Advances in Engineering Software 67 (2014) 164–172

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

An evolutionary geometric primitive for automatic design synthesis of functional shapes: The case of airfoils

L. Di Angelo, P. Di Stefano *

Department of Industrial and Information Engineering and Economics, University of L'Aquila, Italy

ARTICLE INFO

Article history:
Received 7 June 2013
Received in revised form 18 September 2013
Accepted 6 October 2013
Available online 26 October 2013

Keywords:
Geometric primitive
Automatic design synthesis
Artificial neural networks
Airfoils design
Self-adaptive geometric primitive
Functional shape synthesis

ABSTRACT

A novel self-adaptive geometric primitive for functional geometric shape synthesis is presented. This novel geometric primitive, for CAD use, is specifically designed to reproduce geometric shapes with functional requirements, such as the aerodynamic and hydrodynamic ones, once the functional parameters are furnished. It produces a typical CAD representation of a functional profile: a set of Bézier curves. The proposed primitive follows a generate-and-test approach and takes advantage of the use of a properly designed artificial neural network (*BNN*). It combines the properties of a geometric primitive and the capability to manage the engineering knowledge in a specific field of application. The proposed evolutionary primitive is applied to a real engineering application: the automatic synthesis of airfoils. Some examples are simulated in order to test the effectiveness of the proposed method. The results obtained by an original prototypal software are presented and critically discussed.

1. Introduction

The synthesis of new functional shapes is the scope of many design activities. Typical domains are all those where functional constraints qualify the geometric shape, such as the aerodynamics and the hydrodynamics fields. The design task is to obtain a geometric shape that satisfies some predefined functional requirements.

Generally, for this kind of engineering problem, the geometric shape, which satisfies functional requirements, might allow arbitrary alternatives; in other words, there does not exist a unique association between functional requirements and geometric shape. Furthermore, the typical geometric primitives, used to design the geometric shape, require input geometric coefficients (the control points of the shape), which, in many practical cases, are not directly associable with the characteristic functional parameters; such is the case of airfoil design. For these reasons, the design synthesis of a functional geometric shape is a complex task for the designer to accomplish; he can easily define a geometric shape that satisfies geometric requirements, but he cannot design as easily a geometric shape that satisfies some specific functional requirements (for example, aerodynamic requirements).

With a view to proposing a solution to this design problem, a new geometric primitive is suggested in this paper. This geometric primitive is a computational artefact, which autonomously performs the synthesis of a functional geometric shape from the input of its functional parameters. For this purpose the new type of geometric primitive needs to incorporate the knowledge concerning the specific design problem, so that a solution between the many alternatives can be synthesized. The geometric shape that can be generated must be rigorously compatible with the available knowledge, which is integrated into the geometric primitive. Generally, for this kind of engineering problem there is not a known and well-defined mathematical map between functional requirements and geometric data that represent any geometric shape, not even when the required knowledge concerns a specific design domain. The geometric primitive here proposed is therefore conceived as an evolutionary artefact, which can not only use the old knowledge but also produce and reuse the new knowledge which concerns the specific design problem and which has been obtained while generating new geometric shapes.

The proposed geometric primitive benefits from the evolutionary strategies so that, for each generation item, its capability to synthesize a functional geometric shape is improved and may offer a wider set of engineering solutions. The generative capability comes from prior similar designs and is improved thanks to the knowledge acquired during designing.

The automatic synthesis of a geometric shape cannot be performed without the capability of using the previous knowledge concerning the design problem. There is an initial knowledge that represents the imprinting of the automatic process. The starting knowledge must be based on some well-defined patches of valid

---

* Corresponding author. Address: via G. Gronchi 18, 67100 L'Aquila, Italy. Tel.: +39 0862434314; fax: +39 08624333613.

E-mail address: paolo.distefano@univaq.it (P. Di Stefano).

URL: http://ing.univaq.it/ProDesLab/eng/index.php (P. Di Stefano).

0965-9978/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.advengsoft.2013.10.002

knowledge which covers all the essential aspects of the design problem. The functionality of the proposed primitive evolves during its lifetime, based on the experience acquired, and cannot be known in advance. The design solutions generated in the life cycle and their generation history affect the behaviour of the primitive. In order to reuse the knowledge acquired during the design process, it is essential to define efficient and dynamic ways to store and validate it. During the generative process it is crucial to monitor new solutions which may be generated so as to accept or refuse them. The generation of unexpected results may lead to new design solutions if they can be analysed and understood. That is why an important role is played not only by the monitoring of the generative process, but also by its capability both to evaluate the performance of the generated solution and to recognise new cases. Typically, the automatic design process is monitored by a mathematical model that simulates the physical principle that governs the technical solution. This knowledge is valid if the analysis returns correct inferences about the design solution.

This paper actually analyses the possibility of using a new kind of self-adaptive geometric primitive; this primitive excels the traditional approaches to functional geometry design by performing an automatic synthesis of the geometric shape directly from the functional parameters that characterise it. This primitive is designed to autonomously perform the *generate-and-test* process, which typically characterises the geometric shape synthesis, and to reuse the knowledge acquired in that process.

The proposed method has found its generative capability both in the knowledge that is transferred to the geometric primitive and in that knowledge which the primitive generates while testing the attempted solutions. Thus, the method extends its generative capability by generalising from the available knowledge. The present approach benefits from a dedicated artificial neural network that directly performs the synthesis of the geometric shape in the form of Bézier curves.

2. Related works

The approach typically used in airfoil design solves an optimisation problem, in which a shape needs to be computed in order to fit assigned aerodynamic parameters. The optimisation approach requires the integration of different modules; each of them plays an important role in the airfoil design process:

- geometric shape function: it serves to define the geometric shape of the airfoil;
- computational solver: it performs a functionality of the geometric shape;
- search model for shape optimisation.

2.1. Geometric shape function

In airfoil optimisation the parametric curves typically used are spline [1], B-spline [2–6] and Bézier [7–14]. These curves can directly represent the geometric shape of the airfoil or the *mean camber line* and *airfoil thickness* in the way proposed by Abbott et al. [15]. As pointed out by Song and Keane [16], the first approach has a greater capability than the second one to generate radical new shapes but it is nevertheless less suitable to produce efficient designs because of the larger space in terms of alternative shapes. Marinus [17] states that the B-spline representation delivers better aerodynamics results and it is more efficient than the Bézier formulations in terms of convergence. Quite frequently, as far as the use of the Bézier curves is concerned, the uniform parameterisation is used to approximate the airfoil geometry [14]. Tang and Desideri [18] demonstrate that the best Bézier approximations are obtained by using a self-adaptive parameterisation but the convergence of the method proves to be more difficult to achieve.

PARSEC parameterisation [19] has been used by [20–22] to represent an airfoil shape by using geometric parameters having a physical relevance for the aerodynamic problem. It is based on a polynomial expression for both sides of the airfoil with a reduced number of design parameters (airfoil slope at the trailing edge, coordinates and second derivatives of the ordinate of the point at the maximum thickness, radius of curvature at the leading edge). Oyama et al. [23] show that this type of parameterisation improves the robustness and convergence speed for an aerodynamic optimisation involving genetic algorithms. In order to combine the advantages of both the Bézier and the *PARSEC* parameterisation, some authors [24,25] introduce the Bézier–*PARSEC* parameterisation. By this approach the *mean camber line* and *airfoil thickness* are expressed in the form of Bézier curves by transforming the assigned *PARSEC* parameters.

2.2. Computational solver

This module typically consists of a flow solver that, starting from the airfoil coordinates, Reynolds and Mach numbers, calculates the corresponding aerodynamic coefficients (pressure distribution on the airfoil and lift and drag characteristics). In the airfoil design process the *CFD* solver is very time-consuming though. It is estimated that it consumes 90% of the computational time required [26]. The quality of the practical results largely depends on the validity of the results yielded by the *computational solver* which performs the monitoring of the airfoil shape optimisation. The *computational solver* is a very important component of the optimisation module; a detailed analysis of its related literature, which is both wide and voluminous, is outside the scope of this work. For the implementation of the evolutionary geometric primitive described in this work the XFOIL [27] has been used as the computational solver.

2.3. Search model for shape optimisation

Typically, the most important optimisation algorithms fall within two categories: *gradient-based methods* and *global-based methods*.

The *gradient-based methods* are iterative methods that extensively use the information of the gradient of the function to be minimised during iterations. Conventional *gradient-based methods* are used for single-point airfoil design for which one regime of flow condition is analysed [28,29]. These methods are fast but not robust since the search process is very dependent on the starting point and rarely converges to a global minimum. The *gradient-based methods* require that the mathematical function that links functional requirements and geometric shape should be known, and that continuity and derivability conditions should be met. Such a function is very difficult to define in the fields here considered.

The *global-based methods* can achieve a global optimum solution without a mathematical map between functional requirements and geometric shape; they iteratively choose a new solution, trying to reduce the value of the objective function (or functions) by changing the characteristic parameters within the admissible region. The most popular *global-based methods* fall within two categories: *evolutionary algorithm* (*EA*) and *Artificial Neural Network* (*ANN*) based algorithm. Under the category of *EA*, mention should be made of the *Genetic Algorithms* (*GAs*) [30]. *GAs* are considered reliable (in the sense of finding solutions which are very close to the global optimum), robust (in the sense of avoiding local optima) and capable to handle multiple objectives. For these reasons, *GAs* are used for many shape optimisation