Neurocomputing 142 (2014) 39–47

Contents lists available at ScienceDirect

Neurocomputing

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

Evolving parametric aircraft models for design exploration and optimisation

Jonathan Byrne*, Philip Cardiff, Anthony Brabazon, Michael O'Neill

University College Dublin, Belfield, Dublin 4, Ireland

ARTICLE INFO

Article history:
Received 15 November 2013
Received in revised form
22 April 2014
Accepted 23 April 2014
Available online 6 May 2014

Keywords:
Parametric design
Optimisation
Evolutionary algorithms
Computational fluid dynamics

ABSTRACT

Traditional CAD tools generate a static solution to a design problem. Parametric systems allow the user to explore many variations on that design theme. Such systems make the computer a generative design tool and are already used extensively as a rapid prototyping technique in architecture and aeronautics. Combining a design generation tool with an analysis software and an evolutionary algorithm provides a methodology for optimising designs. This work combines NASA's parametric aircraft design tool (OpenVSP) with a fluid dynamics solver (OpenFOAM) to create and analyse aircraft. An evolutionary algorithm is then used to generate a range of aircraft that maximise lift and reduce drag while remaining within the framework of the original design. Our approach allows the designer to automatically optimise their chosen design and to generate models with improved aerodynamic efficiency. Different components on three aircraft models are varied to highlight the ease and effectiveness of the parametric model optimisation.

1. Introduction

Parametric systems are changing the conceptual design process in the same way as spreadsheets changed finance. Both operate on the same principle. The user defines the relationships in a system and then changes variables in that system to rapidly explore alternative possibilities. Instead of manually creating a CAD model by dragging and dropping components, the parametric design is specified using variables and functions. Just as changing the value in a cell causes the spreadsheet to recalculate all related values, changing a variable that defines part of a model will adapt all the connected components so as to maintain a coherent design. Although there is a longer lead time to implement the initial model, once it is encoded the user can easily create endless variations on the original.

Evolutionary algorithms (EAs) have shown their ability to optimise the shape and form of designs [1,2]. One of the primary considerations when applying an evolutionary algorithm to a design problem is the representation used. The representation limits the search space by defining all the designs the algorithm could possibly generate. Poor representations generate designs that are invalid (internal faces, unconnected parts), infeasible (wrong scale) or missing the desired functionality. Creating a suitable representation is a difficult task that requires knowledge of both programming and of the specific domain.

Parametric systems provide a novel solution to the representation problem. A well-implemented parametric system will only generate valid designs and incorporates domain knowledge. It also allows a designer with no formal programming experience to define the representation for the evolutionary algorithm. The designer provides the initial model and specifies the range limits so as to generate appropriate variations of their design. Parametric models make evolutionary optimisation directly accessible to the designer and allows them to use their domain knowledge to create a representation that generates feasible designs.

This work combines NASA's parametric aircraft system (OpenVSP) and a computational fluid dynamics solver (OpenFOAM) with an evolutionary algorithm to generate a variety of optimised and novel designs. Section 2 gives an overview of parametric design systems and their application in industry. Section 3 describes the fluid dynamics solver used to generate the fitness values for the model. Section 4 discusses previous aircraft optimisation examples that used evolutionary approaches. Three parametric aircraft models are optimised in this work. The settings consistent for all the experiments are shown in Section 5. Section 6 describes the experiments carried out on the blended wing body model where the airfoil and the wing were varied. Section 7 describes the experiments carried out on the Cessna 182 model where the wing was exclusively varied. Section 8 describes the experiments carried out on the MIG 21 model where the wing and the tail section were optimised simultaneously.

* Corresponding author. Tel.: +353 863257989.
E-mail address: jonathan.byrne@ucd.ie (J. Byrne).

http://dx.doi.org/10.1016/j.neucom.2014.04.004
0925-2312/

Finally Sections 9 and 10 discuss the results of the experiments and the conclusions that can be drawn from them respectively.

2. Parametric design

Parametric design defines the relationships between components in a design. Generating a model consisting of hierarchical and geometric relations allows for exploration of possible variations on the initial design while still limiting the search space. Instead of manually placing and connecting components as is done in traditional CAD, component generating algorithms are linked with user definable variables. Defining the relationship between the components prevents invalid design generation. A change to one component will automatically effect a change on any connected component.

Parametric systems traditionally consist of basic components tailored for a particular design problem. An example of this would be the wing, fuselage and engine components in OpenVSP. Predefined components allow for domain knowledge to be embedded in the software and simplifies the design process. Although the user can explicitly define design components by programming them, normally model creation is done by combining existing components using a graphical interface. Many parametric design systems, such as grasshopper [3], are implemented using a drag and drop interface, shown in Fig. 1. The user can then manipulate the input and evaluate the benefit of the component to the overall design. An important aspect of parametric design is that the user observes the effects caused by manipulating a variable in real time, allowing the user to treat the underlying algorithm as a black box. Showing the effect of changing input to the system means that the user does not require an understanding of the underlying mechanics of the system, but instead gives them an intuitive understanding of how the components in a system are related to each other (Fig. 2).

Parametric design tools have now been introduced into mainstream design software. There is the Grasshopper parametric design tool plug-in for the Rhino modelling system [3], Bentley Systems have implemented a program called Generative Components [4] based on the parametric design paradigm and Dassault Systems have developed CATIA, a CAD system combined with a parametric design tool. Parametric functionality was introduced to AutoCAD 2010 to allow for algorithmic manipulation of a design.

Combining parametric systems with structural analysis allows the user to make informed decisions about the geometric alterations during the conceptual design stage [5]. EIFForm is a parametric design system that optimises lattice structures by using a structural analysis and a simulated annealing algorithm. The results have been used to design a structure in the inner courtyard of Schindler house [6]. Bollinger et al. [7] have developed parametric design systems that incorporate structural considerations and have used it to generate roofing structures for the BMW Welt Museum, Munich and the Rolex learning centre, EPFL, Lausanne. CATIA was combined with GSA structural analysis software [8] to evolve roofing structures for a football stadium [5].

The software used in this work is open vehicle sketch pad (OpenVSP). It was originally developed by NASA and Sterling Software as a rapid geometry modeler for conceptual aircraft [9] and has since developed into a stand-alone aircraft modelling tool. It was released as open-source software in 2012 under the NASA open source agreement. This work combines aerodynamic analysis with OpenVSP to analyse the lift and drag of the models. The next section discusses how the aerodynamic analysis was performed and the solver that was used.

3. Computational fluid dynamics

Computational Fluid Dynamics (CFD) uses numerical methods to solve how liquids and gases interact with surfaces. Although the calculations are computationally intensive, the dramatic increase in the power of standard hardware enables basic CFD analysis to be carried out on standard desktop machines. OpenFOAM (open-source field operations and manipulation) [10] is used as the CFD solver in the experiments. Although primarily used for fluid dynamics simulations, it provides a toolbox of different solving techniques for applications such as combustion, electromagnetism, solid mechanics and heat transfer. It is designed for parallel execution due to the high processor demand of CFD modelling. It is highly extensible and has been adapted for calculating transonic aerodynamics [11], marine cavitation models [12] and orthotropic solid mechanics [13].

The solver used in the experiments is the semi-implicit method for pressure linked equations (SIMPLE) algorithm [14]. It is a steady state numerical solver for efficiently solving the Navier–Stokes equations that describe fluid motion. The algorithm forms the basis of CFD software and has been adapted to calculate the transfer of mass and momentum in a discretised three dimensional environment. The solver iteratively calculates the pressure and the velocity within the system. Post-processing then calculates the lift and drag forces generated by the model and these are used as the fitness value.

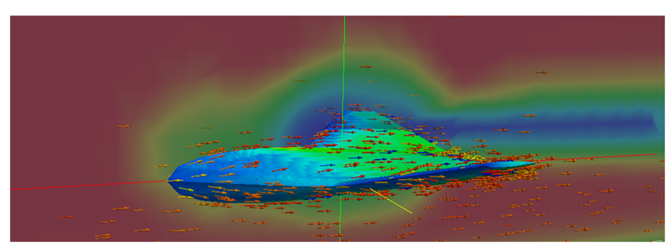

Fig. 2. The relative wind velocity and turbulence caused by the blended wing body model.

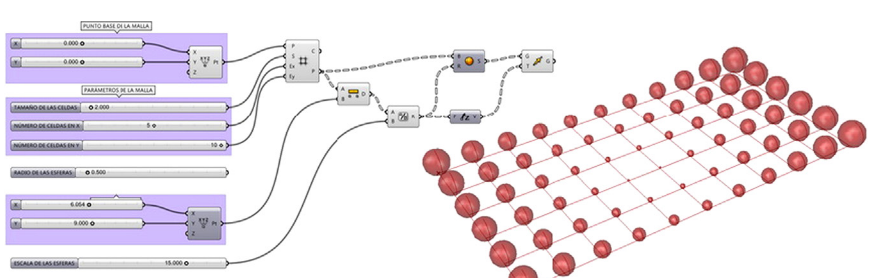

Fig. 1. The GUI for the Grasshopper parametric system. The variables are shown in the purple boxes on the left and are connected to the shape generating functions. The output design is on the right. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)