CFD based draft tube hydraulic design optimization

J McNabb¹, C Devals², S A Kyriacou³, N Murry¹ and B F Mullins¹

¹ Andritz Hydro, Pointe-Claire, Québec, Canada
² École Polytechnique, Montréal, Québec, Canada
³ Andritz Hydro, Zürich, Switzerland

E-mail: jacques.mcnabb@andritz.com

Abstract. The draft tube design of a hydraulic turbine, particularly in low to medium head applications, plays an important role in determining the efficiency and power characteristics of the overall machine, since an important proportion of the available energy, being in kinetic form leaving the runner, needs to be recovered by the draft tube into static head. For large units, these efficiency and power characteristics can equate to large sums of money when considering the anticipated selling price of the energy produced over the machine’s life-cycle. This same draft tube design is also a key factor in determining the overall civil costs of the powerhouse, primarily in excavation and concreting, which can amount to similar orders of magnitude as the price of the energy produced. Therefore, there is a need to find the optimum compromise between these two conflicting requirements. In this paper, an elaborate approach is described for dealing with this optimization problem. First, the draft tube’s detailed geometry is defined as a function of a comprehensive set of design parameters (about 20 of which a subset is allowed to vary during the optimization process) and are then used in a non-uniform rational B-spline based geometric modeller to fully define the wetted surfaces geometry. Since the performance of the draft tube is largely governed by 3D viscous effects, such as boundary layer separation from the walls and swirling flow characteristics, which in turn governs the portion of the available kinetic energy which will be converted into pressure, a full 3D meshing and Navier-Stokes analysis is performed for each design. What makes this even more challenging is the fact that the inlet velocity distribution to the draft tube is governed by the runner at each of the various operating conditions that are of interest for the exploitation of the powerhouse. In order to determine these inlet conditions, a combined steady-state runner and an initial draft tube analysis, using a stage interface between them, must first be performed for each operating condition. Due to the computationally intensive nature of the evaluation process, the efficiency of the optimization algorithm becomes important. Therefore, a state-of-the-art hierarchical-metamodel-assisted evolutionary algorithm is used.

1. Introduction

Hydraulic turbine draft tubes are an important part of the overall machine, as their main function is to decelerate, in an orderly fashion, the flow leaving the runner in such a way as to convert most of the remaining kinetic energy back into static pressure, thus increasing the effective head that the runner is subjected to. This effect is most prevalent in low to medium head applications, where a greater proportion of the total energy is in kinetic form. Several factors make the design of the draft tube a daunting task. The flow itself, being largely decelerating, is subject to viscous turbulent effects such as wall separations which reduces its effectiveness. To make matters worse, the geometry is often complicated by the inclusion of an approximately 90° bend to improve powerhouse compactness and to minimize civil costs. Furthermore, the exit
cross section is often rectangular, while the entrance cross section is circular to couple with the runner. The geometry of the draft tube design needs to be thought out very carefully to achieve the best possible compromise between hydraulic efficiency and construction costs. This leads to a large number of design parameters. A similar approach [1] was used in the optimization of a 45° inclined Agnew turbine using 2 design parameters (i.e. the cone angle and the height above the tailwater level). In the present case, many more design parameters are needed to represent a more realistic design. This paper describes a way to optimize realistic draft tube hydraulic designs by minimizing the combination of the present monetary value of the recovered energy using viscous turbulent CFD throughout the operating range over the expected life of the machine and the civil cost of building that machine.

The use of stochastic optimization methods in hydraulic design/optimization problems has found widespread use during the last decade [2, 3, 4, 5]. All stochastic optimization methods, being gradient-free methods, are able to incorporate existing analysis software (such as CFD code) as the evaluation tool by simply making it I/O compatible. Evolutionary Algorithms (EAs) are a leading member among the various stochastic optimization methods. This paper relies on a generalized \((\mu; \lambda)\) EA, handling populations of \(\lambda\) offspring and \(\mu\) parents which, depending on the configuration of the evolution parameters, can reproduce a number of well-known EA variants such as genetic algorithms and evolution strategies. Furthermore, the aforementioned \((\mu; \lambda)\) EA is enhanced with a number of speed-up techniques (required to obtain optimum solutions in acceptable wall times) such as the use of metamodels for inexact pre-evaluation (IPE) of the candidate solutions and/or hierarchical optimization schemes.

2. Parametric Design

Elbow draft tubes are usually designed using 20 or more planar sections positioned along a spine curve and oriented with respect to that curve. Various section types, ranging from circles, through ovals to radiused rectangles, may be described using from 1 to 7 parameters depending on the shape. Thus a typical design model requires well over 100 parameters. A model capable of approximating a range of existing elbow draft tubes using as few parameters as possible is required. A “Swedish” parameterization, seen in figure 1, requires far fewer parameters but cannot be converted to simple planar sections along a spine curve. This is important for two reasons: It allows us to refine a parametric design (using our in-house design tool) and it allows us to mesh, perform CFD analysis and post-processing in a completely automated manner (with our standard tool chain). The latter is absolutely essential for the automated optimization procedure to be feasible.

The resulting parameterization is illustrated in figure 2. The elevation view is first built from the geometric parameters by dividing the draft tube into the conventional cone, elbow and extension sections.

Geometric parameters could have been used to define the width profile and hence the final shape, but then the resulting diffusion curve would be arbitrarily determined. It was felt that a more meaningful way of controlling the width would be to use a parametrically derived diffusion curve. Draft tube diffusion area curves typically exhibit an initial diffusing region extending up to a peak area point. This is followed by a short region having constant or decreasing area up to a trough point where the second diffusing region starts. A typical area diffusion curve is shown in figure 3. Our model uses two parameters to define the position and area at the peak point and, optionally, the trough point. The area and slope of the curve at the inlet is known from the cone parameters. If the extension width is assumed constant the area and slope at the exit may be derived from the elevation profile.
With these four points and their associated derivatives, a Hermitian interpolation will produce a smooth diffusion curve.

Knowing the section heights from the elevation profile and the required area from the diffusion curve, only the type of section is needed to completely determine the shape. To complete the geometry, oval sections near the inlet, radiused rectangular sections for the transition and a pure rectangular section for the exit are used. Two additional parameters are needed to specify where the transition occurs between the oval inlet sections and the radiused rectangular sections and where the radiused rectangular sections become fully rectangular.

For designs having piers, four additional parameters are required for each pier: the thickness, the edge radius, the edge lateral offset and the pier start position.

In order to keep the desired diffusion curve, the width calculation is adjusted to take account of the pier width at each derived section.

During the course of an optimization various combinations of parameters are generated. Not all of those combinations will produce an “acceptable” draft tube shape. In addition to the obvious range constraints on each parameter, the integrity of the resulting shape must be checked. Does a pier profile intersect a side wall, for example? Even when the shape appears reasonable, there is no sense in analyzing designs that are probably unstable. Empirically determined in-house limits on the maximum local diffusing rate are used to eliminate designs that are probably unstable.
3. Computational Fluid Dynamics

An automatic tool for draft-tube analysis is presented in this section. An in-house mesh generator and a Reynolds Averaged Navier-Stokes equations solver using the standard $k-\varepsilon$ turbulence model with wall-function are used to perform all computations.

3.1. Mesh Grids

The geometry of the draft tube is exported from the in-house draft tube geometry design tool described above. A single multi-bloc structured mesh combining an O-type block structure near all solid walls and a H-type for the inside flow domain is generated using the in-house automatic draft tube mesh generator [6]. The generated mesh, based on the draft tube entrance $D_{top} = 1$ m, with 240K nodes and 230K elements as shown in figure 4 is exported in the CGNS format [7]. Because a large number of calculations are necessary for optimization, a coarse grid is used without losing too much precision*. The first-node wall distance $Y+$ is around 70. The minimum, average and maximum values for each operating point are given in table 1. Once the mesh has been generated, an automatic script creates all the necessary files for the OpenFOAM calculations.

|     | OP1   | OP2   | OP3   | OP4   | OP5   | OP6   | OP7   | OP8   | OP9   |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Min | 5.77  | 3.14  | 5.02  | 6.62  | 4.98  | 9.13  | 11.00 | 9.95  | 9.54  |
| Ave | 81.02 | 73.97 | 62.5  | 59.14 | 62.73 | 68.18 | 72.57 | 80.83 | 88.21 |
| Max | 156.8 | 107.9 | 97.7  | 97.60 | 113.2 | 128.4 | 139.3 | 168.4 | 188.2 |
| Min | 5.99  | 3.55  | 6.6   | 7.89  | 6.9   | 12.69 | 15.18 | 13.84 | 13.11 |
| Ave | 59.66 | 70.95 | 58.59 | 60.02 | 65.9  | 73.26 | 78.91 | 98.04 | 112.4 |
| Max | 159.5 | 195.2 | 127.7 | 126.7 | 153.6 | 168.6 | 181.4 | 220.6 | 276.7 |

*For example, when comparing a 474K node mesh to a 240K one, the difference is between 0.1% to 2.1% of the calculated losses for the different operating conditions. Moreover, the CPU time for the finer mesh is around 1000 s per operating condition instead of 440 s for the coarser mesh.
3.2. Solver
The standard segregated simpleFoam solver based on the C++ OpenFOAM library is used for the steady state flow computations. OpenFOAM (Field Operation and Manipulation) is an open source package under the GNU General Public Licence. The simpleFoam solver is a steady-state solver for incompressible, turbulent flow of Newtonian and non-Newtonian fluids. It has been validated for the FLINDT project [8] and for the Turbine-99 project [9, 10].

3.2.1. Numerical scheme
An upwind convection scheme was used for all terms except $div(\phi, U)$, for which an improved bounded normalized variable diagram (NVD) scheme GammaV was specified with a coefficient $\psi = 1.0$ after 100 iterations. In the first one-hundred iterations, the upwind convection scheme is used to help convergence. To solve the flow equations, the pre-conditioned bi-conjugate gradient (PBiCG) linear solver is used for all variables except pressure, for which the pre-conditioned conjugate gradient (PCG) is specified. Absolute and relative solution tolerances ($tolerance$ and $relTol$) are specified for each solution variable to control the equation solvers convergence. The $tolerance$ is applied to the residual, which is evaluated by substituting the current solution into the equation and taking the magnitude of the difference between the left and right hand sides [11]. $relTol$ is the ratio of the final over initial residual. In this study, a $tolerance$ value of $10^{-6}$ and a $relTol$ value of 0 are used for all variables except pressure, for which a $tolerance$ value of $10^{-5}$ and a $relTol$ value of 0.0001 are applied. A relaxation factor of 0.3 for pressure and 0.7 for other variables are also applied to improve stability. The finite volume method is applied and the coupling of the velocity and pressure equations is performed using the SIMPLE algorithm.

3.2.2. Boundary conditions
Nine different velocity profiles (one for each operating point) along a radial line at the draft tube inlet, obtained from previous runner and draft-tube calculations (not described here) are used as inlet boundary conditions. The inlet boundary conditions are steady-state and axisymmetric. The same velocity profiles are used for all designs. This simplification is valid because the draft-tube cone angle and the draft-tube inlet area do not change from one design to another. Figure 5 shows an example of a normalized velocity profile for a part load condition. At the draft-tube outlet, an average constant pressure of 0 Pa is used.

3.2.3. Additional information
A maximum of 800 iterations is performed. The computations run in parallel on 6 CPUs on a Linux CentOS 5.5 cluster, using the METIS decomposition method. Typical CPU time is less than 400 s for one operating point.

3.3. Results
Figure 6a shows the evolution of the residual for a part load condition and figure 6b shows the evolution of the static pressure scaled by density between the inlet and the outlet. The small burst in figure 6a at iteration 100 represents the scheme change from upwind to GammaV as already stated. Figure 7 shows the normalized losses in the draft-tube, which are typical results for the optimization.

Figure 5. Normalized velocity profile for part load.
4. Evolutionary Algorithm-Based Optimization Method

Evolutionary Algorithms (EAs) are stochastic optimization methods able to solve both single and multi-objective [12] optimization (SOO or MOO) problems. This is achieved through the evolution of populations of candidate solutions (candidate designs in hydraulic optimization). During the aforementioned evolution, three populations, the offspring $S^\lambda$, the parent $S^\mu$ and the elite $S^a$ populations are handled by the EA. These populations interact with each other during the application of the so-called evolution operators (parent selection, crossover, mutation) and elitism which are inspired by natural/Darwinian evolution. The elite set is populated by the “best” individuals found thus far. In SOO, $S^a$ degenerates to a single individual, the best individual found thus far with respect to the single objective. On the other hand, in MOO, $S^a$ includes all or some of the Pareto non-dominated solutions; in Pareto dominance, by definition, solution $A$ dominates $B$ if $A$ is no worse than $B$ for all objectives and better for at least one objective. Once the non-dominated solutions are available, an experienced decision maker is needed to select the final solution. This final $S^a$ is also referred to as the Pareto front of
optimal solutions. EAs are able to solve MOO problems thanks to an additional process that transforms the objective function vectors into scalar fitness values, thus allowing the evolution operators to proceed as if a SOO problem was handled. This procedure is known as fitness assignment and incorporates dominance and proximity based criteria. As far as constrained optimization problems are concerned, candidate solutions are penalized if they violate a said constraint accordingly. Herein, all penalties are computed using exponential functions that multiply the elements of the cost function vector.

4.1. Metamodel-Assisted Evolutionary Algorithms

When expensive evaluation models such as CFD codes are used, EAs can benefit from the use of metamodels to reduce the computational burden. These are known as metamodel-assisted EAs (MAEAs). In this paper, radial basis function networks (RBFNs) are used as metamodels [13]. Metamodels act as low cost approximation tools within the so-called Inexact Pre-Evaluation (IPE) scheme [14] and require the existence of a database (DB) of exactly evaluated individuals to be trained upon. Therefore, MAEAs start as conventional EAs (without using metamodels) and, once enough exactly evaluated individuals are added into the DB, the offspring pre-evaluation via metamodels starts. Herein, locally trained metamodels are used; i.e. for each offspring, its closest DB entries are identified and an RBFN is trained upon them. After all \( \lambda \) offspring are pre-evaluated, the \( \lambda \) most promising among them, according to the metamodel-based fitness, are selected to undergo exact (re-)evaluation using the costly (CFD based) evaluation tool. All exactly evaluated individuals are archived in the DB. This algorithm has been proposed in [14, 15, 16] and the relevant flowchart is shown in figure 8.

4.2. Hierarchical Metamodel-Assisted Evolutionary Algorithms

Furthermore, hierarchical MAEAs (HMAEAs) [17] are multilevel MAEA variants that outperform conventional EAs and/or single-level MAEAs in many engineering applications, including CFD cases. A HMAEA establishes a multilevel search mechanism in order to split the computational burden among its levels (figure 9). On the lower level, a low cost exploration of the search space is carried out through global search methods, less demanding or less accurate evaluation tools or even by utilizing global artificial neural networks trained on previously evaluated individuals. Higher levels typically utilize more accurate (computationally expensive)
tools coupled with enhanced parameterization schemes and/or different search methods (local optimization). Higher levels mainly serve to a) remove inaccuracies resulting from the lower level evaluation tools and b) refine successful immigrants coming from lower levels. For any HMAEA to be successful two-way interconnection between adjacent levels is required. Note that each level must maintain its own DB due to the fact that different evaluation tools are in use. In addition, all migrated solutions must be re-evaluated with the destination level evaluation tool.

5. Draft Tube Design Optimization

A single objective optimization (SOO) run, using a MAEA configuration (using EASY software [18]), was performed on a single pier draft tube test case where, out of a total of 26 geometric variables, only 12 were actually allowed to vary. These included such basic parameters as overall depth, width, inlet cone height and pier nose position, but also some less obvious ones like some diffusion curve parameters such as peak and trough points which influence the shape of the elbow. The reason for reducing the number of parameters was to reduce the size of the design space thereby reducing the computational effort required to achieve an optimum within a reasonable timeframe.

The population sizes used for the MAEA were $\mu = 5$ parents and $\lambda = 20$ offspring. As soon as 100 exactly evaluated individuals were stored in the database, the IPE phase started in which only the $\lambda e = 6$ out of the $\lambda = 20$ most promising offspring were exactly evaluated, thus reducing the computational cost of each generation by approximately 75%. Each exact evaluation of a candidate solution costs nine equivalent flow solutions since one CFD run per operating point must be performed. During the IPE phase, locally trained Radial Basis Function networks (RBFn) for each candidate solution with 45 training-patterns were used for the metamodels.

The objective function was made up of an estimated cost of the civil works based on the draft tube depth and width in addition to the commercial present value of the energy produced and the available power over the life of the machine using the draft tube loss calculations obtained from the nine CFD runs (one per operating point).

Some constraints were imposed to reject non-viable designs before they were submitted for CFD analysis (figure 10). Some of these constraints included simple stability characteristics like maximum diffusion rate, width at the elbow not exceeding the exit width, constant exit area, etc.

Figure 11 shows the evolution of the computed objective function as it went through about 20 generations of the EA, representing approximately 220 individual designs before converging on an elite solution which could not be improved further. With a maximum of 10 designs which could be analyzed in parallel on the Linux cluster, it took approximately 48 hours to find an optimum design.
As can be seen in figure 12, the curves representing the normalized efficiency losses of the base design and the optimum design as a function of unit flow are nearly superimposed. Figures 13 and 14 show that the optimum design was quite a bit shallower than the initial design because the reduction in civil costs far outweighed the reduction in the value of the power and energy produced based on draft tube efficiency. It can also be observed that the optimized design is rather nice looking and could have been produced by a designer versed in the art, rather than a computer generated design. Obviously, further investigation of this outcome would be needed to validate its viability in terms of efficiency and pressure pulsations before engaging substantial resources such as model testing.

**Figure 11.** Convergence history plot in terms of exact evaluations (9 CFD computations). Exact evaluations are proportional to the computational cost of the optimization.

**Figure 12.** Normalized initial (solid) and optimized (dotted) design efficiency loss as a function of normalized unit flow.

**Figure 13.** Elevation, plan and developed view for initial (black) and optimized (red) designs.

**Figure 14.** Rendering of draft tube designs.
Conclusion
It has been shown that realistic draft tube designs using a fairly large number of design parameters can be optimized for hydraulic loss performance and civil construction costs, using a fairly sophisticated 3D viscous flow CFD approach, and that this can be done within a reasonable time frame on generally available computing resources by utilizing a MAEA optimization scheme. Additional performance characteristics, such as the performance within an actual rotor/stator environment or for pressure pulsations at part load, would be advantageous. To achieve this, more sophisticated, and therefore computationally demanding, analysis tools are required. The use of these tools in a single level EA would lead to unacceptable optimization wall time with today’s computer technology. To tackle this problem the use of a hierarchical EA, which utilizes the availability of a variety of analysis tools with varying degrees of sophistication so as to minimize the overall optimization wall time, will be the next step taken by the authors.

References
[1] Shojaeefard M H, Mirzaei A and Babaei A 2014 Energy Conversion and Management 79 681 – 689 ISSN 0196-8904 URL http://www.sciencedirect.com/science/article/pii/S0196890413008017
[2] Georgopoulou C, Kyriacou S, Giannakoglou K, Grafenberger P and Parkinson E 2008 Constrained multi-objective design optimization of hydraulic components using a hierarchical metamodel-assisted evolutionary algorithm. Part 1: Theory 24th IAHR, Symposium on Hydraulic Machinery and Systems (Foz do Iguassu, Brazil)
[3] Grafenberger P, Parkinson E, Georgopoulou C, Kyriacou S and Giannakoglou K 2008 Constrained multi-objective design optimization of hydraulic components using a hierarchical metamodel-assisted evolutionary algorithm. Part 2: Applications 24th IAHR, Symposium on Hydraulic Machinery and Systems (Foz do Iguassu, Brazil)
[4] Kyriacou S, Weissenberger S, Grafenberger P and Giannakoglou K 2010 Optimization of hydraulic machinery by exploiting previous successful designs 25th IAHR, Symposium on Hydraulic Machinery and Systems (Timisoara, Romania)
[5] Kyriacou S, Weissenberger S and Giannakoglou K 2012 International Journal of Mathematical Modelling and Numerical Optimization (SI:Simulation-Based Optimization Techniques for Computationally Expensive Engineering Design Problems) 3 45–63
[6] Guibault F, Zhang Y, Dompierre J and Vu T 2006 Robust and automatic CAD-based structured mesh generation for hydraulic turbine component optimization 23rd IAHR, Symposium on Hydraulic Machinery and Systems (Yokohama, Japan)
[7] Allmaras S and McCarthy D CGNS CFD standard interface data structures - version 3.2.1 http://www.grc.nasa.gov/WWW/cgns/CGNS_docs_current/sids/
[8] Vu T, Devals C, Zhang Y, Nennemann B and Guibault F 2011 Int. J. Fluid Machinery and Systems 4
[9] Nilsson H 2006 Evaluation of openfoam for CFD of turbulent flow in water turbines 23rd IAHR, Symposium on Hydraulic Machinery and Systems (Yokohama, Japan)
[10] Tanase N O, Bunea F and Ciocan G D 2012 University Politehnica of Bucharest Scientific Bulletin, Series D: Mechanical Engineering 74 83–90
[11] OpenFOAM Openfoam user guide http://www.openfoam.org/docs/user/
[12] Zitzler E, Laumanns M and Thiele L 2011 Eurogen
[13] Haykin S 1999 Neural Networks: A Comprehensive Foundation 2nd ed (New Jersey, U.S.A.: Prentice Hall)
[14] Karakasis M, Giotis A and Giannakoglou K 2003 International Journal for Numerical Methods in Fluids 43 1149–1166
[15] Karakasis M and Giannakoglou K 2006 Engineering Optimization 38 941–957
[16] Giannakoglou K, Giotis A and Karakasis M 2001 Inverse Problems in Engineering 9 389–412
[17] Kampolis I and Giannakoglou K 2008 Computer Methods in Applied Mechanics and Engineering 197 2963–2975
[18] Giannakoglou K 2008 The EASY (Evolutionary Algorithms SYstem) software http://velos0.ltt.mech.ntua.gr/EASY