Optimisation of Tidal Turbine Arrays through the use of Surrogate Modelling and the ParEGO Optimisation Algorithm

Ben Ashby - Summer Research Project 2015 – funded through the EPSRC Vacation Bursary Scheme

Project Supervisors: Professor Gavin Tabor and Dr. Jonathan Fieldsend

Introduction and Project Aims

Tidal Turbines, and arrays of such turbines, represent a promising means for the generation of renewable electricity in the UK. In order to design installations of this type a means for the optimisation of array characteristics is needed to minimise the cost of energy generation. When designing arrays of Tidal Turbines a key aim is to maximise power output whilst minimising array area, which is linked to the overall installation cost. This work seeks to identify a new methodology for the optimisation of an array of Tidal Turbines through the use of Computational Fluid Dynamics (CFD) and Evolutionary Algorithms (EA). An actuator disk model was used to simulate turbine behaviour in the open-source CFD code OpenFOAM, and an array of these disks was optimised through the use of the ParEGO optimisation algorithm.

For more information on this work please contact Ben Ashby: ba258@exeter.ac.uk

Actuator Disk Modelling of Tidal Turbines in OpenFOAM

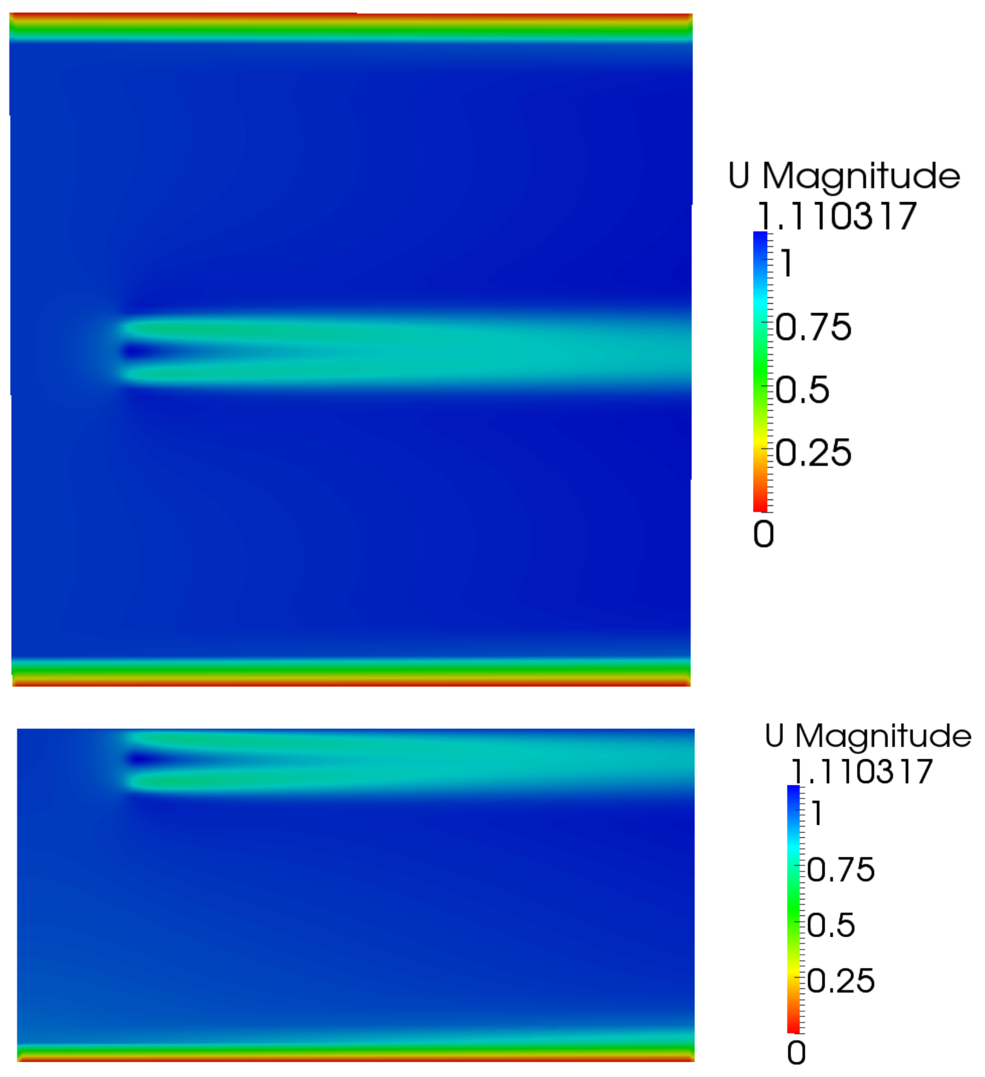

Figure 1: plan and side views of wake structure for the actuator disk model used

The model used throughout this work was a variant on the actuator disk. This model was first implemented by Svenning (2010) and has been further developed for use with multiple turbines as a part of this project. The wake structure behind turbines is recreated in the flow through the use of volume forces which act across the disk area, and are distributed according to the Goldstein Optimum. These volume forces act against the flow and introduce swirl into the wake structure.

For a more detailed description of this model please see (Svenning, 2010).

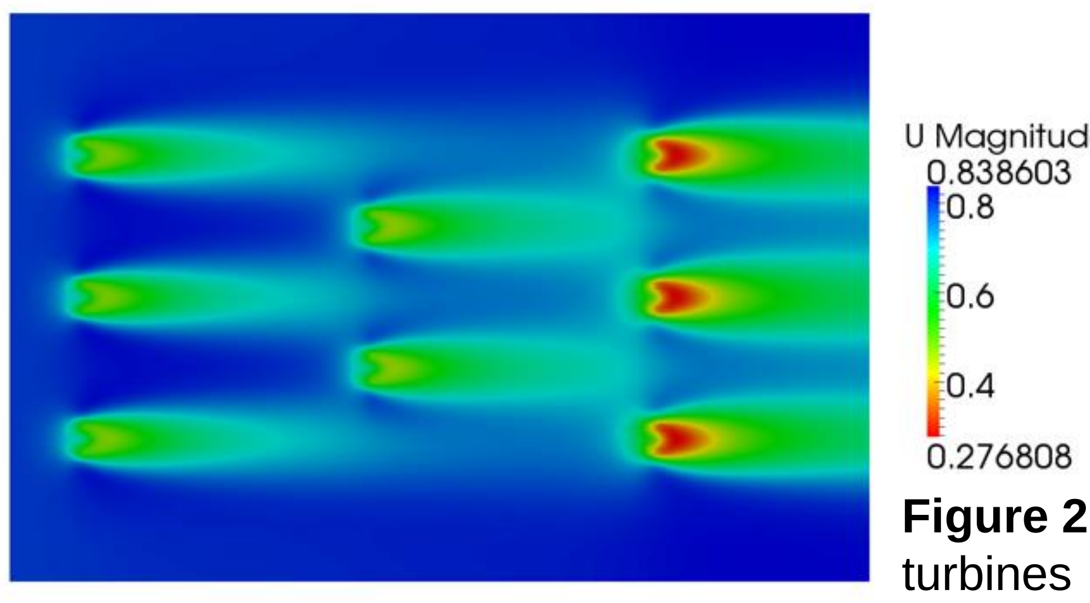

Figure 2: velocity contours through an array of tidal turbines

Power abstracted from the flow by the model was found by applying the conservation of linear momentum across individual control volumes surrounding each actuator disk.

Methodology

The domain used for the CFD modelling was set up as shown in **Figure 3**, which represents a section of an array with three rows of turbines. Symmetry conditions are applied at the side walls in order to shrink the domain and reduce the computational cost. A Latin Hypercube sample of 30 cases was set up and run, varying the stream-wise spacing between each row of turbines, lateral spacing between turbines, and power abstraction for each row of turbines.

This initial sample of 30 cases was used to create a surrogate model using a Kriging approach. This model provided an inexpensive statistical model to the original CFD model.

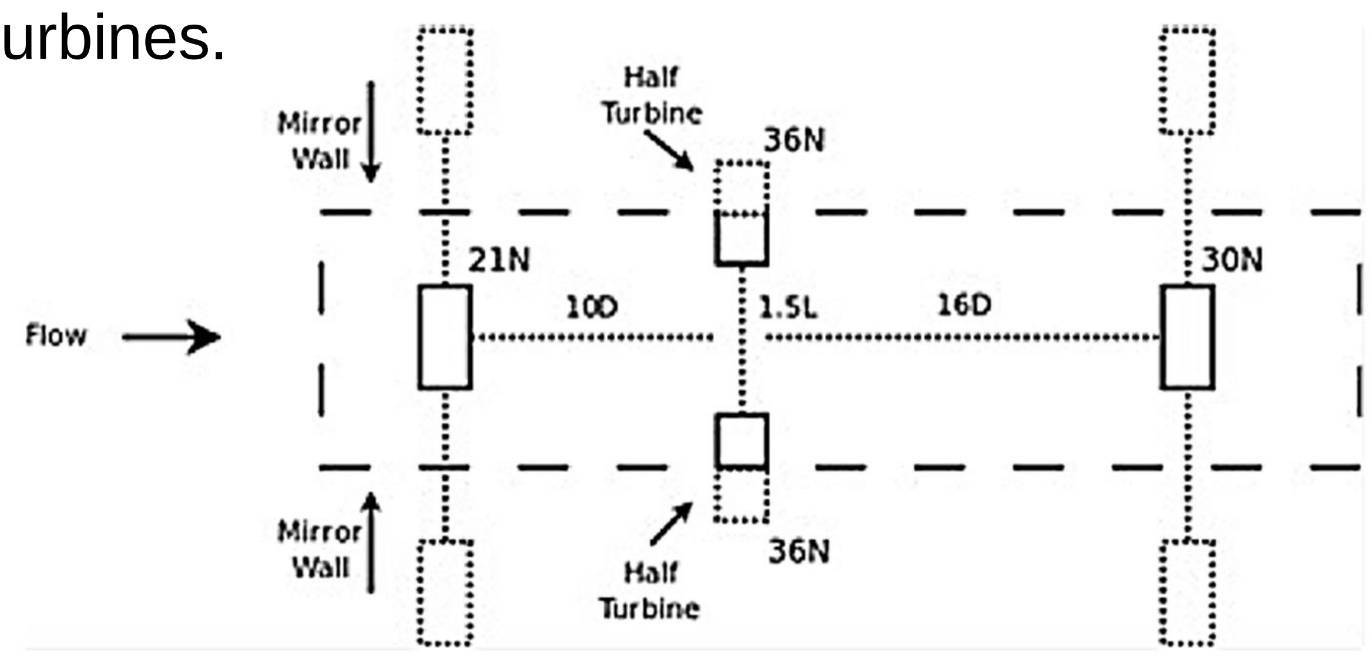

Figure 3: array layout used for optimisation study

The value returned by the surrogate model for a set of inputs, combined with the uncertainty of prediction at this point, could be used to find the Expected Improvement on the previous optimal solution. Once the ParEGO algorithm had been used to maximise the Expected Improvement, this optimal solution was then run on the original CFD model. Two implementations of the ParEGO algorithm were explored in this manner, traditional ParEGO where a single optimal solution is outputted and tested at each iteration, and a parallel version where 5 optimal solutions were run on the CFD model at each iteration, each simulation exploring a different region of the Pareto optimal curve.

The ParEGO Optimisation Algorithm

ParEGO is a multi-objective optimisation algorithm which incorporates surrogate modelling with an internal evolutionary algorithm to produce a Pareto front of non-dominated solutions (Knowles, 2005). At each iteration, the algorithm constructs a surrogate model using the previous solutions to the problem, and uses a genetic algorithm to find a solution that maximises the Expected Improvement. This optimal solution is then run on the original model and the solution set updated.

The algorithm uses an aggregation function to combine the multiple objectives to a single scalar value for the solution set. This function uses a weighting vector to compute the single output value, and this weighting is changed at each iteration to ensure that all areas of the Pareto front are explored.

Results and Conclusions

Upon running both forms of the ParEGO algorithm, Pareto curves were seen to form at the top left of the solution space. It was shown that when running ParEGO in parallel the time taken to reach a good Pareto front is reduced dramatically when compared to traditional ParEGO for highly expensive cost functions such as CFD simulations.

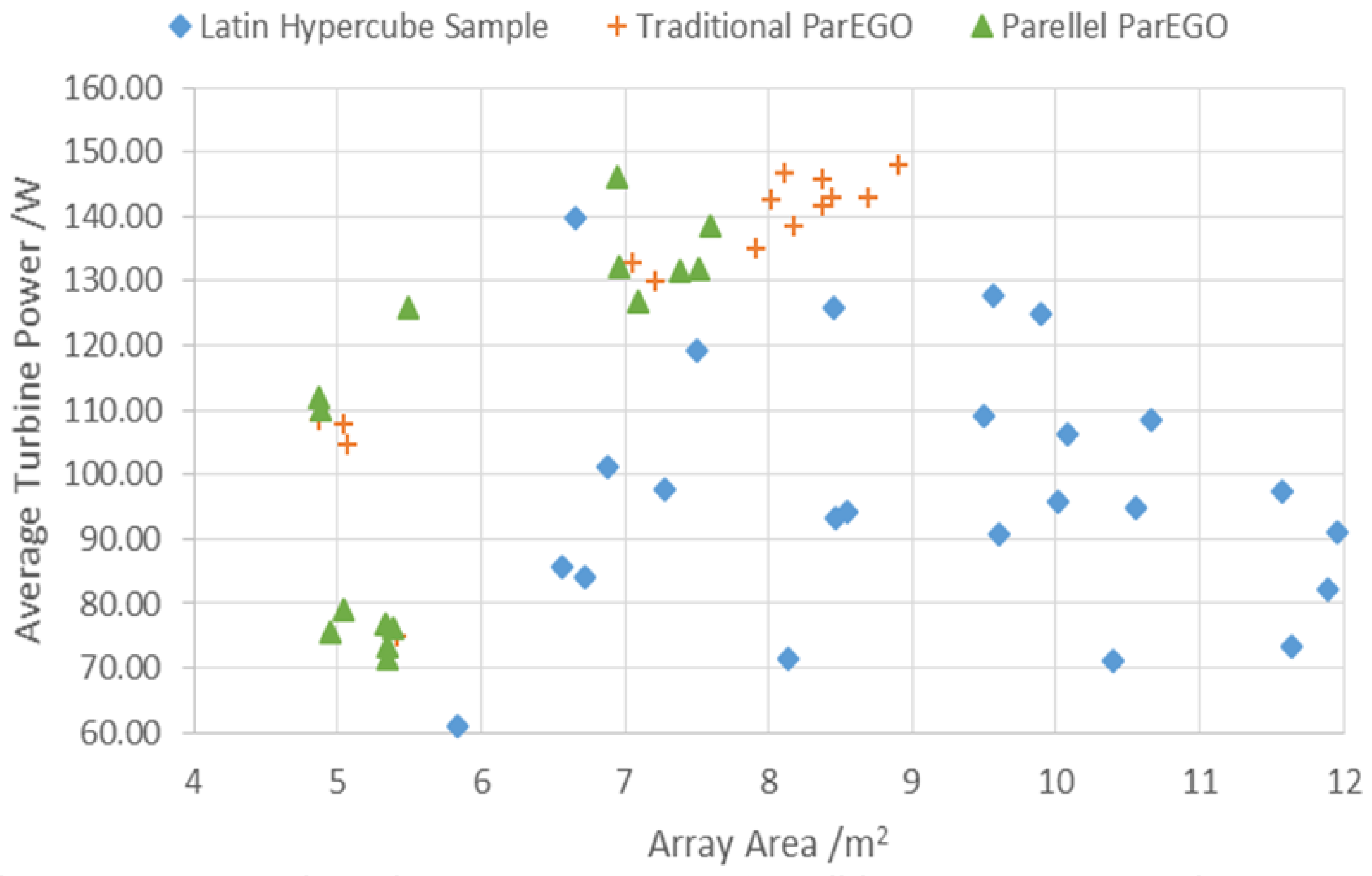

Figure 4: comparison between outputs of traditional ParEGO and a parallel implementation after 15 cases were run on the CFD model

A good Pareto set was achieved from the parallel implementation of ParEGO after just 6 iterations, and although there are areas of the Pareto front where there are few solutions, the algorithm would better explore these regions if it were allowed to run further.

The optimisation process could be streamlined through the automation of the generation of new CFD cases at each iteration, with the use of the PyFoam library in OpenFOAM presenting one possibility for achieving this.

Bibliography

Knowles, J., 2005. ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multi-objective Optimization Problems. IEEE Transactions On Evolutionary Computation, 10(1), pp. 50-66.

Svenning, E., 2010. Implementation of an actuator disk in OpenFOAM, Gothenburg: Chalmers University of Technology.

UNIVERSITY OF EXETER

College of Engineering, Mathematics and Physical Sciences