A New Tool for Topological Optimization of a Rotor for Vertical Axis Wind Turbines

Mateusz JAKUBOWSKI
Poznan University of Technology (Graduate in the field of Mechatronics)
ul. Jana Pawła II 24, 60-965 Poznań, mateusz@konstrubowski.pl

Paweł FRITZKOWSKI
Institute of Applied Mechanics, Poznan University of Technology
ul. Jana Pawła II 24, 60-965 Poznań, pawel.fritzkowski@put.poznan.pl

Abstract

A computer program for topological optimization of a rotor for vertical axis wind turbines of various type is presented. The tool is based mainly on two external modules: the GMSH mesh generator and the OpenFOAM CFD toolbox. Interpolation of rotor blades geometry and computational model of the airflow through a turbine are briefly discussed. Moreover, a simple optimization algorithm is described. Exemplary results for a H-type rotor are presented. Finally, potential directions for the software development are indicated.

Keywords: vertical axis wind turbines; topological optimization; computational fluid dynamics

1. Introduction

Vertical axis wind turbines (VAWT) have great potential in the area of renewable energy generation, although they are relatively rarely used in industry. Modern manufacturing methods make production of complex geometric shapes increasingly cheaper. Thus, topological optimization of rotor blades can provide quite valuable results that are realizable in practice.

Aerodynamics of turbines is complicated and sensitive to slight changes in shape. Therefore, the software for finding the best possible geometry of a rotor is of high importance for design engineers. There are many commercial systems (usually based on the finite element method) that allow one to solve a wide variety of problems in the field of computational fluid dynamics (CFD). However, the general purpose character of such programs makes particular tasks rather burdensome: computational model preparation, geometry parameterization, etc. In this light, developing a specialized optimization tool seems to be an attractive and challenging idea.

From the programming and numerical viewpoint, CFD-related problems are very demanding. A common (but not always occurring) feature of vertical turbines, i.e. a uniform cross-section of a rotor along the axis of rotation (see Fig. 1), simplifies the problem considerably. In any case, it was decided to create the program with a use of commonly available components: free, open source packages which fulfill the crucial and hardest tasks.

a) b) c)

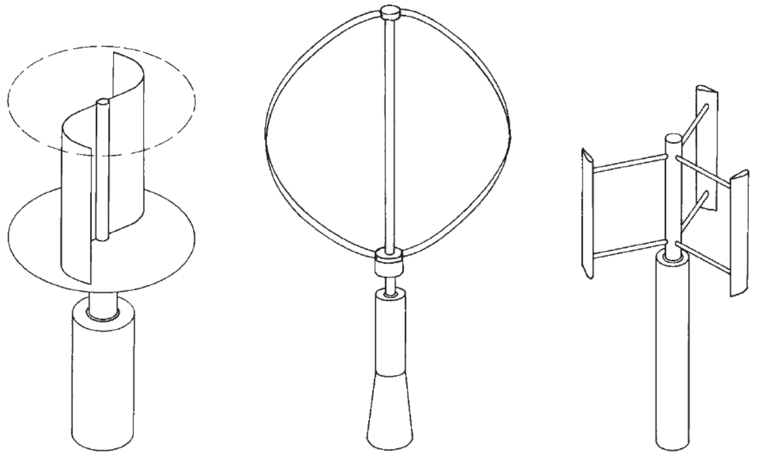


Figure 1. Basic types of rotors for VAWT [1, 4]: a) the Savonius rotor, b) the Darrieus rotor, c) the H-type rotor

2. The OPTIMIZER software

The computer program *Optimizer* developed by the first author has a graphical user interface and was written in the Python programming language. The application accomplishes the following tasks: drawing the initial geometric model of a rotor, generating and previewing a discrete numerical model, changing the simulation and solver settings, results archiving, conducting simulation related to the direct problem (air flow through a wind turbine), optimization of the rotor shape, and results visualization. *Optimizer* employs external modules: the *GMSH* mesh generator and the solvers of the *OpenFOAM* environment. The program window with sample data can be seen in Fig. 2.

In order to reduce the number of parameters describing the rotor geometry, it was decided to use interpolation of curves that pass through some control points specified by user. More precisely, the method known as Piecewise Cubic Hermite Interpolation (PCHI) is used [3]. On each subinterval the given curve is interpolated by a third degree polynomial of Hermite type. To form a smooth contour of a blade, continuity of the first and second derivatives of neighbouring polynomials is ensured at the nodal points. Obviously, this constraint is cancelled in case of a corner vertex (see Fig. 3). All in all, user defines the shape of a single blade and the number of blades.

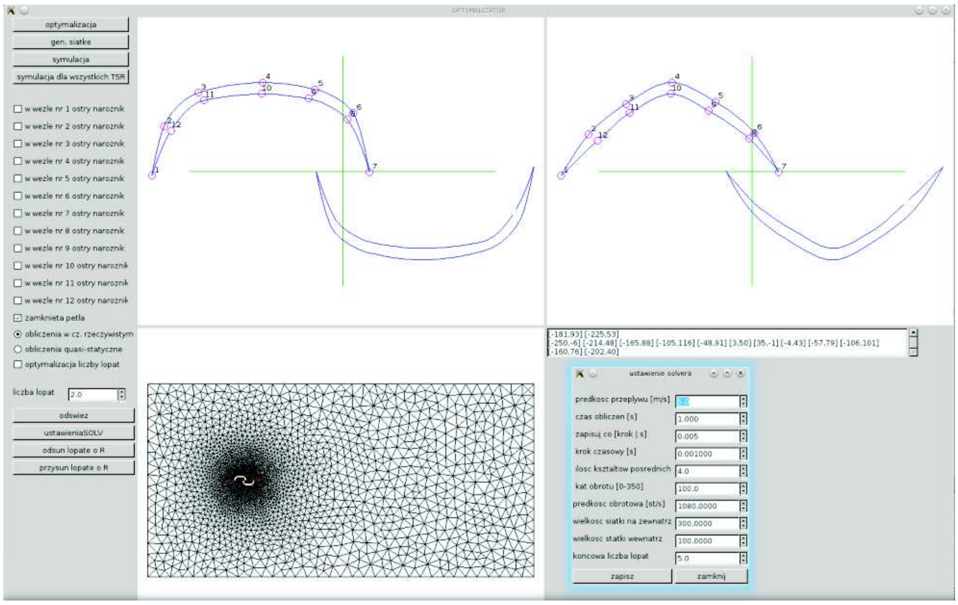


Figure 2. Graphical interface of *Optimizer*

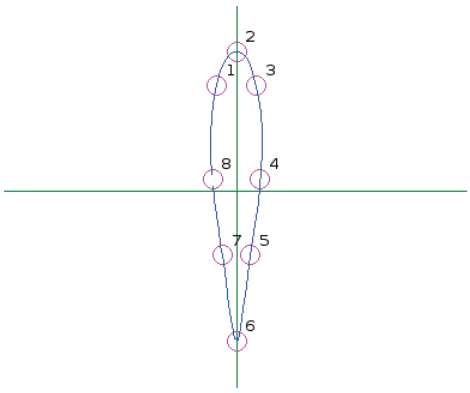


Figure 3. Shape of a blade of the H-type rotor generated via PCHIP

3. Computational model and solver

A schematic view of a virtual wind tunnel is presented in Fig. 4. The problem domain is divided into two parts. The central one includes the rotor and its close neighbourhood, thus, it rotates during simulation. The non-moving subdomain constitutes an outer, dominant part. The interface between the two regions forms a circle centered at the rotor axis.

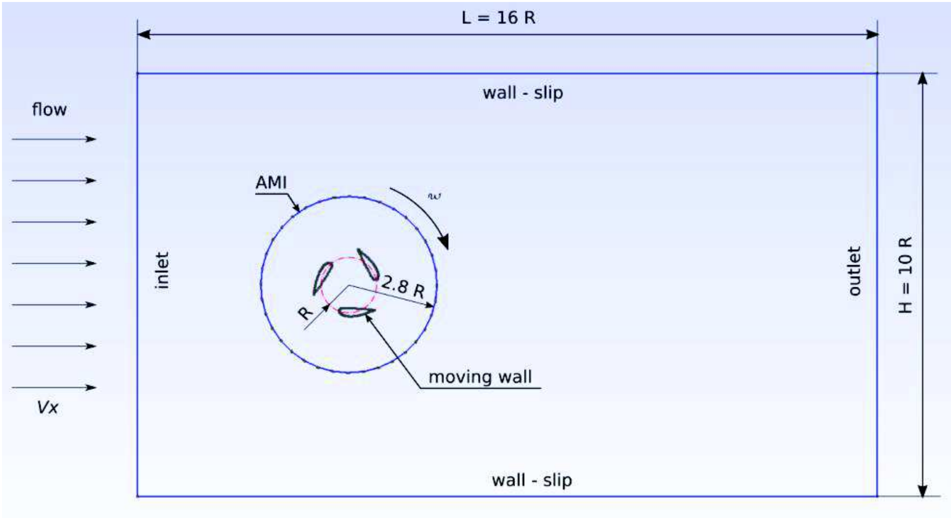


Figure 4. The domain and boundary conditions for the problem

Airflow through the wind turbine is described by the Navier-Stokes equations; the fluid is assumed to be incompressible [4]. On the left boundary, uniform inflow velocity of the air is defined. At the bottom and top walls the slip condition is specified, which prevents the fluid from leaving the domain. The outlet condition corresponds to zero relative pressure. AMI stands for Arbitrary Mesh Interface, and is used to model the mutually sliding subdomains.

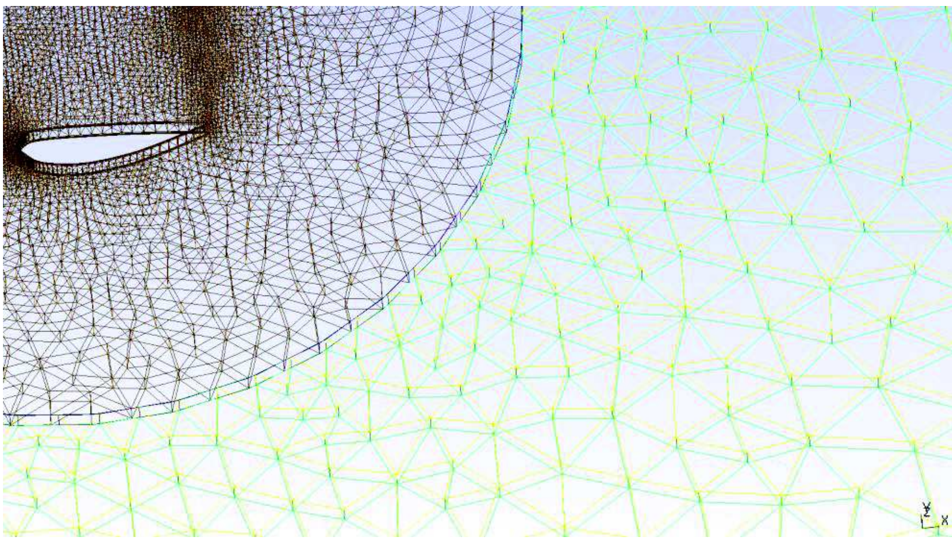


Figure 5. An exemplary finite volume mesh

In general the planar domain is discretized by triangular cells. Additionally, quadrilateral cells are used in the boundary layers on the blades. The solver employed to cope with the initial-boundary value problem is based on the finite volume method (FVM). Since a three-dimensional computational domain is required, the planar one is extruded by a unit distance. Therefore, the final discrete model consists of prismatic finite volumes (see Fig. 5).

Within the *OpenFOAM* environment, the solver *pimpleDyMFoam* allows for dynamic meshes. It is an implementation of the so called PIMPLE algorithm: a combination of the standard PISO (Pressure Implicit Split Operator) and SIMPLE (Semi Implicit Method for Pressure Linked Equation) algorithms. To guarantee continuity of physical quantities on the interface between moving and stationary cells, an additional interpolation is used.

4. Optimization algorithm

The algorithm for topological optimization of rotor blades is made of two modules: a generator of new rotor geometries, and an analyzer and selector of the best solution. The former one requires the following user-defined input data: the start and end blade shapes as well as the number of intermediate profiles (resolution). The algorithm analyzes the given geometries and prepares a set of new shapes according to the simple principle illustrated in Fig. 6. The left and right triangles represent the start and end profiles, respectively. The middle triangle, in turn, illustrates the only intermediate shape (a special case is shown). Starting from the initial shape, translation vectors for all control nodes are determined, which leads to a new interpolated geometry. The number of translations is equal to the number of intermediate profiles.

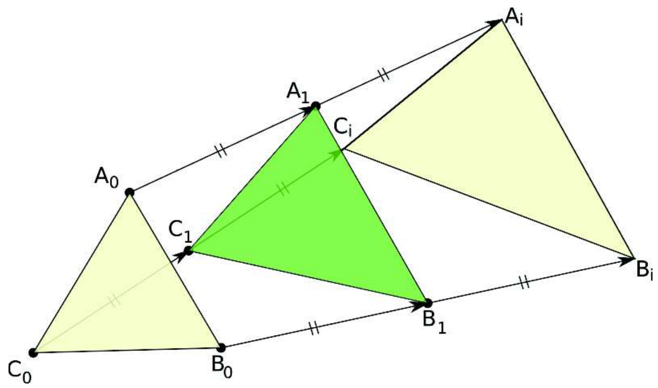


Figure 6. Illustration of the start, intermediate and end shapes

As the comparison criterion (an objective function), the power coefficient is used:

$$C_p = \frac{P_r}{P_w}. \tag{1}$$

The power of the rotor and the wind flowing past the rotor are given by

$$P_r = \omega T\ , \quad P_w = \frac{1}{2}\rho A V_x^3\ , \tag{2}$$

where: ω – angular velocity of the turbine, T – torque generated by the rotor, ρ – air density, A – rotor area in the cross-section normal to the airflow direction, V_x – airflow velocity. If the defined start and end shapes ensure a constant rotor diameter during optimization, this criterion can be simplified and replaced with the torque at the rotor shaft. The torque value is specified on the basis of pressure field at the blades, and is saved to file in real time. As simulations related to all the prepared profiles are completed, the program analyzes the results and presents the best solution.

5. Results

The *Optimizer* software was tested by solving several direct and optimization problems related to the Savonius and H-type rotors. Results of these studies are thoroughly discussed in Ref. [2]. Here, only one example is presented.

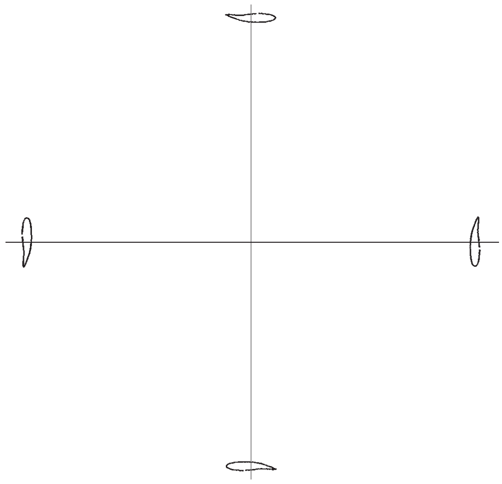


Figure 7. Initial geometry of the NS2L4 rotor

Consider blade shape optimization for a H-type rotor denoted by the code NS2L4 (see Fig. 7). Geometry of the start, end and intermediate profiles is illustrated in Fig. 8. The corresponding values of the rotor torque and power are shown in Fig. 9. As can be seen, the best solution (in terms of the torque criterion) is denoted by SERIES-4. A detailed analysis of the results has indicated that this blade variant generates the weakest vortices. Time history of the torque T for the initial and optimized shapes is presented in Fig. 10. The simulations were performed for R = 700 mm, V_x = 21.6 km/h. Fluid-structure interaction was analyzed for time $0 \le t \le 2$ s. The optimization lasted 17 690 s.

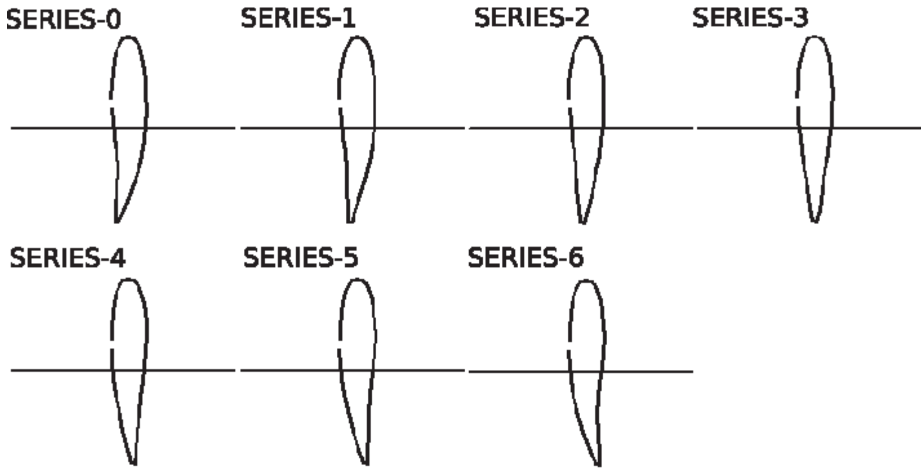


Figure 8. Blade profiles in consecutive iterations

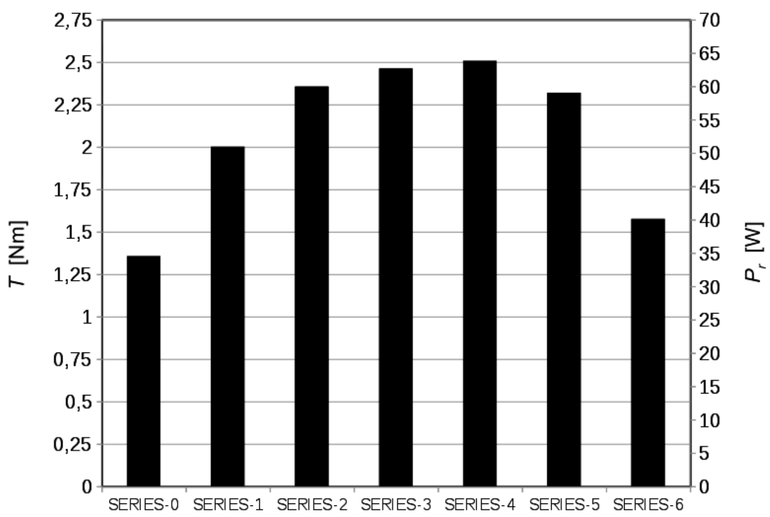
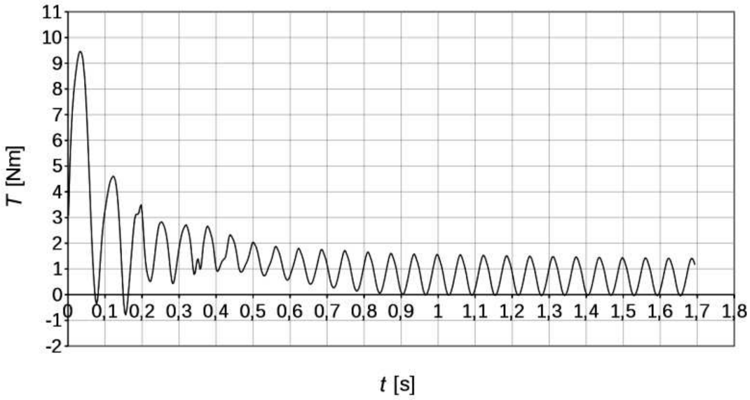


Figure 9. Maximal torque and power (steady-state) of the rotor in consecutive iterations

6. Conclusions

The presented software is a specialized tool that can be applied for topological optimization of various types of VAWTs. To create a true alternative to commercial systems, the program should be improved by increasing functionality of the preprocessor and postprocessor. Nevertheless, *Optimizer* is a solid basis for implementation of advanced optimization approaches, e.g. artificial neural networks (ANN) or genetic algorithms (GA).

a)



b)

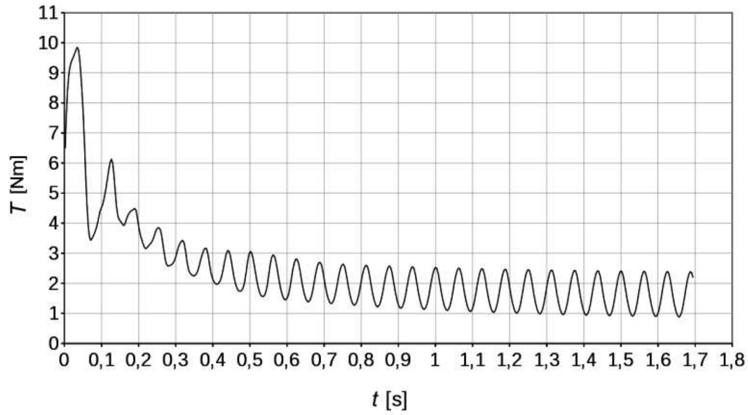


Figure 10. Time history of the rotor torque for SERIES-0 and SERIES-4

References

1. E. Hau, *Wind Turbines – Fundamentals, Technologies, Application, Economics,* Springer, Berlin 2014.
2. M. Jakubowski, *Software for topological optimization of a rotor for a vertical axis wind turbine*, Master thesis, Poznan University of Technology, Poznan 2015.
3. N. Madden, *Lecture 10: Piecewise Hermite Interpolation,* National University of Ireland, Galway 2010.
4. A. P. Schaffarczyk, *Introduction to Wind Turbine Aerodynamics*, Springer, Berlin 2014.