uc3m | Universidad Carlos III de Madrid

Bachelor's Degree in Aerospace Engineering
2016/2017

Bachelor Thesis

Development of a computational tool for turbomachinery Blade Generator

Ionuț Alexandru, Coșuleanu

Tutor
Antonio Antorranz Perales
Escuela Politénica Superior (Leganés)

To my family, friends and J.P.M.

Contents

List of Figures

1 Abstract

2 Nomenclature

2.1 Symbols

2.2 Abbreviations

3 Introduction

Along this chapter, a brief introduction of turbomachinery is presented (such as definition, the physics behind and other related concepts). Further, related more specifically with this project, the motivation and objectives of this work are specified, together with an explanation of this project's organization.

Generally, a turbomachine refers to those devices in which there is a flowing fluid and its energy is transferred to a rotating blade row. This changes the stating enthalpy of the fluid by extracting work from the device or from the fluid. These devices are also referred as rotordynamic devices because the dynamic action of the fluid, which is guided through an annular duct, is generated by the rotor. Therefore, according to this definition, two main types of devices can be encountered:

- Those which *absorb* power to increase the fluid pressure - this is the case of fans, compressors and pumps
- and those which *produce* power by decreasing the fluid pressure - gas turbines, for example.

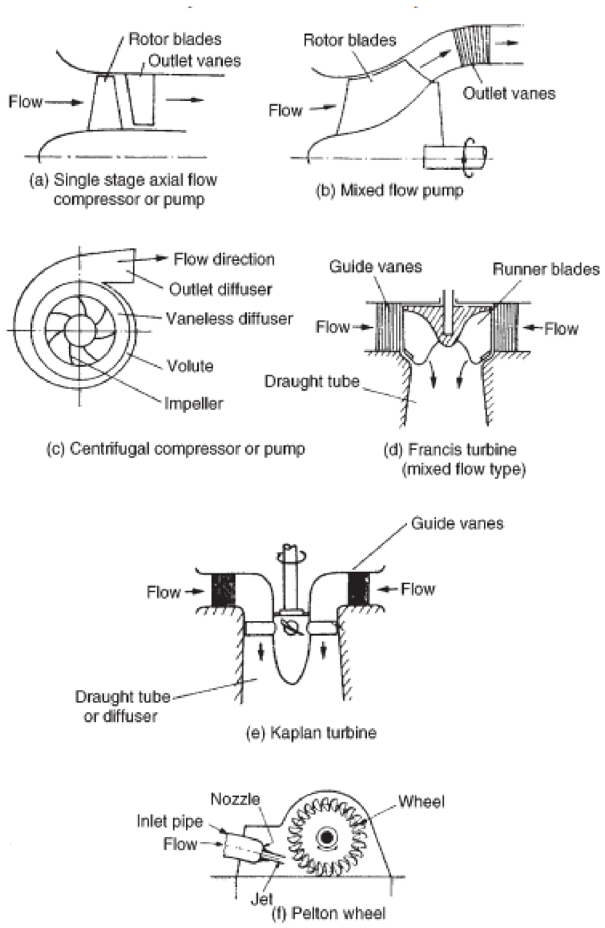

Figure 1: Examples of types of turbomachines: a) and e) are axial devices; c) radial; and b) and d) are mixed flow turbomachines. f) is an impulse turbine

Apart from that, there is another categorization of turbomachines depending on the flow direction; thus, when the flow is parallel to the axis of rotation, the device is called *axial flow turbomachine*; the flow can be also perpendicular to the axis of rotation and in this case is referred to us as a *radial flow turbomachine*; and the last type is *mixed flow turbomachine*.

There is a last classification depending on the amount of energy transformed into kinetic energy. That is, if all the hydraulic energy is converted into kinetic energy the machine is called *impulse* and *reaction turbomachine* if only some amount is converted.

All these classification is sketched in Figure 1.

Along this project, it will be analysed the performance of axial turbine and compressors, depending on the the blade geometry, as it is going to be shown in the next sections. This analysis is done under a specific organization for the blades: they are said to form a *cascade* and that means that they are close in terms of proximity. And this implies that the individual behaviour of a blade is affected by the adjacent blades.

Improve this and insert a figure!

3.1 Motivation

As said before, one of the variable of the performance of a turbomachine is the geometry of the blade within the cascade. So, having proper tools to determine and analyse easily the performance of the blade is of importance for many industries such as aerospace. To be kept in mind that the main source of power of an aircraft is a gas turbine.

It is of importance then that students and researchers use these kind of tools (and in this case a blade geometry designer for turbine and compressors) to perform their analysis. Therefore, the motivations behind this project are:

1. The opportunity of acquiring deeper turbomachine geometry knowledges

2. Acquiring computational skills related with Matlab, parametrized curves theory and interface programming

Mejorar!!

3.2 Objectives

vbcvb

3.3 Organization of the document's sections

jkljk

4 State of the art

5 Methodology

5.1 Basic concepts and principles

5.2 Introduction to Parametric graphs

5.2.1 Bézier Curves

In computer graphics, it is convenient (because of its easiness) to use the so-called "parametric curves". Therefore, a curve is represented like a n grade polynomial that depends on a parameter t:

$$\vec{c}(t) = \vec{a_0} + \vec{a_1}t + \cdots + \vec{a_n}t^n \tag{1}$$

where the coefficients $\vec{a}_i$ are points of the plane and t is a parameters that goes from 0 to 1: from the starting point of the curve ($\vec{a}_0$) and the final point ($\vec{a}_0$).

This way of representing a curve has one important advantage that is its easiness (as said before) but its main drawback is the fact that coefficients behave in a complex way when, for example, one wants to rotate the curve or translate it. Therefore, it is convenient to use another type of representation instead of polynomial curves (bur following the same *philosophy*). And one of this type of computational representation of curves are *Bézier curves*.

Bézier curves are very used in computer graphics because they are smooth. In few words, they are constructed by a set of control points, and these points can be displaced in order to modify the plot. As previously mentioned, parametric curves can be transformed (namely, they can be rotated and/or rotated) and in the case of Bézier curves, one has just only to transform the control points properly in order to have the line transformed.

The degree of a Bézier curve is defined by the set of control points. If $\vec{c}(t)$ is only defined by $\vec{a}_0$ and $\vec{a}_1$, the curve is linear. If it is defined by three points, then it is quadratic. And the degree can be increased up to n. The computational complexity increases with n. In the next figure (Figure 2) , one can see these two types of curves in a 2D plane:

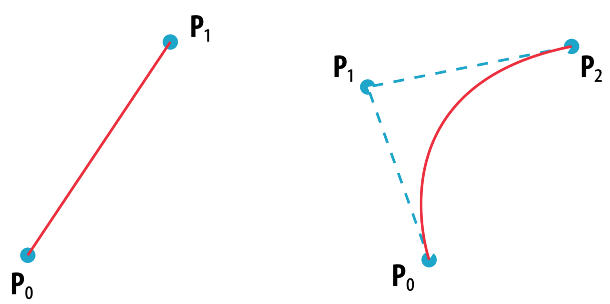

Figure 2: Two examples of Bézier curves. Left side - linear cuve; Right side - quadratic curve

For a linear curve, the line is expressed as a straight line:

$$\vec{c}(0) = P_0 + t(P_1 - P_0) = (1-t)P_0 + tP_1 \tag{2}$$

For higher orders of n, the process is more complex. Let us develop the Bézier curve for a quadratic curve. For this kind of lines, implicitly, the control point P_1 depends on the tangent line of the finishing points.

Making use of Equation 1 and defining $\vec{c}(t)$ as a $(x_B(t), y_B(t))$,

$$x_B(t) = a_{0x} + a_{1x}t + a_{2x}t^2$$

$$y_B(t) = a_{0y} + a_{1y}t + a_{2y}t^2$$

or in a matrix form:

$$\vec{c}(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{Bmatrix} t^2 & t & 1 \end{Bmatrix} \begin{bmatrix} a_{2x} & a_{2y} \\ a_{1x} & a_{1y} \\ a_{0x} & a_{0y} \end{bmatrix} = TC_B = TM_BG_B \tag{3}$$

From Equation 3, T, G_B and M_B are defined as:

- $T = \begin{Bmatrix} t^2 & t & 1 \end{Bmatrix}$
- $G = \begin{bmatrix} P_{2x} & P_{2y} \\ P_{1x} & P_{1y} \\ P_{0x} & P_{0y} \end{bmatrix}$
- and matrix M_B is the unknown of the equation and it relates matrices G_B (which depends on the control points) and C_B (which is the coefficients matrix)

In order to determine M_B, let us state the next conditions:

1. $\vec{c}(0) = P_0 = \begin{Bmatrix} 0 & 0 & 1 \end{Bmatrix} M_BG_B$
2. $\vec{c}(1) = P_2 \begin{Bmatrix} 1 & 1 & 1 \end{Bmatrix} M_BG_B$
3. $R_1 = \frac{P_2 - P_1}{t_2 - t_1} = \frac{P_1 - P_0}{1 - 1/2} = 2(P_2 - P_1) = \vec{c}\ '(1) = T'(1)M_BG_B = \begin{Bmatrix} 2 & 1 & 0 \end{Bmatrix} M_BG_B$

R_1 is defined as the slope (or tangent) between points P_2 and P_1, and also note that at P_2, $t = 1$ and at P_1, $t = 1/2$. In a matrix form, these conditions are rewritten as follow:

$$G_H = \begin{Bmatrix} P_0 \\ P_2 \\ R_1 \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 2 & -2 & 0 \end{bmatrix} G_B = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 2 & 1 & 0 \end{bmatrix} M_BG_B \tag{4}$$

From this last equation, M_B can be calculated and it is equal to:

$$M_B = \begin{bmatrix} 1 & -2 & 1 \\ 0 & 2 & -2 \\ 0 & 0 & 1 \end{bmatrix} \tag{5}$$

and since $\vec{c} = TM_BG_B$,

$$\vec{c} = (t-1)^2P_0 + 2t(t-1)P_1 + t^2P_2 \tag{6}$$

A similar procedure can be followed for a cubic Bézier curve. And, for this case, it is define as:

$$\vec{c}(t) = (1-t)^3P_0 + 3(1-t)^2tP_1 + 3(1-t)t^2 + t^3P_3 \tag{7}$$

and it can be observed it has the shape of *Bernstein basis polynomial*, therefore, the explicit equation of order n is:

$$\vec{c}(t) = \sum_{i=0}^{n} \binom{n}{i} (1-t)^{n-1}t^iP_i \tag{8}$$

From the practical point of view, only quadratic form of Bézier parametric graphs is the one that it is used for this work.

5.3 Blade profile geometry

There are two ways to design a blade for a compressor and a turbine. These approaches are called:

- *Direct (or analysis) method*: it is basically a geometrical technique. The cascades generated are further analysed by experimental test or theoretical approaches in order to find performance desired.
- *Inverse (or synthesis) method*: this is an advanced method and it is the opposite than the direct approach; in this case, the user specifies the velocity and pressure distribution along the surface of the blade. The method that generate the blade profile from these distribution is called *Prescribed Velocity Distribution* or *PVD* analysis.

It may be thought that inverse method is the one that gives the best approach but, from a practical point of view, direct method is widely used by engineers. And this method is used for this project. In the next sections, it is explained how these profiles are obtained.

5.3.1 Direc Method

As explained previously, with this method, the engineer elaborate a cascade family and then it is analysed experimentally or by theoretical means (e.g. MISES software). There are distinguished two ways: one given the thickness family; and another, that was named as *Higher parameters design*, the user has fully control about the design. Both algorithms are similar in the way that the software ask the user about certain input and, making use of geometrical techniques, the software gives as output the blade profile.

Related Cascade families

The procedure follow, as shown in the next flow chart (Figure 3), is:

1. Build the Camber Line
   (a) Ask the user for the Camber Line inputs: metal and stagger angles
   (b) Build the Camber Line with Bézier Theory
2. Add perpendicularly (to the Camber Line) the thickness family

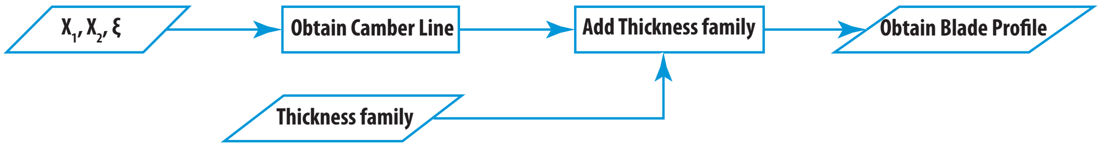

Figure 3: Direct method flow chart for a prescribed thickness family

The Camber Line is obtain using three control points:

Higher parameters design

5.4 MISES

6 Results

6.1 Design

6.1.1 Create a Turbine Blade profile

6.1.2 Create a Compressor Blade profile

6.1.3 Higher parameters design

Turbine

Compressor

6.1.4 Other features of the software

6.2 Analysed designs

6.2.1 Blade Profiles

6.2.2 Analysis of MISES results

7 Possible Improvements

8 Conclusion

9 Bibliography