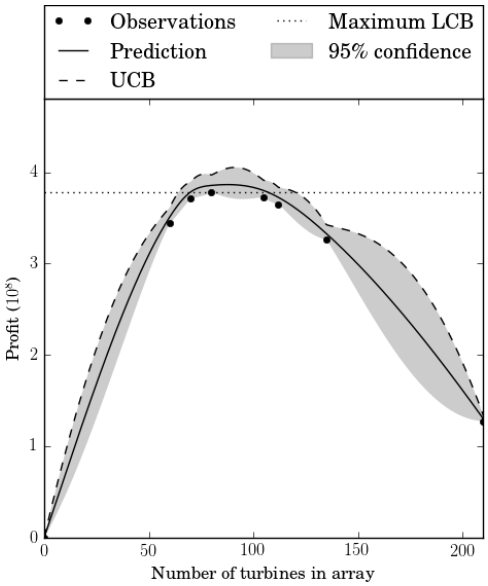


Figure 6: Plot of the Profit(N) surrogate before switching to high fidelity sampling.

surrogate shown in figure 7 where the upper confidence bound does not fully 'pinch' in at a data point.

The finished surrogate can be seen in figure 7, along with the final turbine layout in figures 8 and 9. It can be seen that the optimisation was achieved in four low fidelity and five high fidelity *OpenTidalFarm* calls. In this instance an optimum number of 80 turbines (± 5 turbines) is returned as the optimum array size, this is less than half of the maximum number the site can support given the minimum distance constraint imposed between turbines.

The turbine layout is significantly different from Funke et al. [9] which shows a similar problem, though on a smaller domain and without any power curve (in Funke et al. [9] turbines are not realistically parametrised, rather Power $\propto \mathbf{u}^3$ and is uncapped). At the up and downstream ends of the turbine area, the turbines are lined up in a clearly recognisable barrage formation. Between these two barrages the layout seems less coherent. Rather than large scale barrages, turbines are grouped in sets of 3 to 5 turbines which are staggered in diagonal formations, allowing downstream turbines to capitalise on the local flow acceleration around the sides of the upstream turbine. Turbines along the north and south turbine area boundaries capture energy as the streamlines diverge while the flow moves through the turbine area, but also as flow within the turbine area is re-energised through shear effects from the bypass flow. This layout has an represents an 11 % improvement on the starting layout of a regular grid formation, the energy extracted over the tidal cycle for the optimised array was 2.458 TJ.

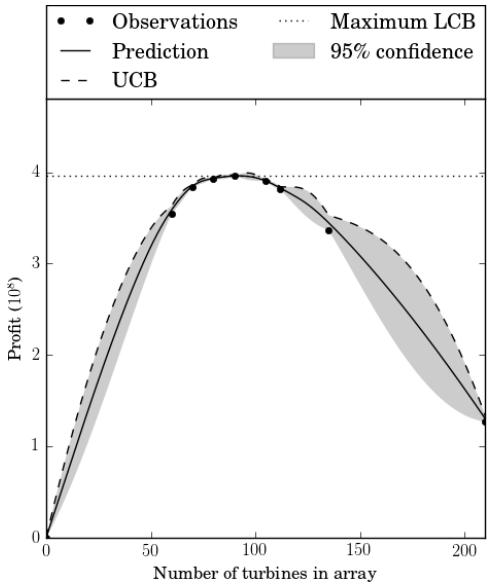


Figure 7: GPR surrogate at end of optimisation.

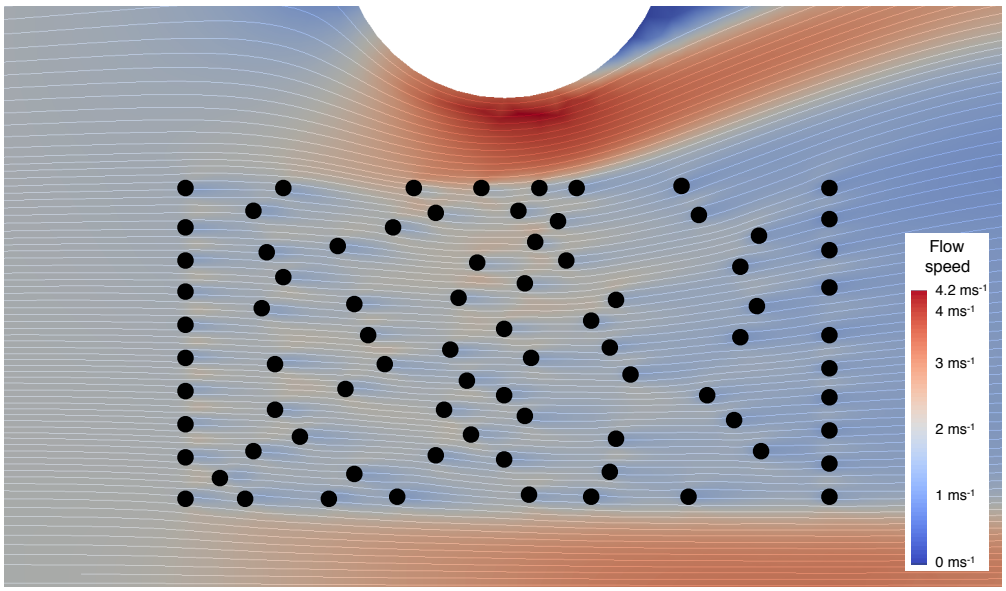


Figure 8: Optimised location of the optimal number of turbines for the site showing flow velocity magnitude and streamlines for 'flood' tide (flow left to right).

7 Pentland Firth example

480 This example uses a domain based on the Pentland Firth, a site between the Scottish mainland and the Orkney Islands. This is one of the prime sites being considered for deployment of tidal stream turbine arrays and the turbine area used in this example is based on the phase 1 deployment site for which the developer *MeyGen* has obtained consent to develop an 86 MW array. A simplified version of this example on a smaller, 485 flat-bottomed analogue of this domain was considered in Culley et al. [4].

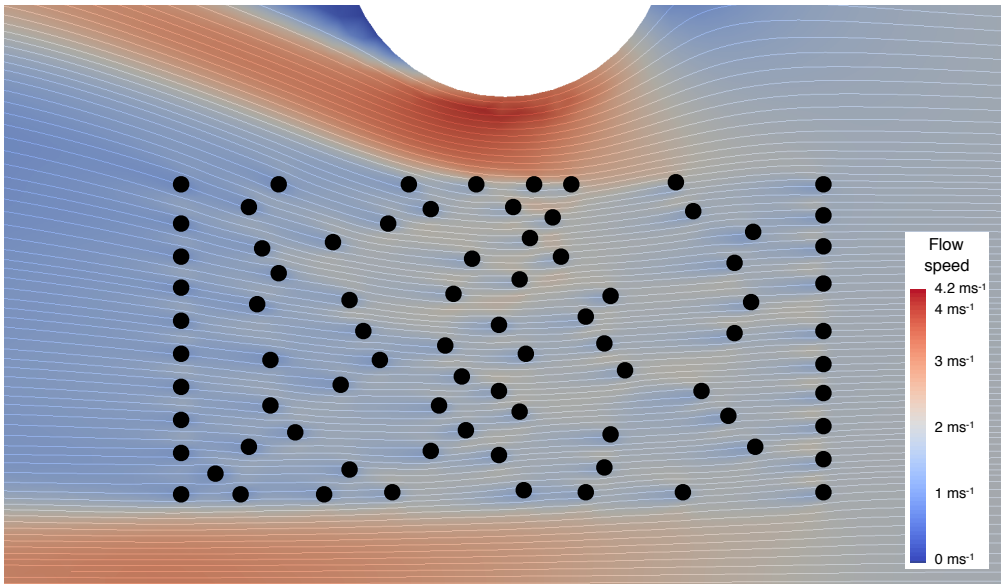


Figure 9: Optimised location of the optimal number of turbines for the site showing flow velocity magnitude and streamlines for 'flood' tide (flow left to right).

The unstructured mesh is shown in figure 10, the element size ranges from 5 metres in the turbine area, to 40 metres along the coastline and islands to a maximum of 400 metres away from these features. The mesh was generated by the open-source meshing software Gmsh [12]. The domain is bounded to the south by a portion of the Scottish mainland coast from Dunnet Head to just round Duncansby Head. This is artificially extended out to the eastern open boundary, which is situated a few kilometres east of the Skerry Islands. The western open boundary stretches from Dunnet Head to the nearest point on the island Hoy. The northern boundary is modelled (as are the coastlines and island boundaries) with a zero velocity Dirichlet condition – i.e. no-slip and no-normal flow. Clearly this is a simplification as it closes off the Pentland Firth's connection to Scapa Flow through the waterways between Hoy, Flotta and South Ronaldsay. This simplification was made to keep the problem size tractable on a single HPC node of 12 cores.

The domain is based on a simplification of the validated set-up used by Funke et al. [11]. Bathymetry information is provided by combining data from four sources (as outlined in Funke et al. [11]). Flow is head-driven with the head difference across the eastern and western boundary calculated using the OSU Tidal Prediction Software [7]. The model simulates one tidal cycle starting at 10:40 on 18-09-2001 using the Q1, O1, P1, K1, N2, M2, S2 and K2 tidal constituents. The start of the simulation time was chosen as the head difference is close to being zero using these constituents at this time.

A period of 12 hours is used, subdivided into 40 time steps. The computational expense of a solve at a given iteration is proportional to the number of time steps so with additional computational resource the number of timesteps could be increased. The viscosity varies over the domain with proximity to the open boundaries to aid stability. At the turbine site, the viscosity is 10 m^2s^{-1}, a higher viscosity of 1000 m^2s^{-1} was used in a narrow region (1 km) near the boundaries. Once again, the turbines were parametrised as per (10) and figure 1. The turbine area is 1000 by 600

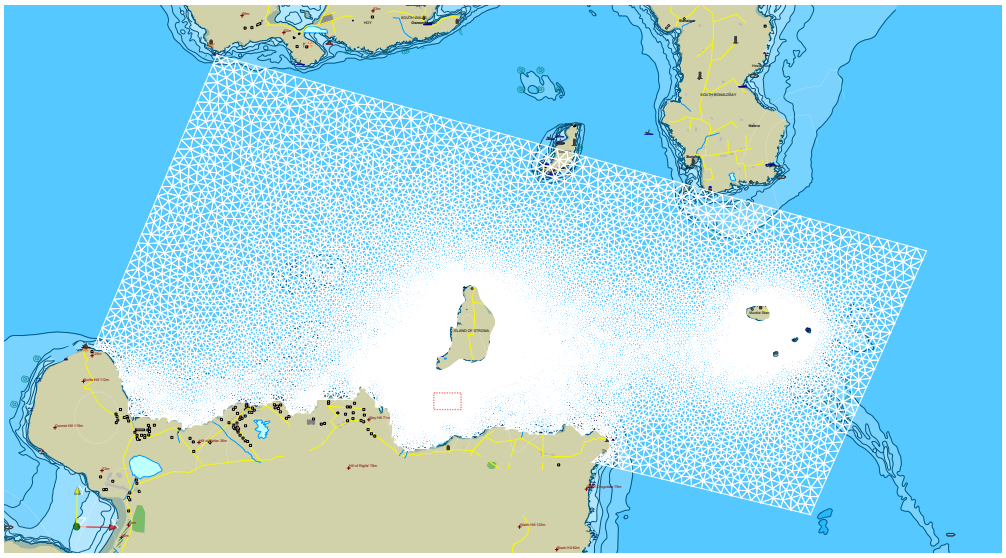


Figure 10: Map of Pentland Firth with unstructured mesh superimposed in white

metres in size and located to the south of Stroma in the Inner Sound of the Pentland Firth, as shown in figure 10 by a dashed red box.

As before, the surrogate was initialised with $N = 0$, $P = 0$ and low fidelity optimisation runs of $N_{\max}$ and $N_{\max}/2$, where again $N_{\max}$ is the maximum numbers of turbines that can fit on the site (assuming the same minimum centre-to-centre spacing as before $N_{\max} = 33 \times 20 = 660$).

Once again, low fidelity samples were taken over the range of $0 \leq N \leq N_{\max}$ from which the surrogate shown in figures 11 and 12 was constructed.

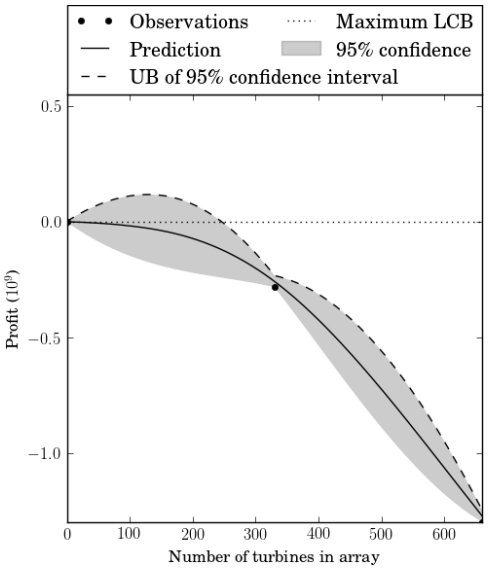


Figure 11: Surrogate model initialised with starting training data.

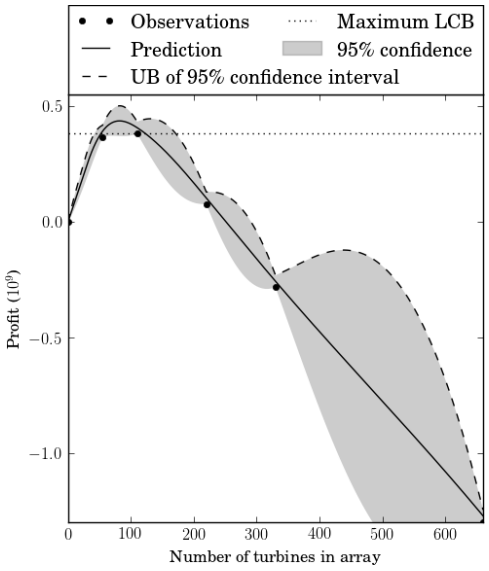


Figure 12: Surrogate model at conclusion of low fidelity runs.

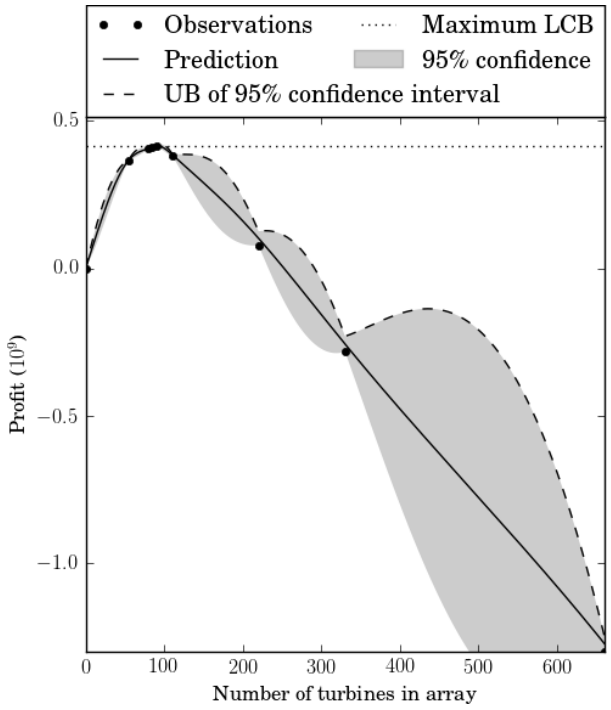


Figure 13: Final surrogate model of N to P mapping for the Inner Sound example.

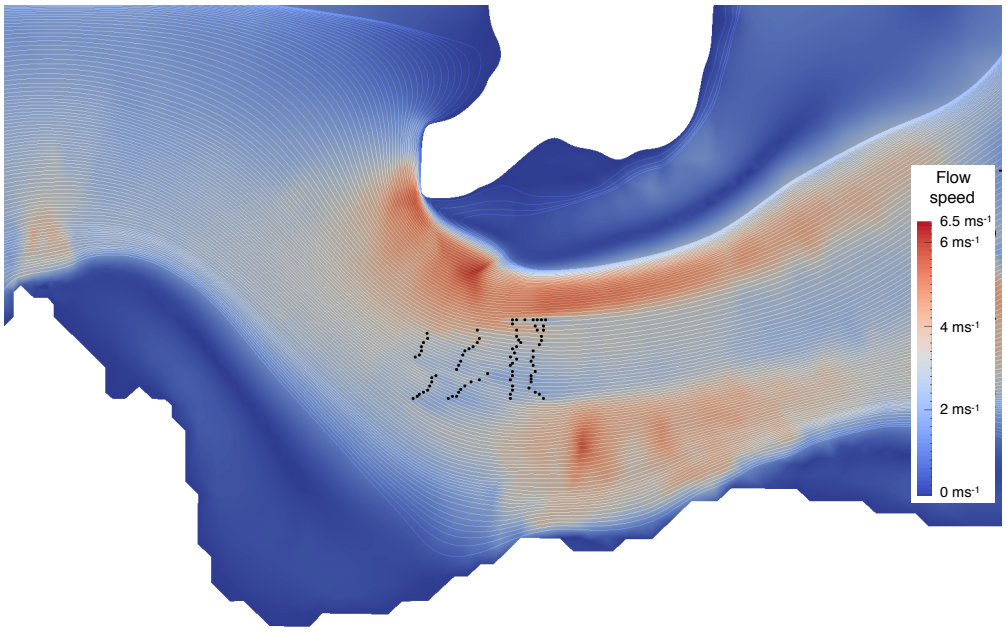


Figure 14: Optimised location of the optimal number of turbines for the site showing flow velocity magnitude and streamlines for 'flood' tide (flow left to right).

In contrast to the results for the example in section 6, in this case, the optimised turbine micro-siting design for the optimal number of turbines is much more ordered (see figure 15), exhibiting several clearly defined rows of turbines. To the east of the

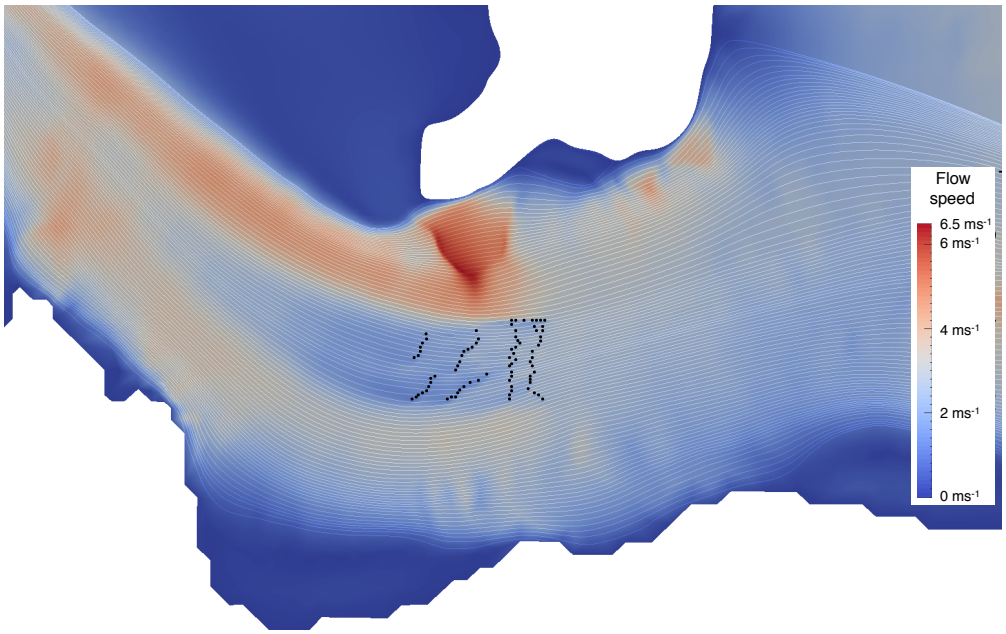


Figure 15: Optimised location of the optimal number of turbines for the site showing flow velocity magnitude and streamlines for 'ebb' tide (flow right to left).

turbine area, two well ordered rows of turbines have emerged, within these rows there is some offsetting of one turbine to the next. This staggering is, once again, likely to exploit local blockage induced flow accelerations around the sides of turbines. One reason for this may be that the previous example was driven by a constant velocity in a comparatively small domain, while this example is driven by a head difference over the domain. This means that (in the previous example) flow is forced through the site and bulk flow redirection to the north of the island is a less important mechanism. The result of this is that the site can support more turbines and the site is much more densely populated than in this Pentland Firth example. With more densely packed sites, the interactions between the turbines over the tidal cycle are that much more intricate and chaotic – leading to a less well ordered arrangement of turbines.

To the west of the turbine area, several smaller barriers have formed, which lie broadly perpendicular to flow entering the turbine area during the flood phase, but also perpendicular to flow leaving the turbine area during the ebb phase. Acceleration of the flow around the sides of these mini-barrages is exploited (where the position of turbine area boundaries allows) by downstream barrages.

The proposed array consists of 90 turbines, which in the optimised layout extract 32.59 TJ over the tidal cycle. This number of turbines is broadly consistent with the array size proposed by MeyGen for phase one of their development of the Inner Sound – on which this example is based. Clearly this example is based upon simplified modelling assumptions (for example the idealised financial model), however these assumptions are grounded in reality and the similarity is encouraging.

8 Conclusions

In this paper, a methodology for the optimisation of tidal turbine array design has been presented which considers both the number of turbines in the array (array size) and the locations of those turbines relative to one another and the domain (micro-siting design). Both of these design parameters are strongly coupled to the regional and local hydrodynamics to which the power extracted by the array is highly sensitive (and thus are any prognostic variables based on it).

Previous work has demonstrated the strong impact turbine micro-siting design can have on the performance of the array, and has introduced *OpenTidalFarm* as a tool to maximise some desirable quantity through 'tuning' the array to the site through manipulation of turbine locations. This approach has shown utility whether the array performance is measured in terms of the energy extracted by the array [9], the financial performance of the array [5], or indeed by some other barometer. However, until a given array has been tuned to the site, the designer does not know how that array will perform (because of the strong coupling of hydrodynamics to design parameters and the high sensitivity of power to them), i.e. for a given array size the performance of that array is unknown until its micro-siting design has been optimised. Consequently, for a given site and conditions, finding the array size which maximises the performance indicator is an 'outer' optimisation for which each sample point investigated requires an 'inner' micro-siting optimisation provided by *OpenTidalFarm*.

Due to the computational cost of performing a full micro-siting optimisation at each array size sample point taken, a methodology suited to the optimisation of expensive black-box functions is required – where the micro-siting optimisation performed by *OpenTidalFarm* is treated as the black-box – an unknown non-linear solver which for a given array size determines the optimal performance of that array and the micro-siting design which achieves this. This work demonstrates the utility of a surrogate-based approach which uses a Bayesian methodology combining prior-assumed knowledge of the function of interest with a training set of recorded observations, to achieve a posterior expectation of how unobserved array sizes may be capable of performing. This approach minimises the number of observations which are required by using Gaussian process regression to maximise the utility of the observations.

This approach was used to optimise the array size and micro-siting design for two test examples; the first being a simple channel with a centrally located circular island and the second being loosely based on the Inner Sound of the Pentland Firth, in the Scottish Orkney Islands. In both cases, turbines were parametrised with realistic thrust and power curves and optimised for a full flood-ebb cycle. In the first case, the array size was optimised in four low fidelity and five high fidelity *OpenTidalFarm* runs, an array size of 80 turbines was returned as optimal to maximise the financial profitability of the project, and an array design of those turbines was proposed to maximise the power extracted by the array. In the Inner Sound example, the array size was likewise optimised for financial profitability and in four low and five high fidelity *OpenTidalFarm* runs an optimal array size of 90 turbines was found.

9 Acknowledgements

OpenTidalFarm is an open source project that can be freely downloaded from *opentidalfarm.org* and used under the GNU-GPL version 3 license.

The authors would like to acknowledge Imperial College London, the Engineering and Physical Sciences Research Council (projects: EP/J010065/1, EP/L000407/1 and EP/M011054/1) and a Center of Excellence grant from the Research Council of Norway to the Center for Biomedical Computing at Simula Research Laboratory for funding which supported this work. Finally, the authors would like to acknowledge helpful discussions with T. Griffiths and B. Shuttleworth.

Bibliography

[1] G.L. Barnett, S.W. Funke, and M.D. Piggott. Hybrid global-local optimisation algorithms for the layout design of tidal turbine arrays. *Submitted to Renewable Energy*, 2014.

[2] I.G. Bryden and S.J. Couch. ME1: marine energy extraction: tidal resource analysis. *Renewable Energy*, 31:133–139, February 2006.

[3] D.M. Culley, S.W. Funke, S.C. Kramer, and M.D. Piggott. A hierarchy of approaches for the design of tidal turbine arrays. *5th International Conference on Ocean Energy, ICOE, Halifax*, November 2014.

[4] D.M. Culley, S.W. Funke, S.C. Kramer, and M.D. Piggott. Tidal stream resource assessment through the optimisation of array design with quantification of uncertainty. *Proceedings of the 11th European Wave and Tidal Energy Conference, EWTEC, Nantes*, 2015.

[5] D.M. Culley, S.W. Funke, S.C. Kramer, and M.D. Piggott. Integration of cost modelling within the micro-siting design optimisation of tidal turbine arrays. *Renewable Energy*, 85(0):215 – 227, 2016. doi: 10.1016/j.renene.2015.06.13.

[6] N. de Freitas, A. Smola, and M. Zoghi. Regret bounds for deterministic gaussian process bandits. *arXiv preprint arXiv:1203.2177*, 2012.

[7] G. Egbert and L. Erofeeva. OSU Tidal Prediction Software (OTPS). *http://volov.oce.orst.edu/tides/otps.html*, 2010.

[8] S.W. Funke and P.E. Farrell. A framework for automated PDE-constrained optimisation. *arXiv:1302.3894*, 2013.

[9] S.W. Funke, P.E. Farrell, and M.D. Piggott. Tidal turbine array optimisation using the adjoint approach. *Renewable Energy*, 63:657 – 673, 2014.

[10] S.W. Funke, D.M. Culley, S.C. Kramer, and M.D. Piggott. OpenTidalFarm: A software package for the optimisation of tidal turbine arrays. *In preparation*, 2015.

[11] S.W. Funke, S.C. Kramer, and M.D. Piggott. Design optimisation and resource assessment for tidal-stream renewable energy farms using a new continuous turbine approach. *submitted to Renewable Energy*, 2015. doi: http://arxiv.org/abs/1507.05795.

[12] C. Geuzaine and J.F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[13] M.D. Gunzburger. *Perspectives in flow control and optimization*, volume 5. Siam, 2003.

[14] D.R. Jones, M. Schonlau, and W.J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[15] A. Keane and P. Nair. *Computational approaches for aerospace design: the pursuit of excellence*. John Wiley & Sons, 2005.

[16] S.C. Kramer and M.D. Piggott. A correction to the enhanced bottom drag parameterisation of tidal turbines. *Submitted to Renewable Energy*, 2015.

[17] R. Malki, I. Masters, A.J. Williams, and T.N. Croft. Planning tidal stream turbine array layouts using a coupled blade element momentum–computational fluid dynamics model. *Renewable Energy*, 63:46–54, 2014.

[18] R. Martin-Short, J. Hill, S.C. Kramer, A. Avdis, P.A. Allison, and M.D. Piggott. Tidal resource extraction in the Pentland Firth, UK: Potential impacts on flow regime and sediment transport in the Inner Sound of Stroma. *Renewable Energy*, 76:596 – 607, 2015. doi: 10.1016/j.renene.2014.11.079.

[19] R. Pinnau and M. Ulbrich. *Optimization with PDE constraints*, volume 23. Springer Science & Business Media, 2008.

[20] C.E. Rasmussen. Gaussian processes for machine learning. 2006.

[21] T. Roc, D.C. Conley, and D. Greaves. Methodology for tidal turbine representation in ocean circulation model. *Renewable Energy*, 51:448–464, March 2013.

[22] J.L. Steward, I.M. Navon, M. Zupanski, and N. Karmitsa. Impact of non-smooth observation operators on variational and sequential data assimilation for a limited-area shallow-water equation model. *Quarterly Journal of the Royal Meteorological Society*, 138(663):323–339, 2012.

[23] P. Sullivan and P. McCombie. Optimisation of tidal power arrays using a genetic algorithm. *Proceedings of the Institution of Civil Engineers-Energy*, 166(1):19–28, 2013.

[24] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers & Fluids*, 1(1):73–100, 1973.

[25] M.D. Thomson, J.I. Whelan, and L. Gill. The development of a tool for the design and optimisation of tidal stream turbine arrays. In *Proceedings of the European Wave and Tidal Energy Conference*, 2011.

[26] R. Vennell. Tuning tidal turbines in-concert to maximise farm efficiency. *Journal of Fluid Mechanics*, 671:587–604, 2011.

[27] R. Vennell, S.W. Funke, S. Draper, C. Stevens, and T. Divett. Designing large arrays of tidal turbines: A synthesis and review. *Renewable and Sustainable Energy Reviews*, 41:454–472, 2015.

[28] Z. Zhou, Y.S. Ong, and P.B. Nair. Hierarchical surrogate-assisted evolutionary optimization framework. In *Evolutionary Computation, 2004. CEC2004. Congress on*, volume 2, pages 1586–1593. IEEE, 2004.