Expert Systems with Applications 39 (2012) 10700–10708

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

Efficient aerodynamic design through evolutionary programming and support vector regression algorithms

E. Andrés [a], S. Salcedo-Sanz [b,*], F. Monge [a], A.M. Pérez-Bellido [b]

[a] Fluid Dynamics Branch, National Institute for Aerospace Technology (INTA), Spain
[b] Department of Signal Theory and Communications, Universidad de Alcalá, Spain

ARTICLE INFO

Keywords:
Aerodynamic design
Airfoil optimization
Evolutionary optimization
Support vector regression algorithms
Aerodynamic coefficient prediction
Computational fluid dynamics

ABSTRACT

The shortening of the design cycle and the increase of the performance are nowadays the main challenges in aerodynamic design. The use of evolutionary algorithms (EAs) seems to be appropriate in a preliminary phase, due to their ability to broadly explore the design space and obtain global optima. Evolutionary algorithms have been hybridized with metamodels (or surrogate models) in several works published in the last years, in order to substitute expensive computational fluid dynamics (CFD) simulations. In this paper, an advanced approach for the aerodynamic optimization of aeronautical wing profiles is proposed, consisting of an evolutionary programming algorithm hybridized with a support vector regression algorithm (SVMr) as a metamodel. Specific issues as precision, dataset training size and feasibility of the complete approach are discussed and the potential of global optimization methods (enhanced by metamodels) to achieve innovative shapes that would not be achieved with traditional methods is assessed.

1. Introduction

The challenges of the aeronautical industry in the near future will require new computational tools for the design of the type of aircraft that will be demanded by the European industry, according to the guidelines stated at the ACARE 2020 (Argüelles et al., 2001) and 2050 (ACARE Advisory Council for Aeronautics Research in Europe, 2010) flight paths. The aeronautical industry agrees that these objectives make necessary the design of an innovative aircraft shape rather than further local improvements in the traditional wing-body-tail configuration. Efficient and accurate shape design optimization tools, able to consider novel concepts through the use of flexible geometry parametrization, are becoming a must for the aeronautical industry.

The aerodynamic design problem can be solved using either deterministic or non deterministic methods. Deterministic approaches often require the gradient information of the objective function. These gradient-based methods have been broadly used but they need a continuous evaluation function and have a weak performance in a noisy environment. In addition, they are strongly dependent on the initial configuration and could get trapped into a local minimum. On the other hand, non-deterministic methods such as evolutionary algorithms (EAs) have the ability to work with noisy objective functions, without assumptions on continuity (Lian, Oyama, & Liou, 2010). They also have a high potential to find the global optimum of complex problems involving a large amount of design parameters. However, they require a vast number of evaluations to obtain the optimum solution, even for a small number of design variables.

In the case of aerodynamic design, each evaluation of an individual in the EA requires a complete CFD analysis which makes the method unfeasible, in terms of computational cost. To overcome this problem there are different approaches in the literature, such as the use of powerful processing machines, such as Graphic Processing Units (Kampolis, Trompoukis, Asouti, & Giannakoglou, 2010) or, more frequently, the use surrogate models or metamodels (Giannakoglou, 2002; Jin, 2005; Newman, Taylor, Barnwell, Newman, & Hou, 1999; Zhong-Hua, Zimmermann, & Görtz, 2010). A metamodel is an inexpensive and approximated model of a costly evaluation method. Regarding the aerodynamic design using EAs, the metamodel technique could be used to calculate the fitness of the candidate solutions by replacing the time demanding CFD tools, as previously shown in the literature (Giannakoglou, Papadimitriou, & Kampolis, 2006; Liakopoulos, Kampolis, Giannakoglou, & enabled, 2008). For this purpose, regressors based on neural computation could be used as metamodels, once they have been trained based on previous evaluations.

There are well-documented examples of the applicability of soft-computing approaches (neural networks and evolutionary-based techniques) in a broad range of prediction and optimization problems including some parts of aerodynamic or multidisciplinary optimization processes. The majority of published studies uses some type of evolutionary algorithms hybridized with neural

* Corresponding author. Address: Department of Signal Theory and Communications, Universidad de Alcalá, 28871 Alcalá de Henares, Madrid, Spain. Tel.: +34 91 885 6731; fax: +34 91 885 6699.
E-mail address: sancho.salcedo@uah.es (S. Salcedo-Sanz).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.02.197

networks as metamodels: in Giannakoglou et al. (2006) a complete study of different types of neural networks working as metamodels in an aerodynamic shape design problem is carried out. The study includes multilayer perceptrons and radial basis functions networks. In Liakopoulos et al. (2008) a grid-based hierarchical evolutionary algorithm hybridized with a radial basis function network is proposed also in different parts of aerodynamic design problems. There are more recent works discussing different aspect of hybridizing evolutionary algorithms and neural networks as metamodels for airfoil design (Asouti & Giannakoglou, 2009; Bompard, Peter, & Desideri, 2010; Cohen, Siegel, Seidel, Aradag, & McLaughlin, 2012; Santos, De Mattos, & Girardi, 2008; Di Stefano & Di Angelo, 2003). Other perspectives of the problem are also discussed in the literature, such as in Jahangirian and Shahrokhi (2011), where an approach based on evolutionary algorithms directly hybridized with an unstructured CFD solver and a neural network (multi-layer perceptron) as metamodel for the first step of the approach is proposed. Other authors have tested the performance of alternative evolutionary approaches such as particle swarm optimization (Khurana, Winarto, & Sinha, 2009; Praveen & Duvigneau, 2009). There are also alternative methods applied to aerodynamic shape design, such as fuzzy logic approaches (Hossain, Rahmanb, Hossen c, Iqbal, & Zahirul, 2011), multiobjective algorithms (Kampolis & Giannakoglou, 2008), works involving cokriging techniques (Zhong-Hua et al., 2010), and papers that describe computation frameworks developed to enhance the design process (Kim et al., 2009).

This work focuses on the first phase of the aerodynamic design process, i.e., obtaining an approximation to the best candidate from a broad design space (dataset of different geometries, including unconventional ones). The aim of this paper is to study the performance of an evolutionary programming approach hybridized with a support vector regression algorithm as metamodel in a problem of optimal airfoil design. To our knowledge, this important regression technique has not been extensively applied to aerodynamic design, and may have important advantages over previously mentioned metamodels, such as neural networks. It will be showed that the proposed approach is able to obtain accurate first airfoil designs which can be used, at a later stage, as input for a detailed design process using methods which requires more computational resources, such as CFD.

This paper is structured as follows: next section describes the proposed hybrid evolutionary programming – SVMr approach, giving details on the EP and SVMr algorithms. Then, the experimental part of the paper is explained, where different results on the SVMr performance as a metamodel are displayed. Finally, some final remarks on the feasibility of the proposed approach in case of industrial configurations are outlined.

2. Proposed approach

The process of the proposed approach is shown in Fig. 1. The objective is the shortening of the design cycle through a combined approach, where the fist stage makes use of evolutionary algorithms together with metamodels to estimate the aerodynamic data. In this phase, the inputs to the process are the target design point (flow conditions), the objective function and the constraints. The output is an approximation to the global optimal solution. As evolutionary algorithm, an evolutionary programming approach (Yao, Liu, & Lin, 1999) is proposed, which will evolve different airfoil geometries, in terms of an objective function, given by the metamodel. The use of a support vector regression (SVMr) is proposed to this end. In this section, the EP algorithm used to tackle the optimal evolution of airfoil geometries is explained, together with a detailed description of the EP encoding and the SVMr, which will be applied to obtain the aerodynamic coefficients associated to each geometry, in a fast and accurate way.

2.1. EP encoding: airfoil parametrization

Sobieczky parametrization (Li, Seebass, & Sobieczky, 1998) is used, which can represent a wide variety of airfoils with a reasonable number of parameters. This parametrization employs mathematical expressions for the proper representation of generic airfoil geometry (shape functions). This is accomplished by the use of polynomial functions for the airfoil thickness (y_t) and camber (y_c) lines:

$$y_t = a_1\sqrt{(x)} + a_2x + a_3x^2 + a_4x^3 + a_5x^4 \tag{1}$$

$$y_c = b_1x + b_2x^2 + b_3x^3 + b_4x^4 + b_5x^5 + b_6x^6 \tag{2}$$

The upper-and lower-side y-coordinates at a given chord location are given by:

$$y_u = y_t + y_c \tag{3}$$

$$y_l = y_t - y_c \tag{4}$$

The geometric parameters defining the airfoil are: position of the leading edge control point, position and airfoil maximum thickness, trailing edge thickness line angle, trailing edge thickness, leading edge camber line angle, camber at maximum thickness, position and maximum camber, camber at maximum thickness, trailing edge camber line angle and trailing edge camber. Fig. 2 shows the parameters used for airfoil definition. Note that the mean curvature line (colored in red) has been multiplied by 10 only for representation purpose.

It is possible to link a_n and b_n coefficients in Eqs. (1) and (2) to the geometric variables described in Table 1 as it is shown below. Using this parametrization, an airfoil shape is defined by basic geometric parameters, instead of the coefficient of shape functions directly. This provides more knowledge about the flow around the airfoil and therefore, about the aerodynamic performance.

To obtain the a_n coefficients related to thickness distribution Eqs. (5)–(9) are used:

$$0 = \frac{a_1}{2\sqrt{(xtth)}} + a_2 + 2a_3xtth + 3a_4(xtth)^2 + 4a_5(xtth)^3 \tag{5}$$

$$ytth = a_1\sqrt{(xtth)} + a_2xtth + a_3(xtth)^2 + a_4(xtth)^3 + a_5(xtth)^4 \tag{6}$$

$$atte = \frac{a_1}{2} + a_2 + 2a_3 + 3a_4 + 4a_5 \tag{7}$$

$$ytte = a_1 + a_2 + a_3 + a_4 + a_5 \tag{8}$$

$$ytle = a_1\sqrt{(xtle)} + a_2xtle + a_3(xtle)^2 + a_4(xtle)^3 + a_5(xtle)^4 \tag{9}$$

Eqs. (10)–(14) are used to compute the b_n coefficients for camber:

$$acle = b_1 \tag{10}$$

$$ycth = b_1xtth + b_2(xtth)^2 + b_3(xtth)^3 + b_4(xtth)^4 + b_5(xtth)^5 + b_6(xtth)^6 \tag{11}$$

$$0 = b_1 + 2b_2xcmc + 3b_3(xcmc)^2 + 4b_4(xcmc)^3 + 5b_5(xcmc)^4 + 6b_6(xcmc)^5 \tag{12}$$

$$acte = b_1 + 2b_2 + 3b_3 + 4b_4 + 5b_5 + 6b_6 \tag{13}$$

$$ycte = b_1 + b_2 + b_3 + b_4 + b_5 + b_6 \tag{14}$$

The geometries used for training and validation, are generated from variation of these design variables, within the considered ranges displayed in Table 1. As it can be observed in the table, the leading edge control point, the trailing edge thickness and the trailing edge camber are maintained constant in order to compare the results with previous work (Santos et al., 2008). Therefore, two or three values in the range of each of the remaining eight geometric variables are used to generate the database of 5000 geometries. Fig. 3(a) shows the airfoils defined by the minimum, maximum and averaged value in each of the geometric variables. A huge set of airfoils are included in the database in order to exploit the

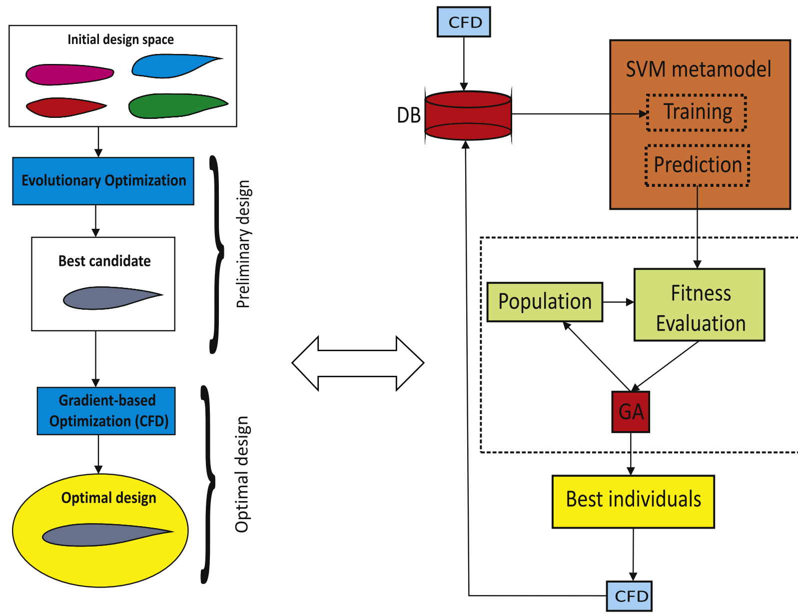

Fig. 1. Proposed two-step design cycle: first step using evolutionary algorithms together with metamodels and final step for fine optimization.

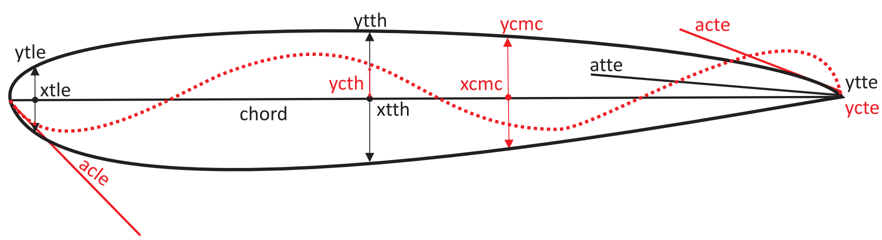

Fig. 2. Sobieczky's parametrization for airfoil definition.

Table 1
Variables for airfoil geometry parametrization and their values' range.

Short name	Variable for geometry parametrization	Min	Max
xtle	*X* Position for leading edge control point	0.015	0.015
ytle	*Y* Position for leading edge control point	0.036	0.036
xtth	*X* position for maximum thickness	0.300	0.450
ytth	Maximum thickness	0.100	0.170
atte	Trailing edge thickness line angle	−10.000	−4.500
ytte	Trailing edge thickness	0.006	0.006
acle	Leading edge camber line angle	−7.500	5.000
ycth	Camber at maximum thickness	−0.008	0.005
xcmc	*X* position for maximum camber	0.700	0.800
ycmc	Maximum camber	−0.010	0.020
acte	Trailing edge camber line angle	−15	0
ycte	Trailing edge camber	0	0

potential of the evolutionary optimization methods to broadly explore the design space and find the global optimum. Unconventional airfoils are also included in the database to be considered in the optimization process. Fig. 3(b) shows examples of ten of the airfoils included in the data base and their geometric variables are displayed in Table 2.

2.2. Evolutionary programming

Evolutionary algorithms (Fogel, 1994; Yao et al., 1999), are robust problems' solving techniques based on natural evolution processes. They are population-based techniques which codify a set of possible solutions to the problem, and evolve it through the application of the so called *evolutionary operators* (Goldberg, 1989). Among EAs, evolutionary programming (EP) approaches are usually applied to continuous optimization problems. This algorithm is characterized by only using mutation and selection operators (no crossover is applied). Several versions of the algorithm have been proposed in the literature: The classical evolutionary programming algorithm (CEP) was first described in the work by Bäck and Schwefel (1993), and analyzed later by Yao et al. (1999). It is used to optimize a given function $f(\mathbf{x})$ (ψ or φ in our case), i.e. obtaining $\mathbf{x}_o$ such that $f(\mathbf{x}_o) < f(\mathbf{x})$, with $\mathbf{x} \in [lim\_inf, lim\_sup]$. The CEP algorithm performs as follows:

1. Generate an initial population of μ individuals (solutions). Let t be a counter for the number of generations, set it to $t = 1$. Each individual is taken as a pair of real-valued vectors $(\mathbf{x}_i, \boldsymbol{\sigma}_i)$, $\forall i \in \{1, \ldots, \mu\}$, where $\mathbf{x}_i$'s are objective variables, and $\boldsymbol{\sigma}_i$'s are standard deviations for Gaussian mutations.
2. Evaluate the fitness value for each individual $(\mathbf{x}_i, \boldsymbol{\sigma}_i)$ (using the problem's objective function, ψ or φ).
3. Each parent $(\mathbf{x}_i, \boldsymbol{\sigma}_i)$, $\{i = 1, \ldots, \mu\}$ then creates a single offspring $(\mathbf{x}'_i, \boldsymbol{\sigma}'_i)$ as follows:

$$\mathbf{x}'_i = \mathbf{x}_i + \boldsymbol{\sigma}_i \cdot \mathbf{N_1}(\mathbf{0}, \mathbf{1}) \quad (15)$$

$$\boldsymbol{\sigma}'_i = \boldsymbol{\sigma}_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot \mathbf{N}(\mathbf{0}, \mathbf{1})) \quad (16)$$

where $N(0,1)$ is an unidimensional normal distribution with mean zero and standard deviation one. The parameters τ and τ' are commonly set to $\left(\sqrt{2\sqrt{n}}\right)^{-1}$ and $\left(\sqrt{2n}\right)^{-1}$, respectively (Yao et al., 1999), where n is the length of the individuals.

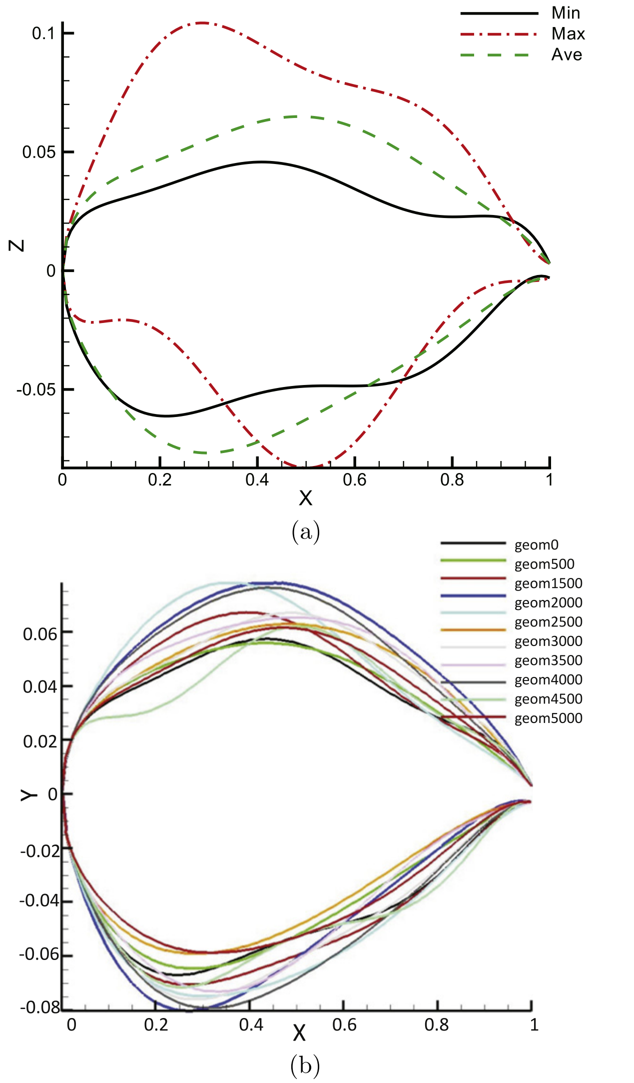

Fig. 3. Examples of airfoils in the training database; (a) maximum and minimum configurations in the data base; (b) example of ten airfoils in the database.

4. If $x_i(j) > lim\_sup$ then $x_i(j) = lim\_sup$ and if $x_i(j) < lim\_inf$ then $x_i(j) = lim\_inf$.
5. Calculate the fitness values associated with each offspring $(\mathbf{x}'_i,\ \boldsymbol{\sigma}'_i), \forall i \in \{1, \cdots, \mu\}$.
6. Conduct pairwise comparison over the union of parents and offspring: for each individual, p opponents are chosen uniformly at random from all the parents and offspring. For each comparison, if the individual's fitness is better than the opponent's, it receives a "win".
7. Select the μ individuals out of the union of parents and offspring that have the most "wins" to be parents of the next generation.
8. Stop if the halting criterion is satisfied, and if not, set $t = t + 1$ and go to Step 3.

A second version of the algorithm is the so called fast evolutionary programming (FEP). The FEP was described and compared with the CEP in Yao et al. (1999). The FEP is similar to the CEP algorithm, but it performs a mutation following a Cauchy probability density function, instead of a Gaussian based mutation. The one-dimensional Cauchy density function centered at the origin is defined by:

$$f_t(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2} \tag{17}$$

where $t > 0$ is a scale parameter (Yao et al., 1999). Using this probability density function, the FEP algorithm is obtained by substituting step 3 of the CEP, by the following equation:

$$\mathbf{x}'_i = \mathbf{x}_i + \boldsymbol{\sigma}_i \cdot \boldsymbol{\delta} \tag{18}$$

where $\boldsymbol{\delta}$ is a Cauchy random variable vector with the scale parameter set to $t = 1$.

Finally, in Yao et al. (1999) the *improved FEP* (IFEP) is also proposed, where the best result obtained between the Gaussian mutation and the Cauchy mutation is selected to complete the process.

Note that, in the design problem to be considered in this paper, each vector $\mathbf{x}$ is composed by a given parametrization of an airfoil geometry, i.e., $\mathbf{x} = [xtle, ytle, xtth, ytth, atte, \ldots, ycte]$. On the other hand, the objective function f to be optimized is given by the airfoil performance, that in this case will be modeled using a support vector machine approach (that acts as a meta-model), as it is described in the next subsection.

2.3. Objective function: approximation using a support vector regression algorithm

One of the most important statistic models for prediction are the support vector regression algorithms (SVMr) (Smola, Murata, Scholkopf, & Muller, 1998; Smola & Schölkopf, 1998). The SVMrs are appealing algorithms for a large variety of regression problems, in many of them mixed with evolutionary computation algorithms (Cheng, Chen, & Huang, 2011; Jiang & He, 2012; Salcedo-Sanz, Ortiz-García, Pérez-Bellido, Portilla-Figueras, & Prieto, 2011). Although there are several versions of SVMr, in this case the classic model presented in Smola et al. (1998) will be described.

The ϵ-SVMr method for regression consists of training a model of the form $y(\mathbf{x}) = f(\mathbf{x}) + b = \mathbf{w}^T\phi(\mathbf{x}) + b$, given a set of training vectors $C = \{(\mathbf{x}_i, y_i), i = 1, \ldots, l\}$, to minimize a general risk function of the form

Table 2
Geometric variables of different airfoils in the training database.

Geometry	*xtth*	*ytth*	*atte*	*acle*	*ycth*	*xcmc*	*ycmc*	*acte*
0	0.337	0.117	−8.625	−4.375	−0.004	−0.725	−0.002	−11.250
500	0.337	0.117	−5.875	−4.375	−0.004	−0.750	−0.005	−3.750
1000	0.337	0.135	−7.250	−4.375	−0.001	−0.725	−0.002	−7.5
1500	0.337	0.152	−8.625	−4.375	−0.001	0.750	0.012	−11.250
2000	0.337	0.152	−5.875	−4.375	−0.001	0.725	−0.002	−3.750
2500	0.375	0.117	−7.250	−4.375	−0.001	0.750	0.012	−7.5
3000	0.375	0.135	−8.625	−1.250	−0.004	0.725	0.005	−11.250
3500	0.375	0.135	−5.875	−1.250	0.004	0.750	0.012	−3.750
4000	0.375	0.152	−7.250	−1.250	0.001	0.725	0.005	−7.500
4500	0.412	0.117	−8.625	−1.250	0.001	0.775	−0.002	−11.250
5000	0.412	0.117	−5.875	−1.250	0.001	0.725	0.005	−7.5

$$R[f] = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} L(y_i, f(\mathbf{x})) \tag{19}$$

where $\mathbf{w}$ controls the smoothness of the model, $\phi(\mathbf{x})$ is a function of projection of the input space to the feature space, b is a parameter of bias, $\mathbf{x}_i$ is a feature vector of the input space with dimension N, y_i is the output value to be estimated and $L(y_i, f(\mathbf{x}))$ is the loss function selected. In this paper, we use the L1-SVMr (L1 support vector regression), characterized by an ϵ-insensitive loss function (Smola & Schölkopf, 1998)

$$L(y_i, f(\mathbf{x})) = |y_i - f(\mathbf{x_i})|_\epsilon \tag{20}$$

In order to train this model, it is necessary to solve the following optimization problem (Smola & Schölkopf, 1998):

$$\min\left(\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right) \tag{21}$$

subject to

$$y_i - \mathbf{w}^T\phi(\mathbf{x_i}) - b \leqslant \epsilon + \xi_i, \quad i = 1, \ldots, l \tag{22}$$

$$-y_i + \mathbf{w}^T\phi(\mathbf{x_i}) + b \leqslant \epsilon + \xi_i^*, \quad i = 1, \ldots, l \tag{23}$$

$$\xi_i, \xi_i^* \geqslant 0, \quad i = 1, \ldots, l \tag{24}$$

The dual form of this optimization problem is usually obtained through the minimization of the Lagrange function, constructed from the objective function and the problem constraints. In this case, the dual form of the optimization problem is the following:

$$\max\left(-\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)\left(\alpha_j - \alpha_j^*\right)K(\mathbf{x_i}, \mathbf{x_j}) - \epsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*)\right) \tag{25}$$

subject to

$$\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \tag{26}$$

$$\alpha_i, \alpha_i^* \in [0, C] \tag{27}$$

In addition to these constraints, the Karush–Kuhn–Tucker conditions must be fulfilled, and also the bias variable, b, must be obtained. The complete process is not detailed here for simplicity, the interested reader can consult (Smola & Schölkopf, 1998) for reference. In the dual formulation of the problem the function $K(\mathbf{x_i}, \mathbf{x_j})$ is the kernel matrix, which is formed by the evaluation of a kernel function, equivalent to the dot product $\langle\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\rangle$. A usual selection for this kernel function is a Gaussian function, as follows:

$$K(\mathbf{x_i}, \mathbf{x_j}) = exp(-\gamma \cdot \|\mathbf{x_i} - \mathbf{x_j}\|^2). \tag{28}$$

The final form of function $f(\mathbf{x})$ depends on the Langrange multipliers α_i, α_i^*, as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)K(\mathbf{x_i}, \mathbf{x}) \tag{29}$$

In this way it is possible to obtain a SVMr model by means of the training of a quadratic problem for given hyper-parameters C, ϵ and γ. However, obtaining these parameters is not a simple procedure, being necessary the implementation of search algorithms to obtain the optimal ones or the estimation of them (Ortiz-Garcıa, Salcedo-Sanz, Pérez-Bellido, & Portilla-Figueras, 2009).

Given the amount of aerodynamic data available to train the SVMr-based model, it was necessary to split them into several groups, each corresponding to a different value of the angle of attack. Subsequently, each of these datasets is employed to train a different model, which will be associated to its corresponding angle. In case of values different from those 24 angles existing in the original data, a linear interpolation of the two nearest models has been carried out. Fig. 4 shows the architecture of the proposed SVMr model. Note that the problem is tackled with a network of SVMr banks, defining a single SVMr for each angle of attack, both in the train and test periods.

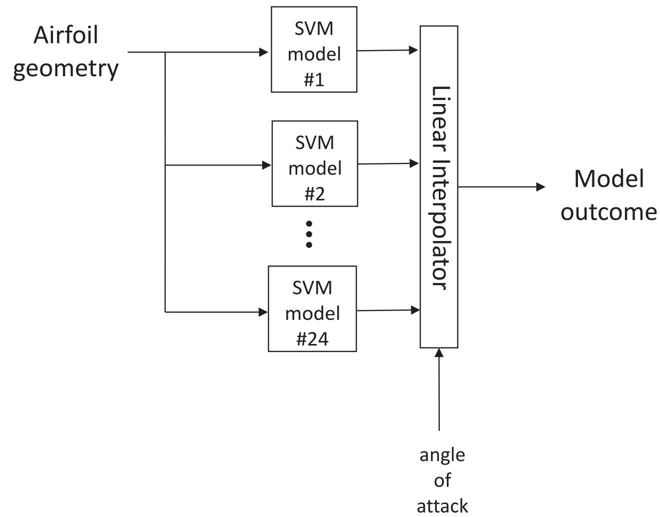

Fig. 4. SVMr banks architecture applied in this work.

3. Experiments and results

3.1. Experiments on the metamodel obtention (SVMr)

In this work, only subsonic computations were considered for both prediction and airfoil design optimization. The panel code XFOIL (Drela, 1989) was used to generate the necessary data. In addition, the first steps in the prediction of aerodynamic coefficients of transonic configurations have been performed using the RANS TAU code and preliminary results will be mentioned. The data base is comprised of 5000 geometries and their related aerodynamic coefficients (drag, lift and momentum coefficients). In particular, 4000 geometries and their related data were used for training, and the other 1000 geometries were used as validation tests to measure the precision. The chosen Mach and Reynolds numbers for the dataset samples are 0.3 and 10^7 respectively. The angle of attack is ranging from 4° to 14° and free transition is used.

Once the network has been trained, a set of 1000 geometries is used for validation purposes. The airfoil geometries used in the validation were obtained in a random way from the initial data set. Fig. 5 shows a comparison of C_l curve (a) and drag polar (b) predicted and the ones calculated with XFOIL for one of the validation geometries. Note that there is a very good agreement between the predicted and calculated curves. Indeed, these values could be predicted by the neural network with a small mean error as it is showed in the next section.

The error of the network in the prediction of the aerodynamic coefficients (drag, lift and momentum) was analyzed with all the geometries selected for validation. Table 3 shows the root mean square error (RMSE) and the maximum error for each coefficient. At a first glance, the error seems to be high because the maximum error for the lift coefficient reaches a value higher than 200 lift counts, but the RMSE provides a more reasonable value. It is clear that all the errors could be reduced by increasing the training dataset size, but precision should be balanced together with time constraints. The relation between the training dataset size and precision is briefly analyzed later.

Fig. 7(a) shows the histogram of the error in predicting the drag coefficient, and reflects how most of the samples have an error

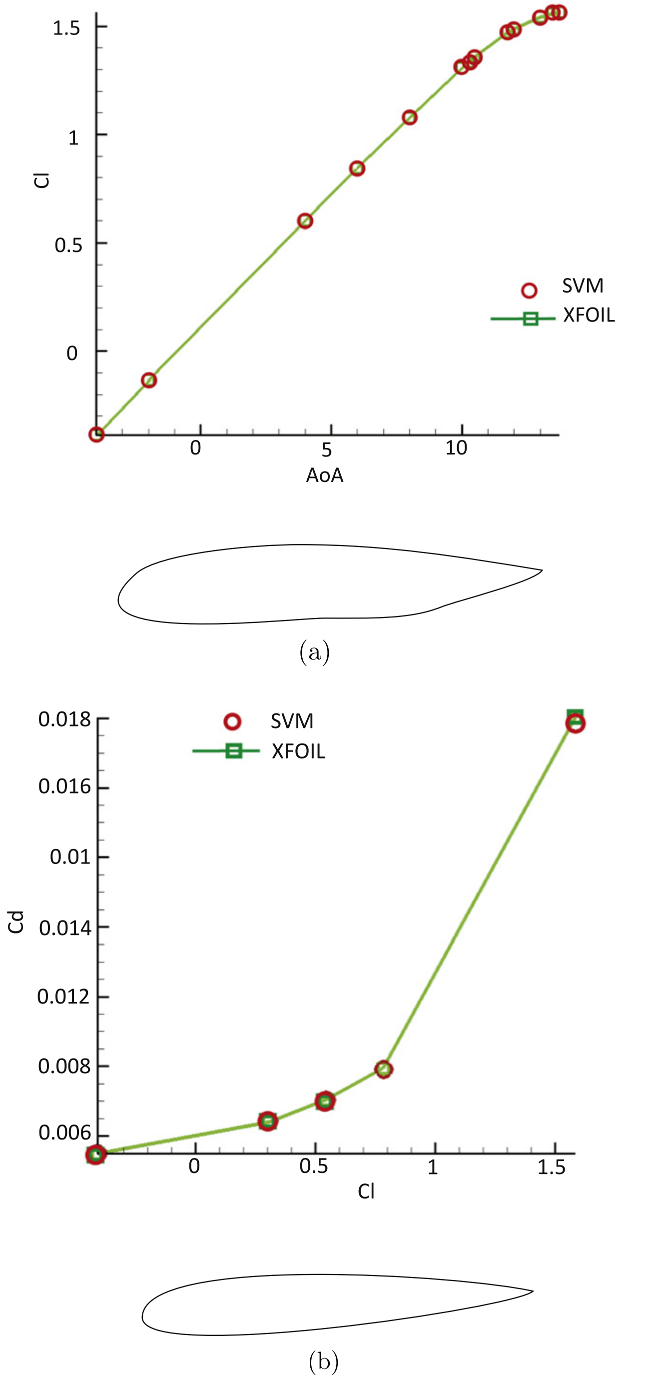

Fig. 5. Coefficients prediction for the displayed airfoils; (a) Cl curve prediction; (b) Cd curve prediction.

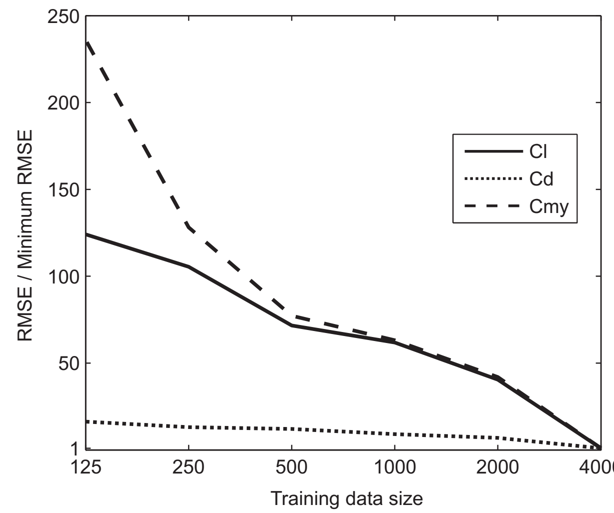

Fig. 6. SVMr performance in terms of the training data size.

Table 3
Maximum error and RMSE for lift, drag and momentum coefficients in subsonic configurations.

	RMSE	Maximum error
Cl	0.00475	0.20473
Cd	0.00055	0.03380
Cm	0.00099	0.00756

close to zero, and only isolated samples have a significant error. Fig. 7(b) shows the RMSE for the prediction of drag, lift and momentum coefficients when using different angles of attack. It can be observed that the maximum RMSE occurs for lift coefficient prediction in high angles of attack (10–14 degrees), close to the stall, but the values are still lower than 9×10^{-3} which could be considered reasonable for a fast prediction method.

Regarding the precision of the proposed approach, several experiments have been carried out in order to analyze how the obtained results depend on the training data size. As it could be expected, the achieved accuracy is reduced to some extent as the number of patterns employed to train the network decreases. To illustrate this effect, Fig. 6 represents the performance provided by the metamodel (SVMr) respect to the training data size. Specifically, the ratio between the RMSE produced at each point and the best RMSE value (achieved with the maximum number of patterns, 4000) is plotted. It is interesting to note how each of the considered aerodynamic coefficients behaves on a different way when the number of training samples decreases. While C_l maximum error only increases about 15 times as training data sets are shortened, C_{my} seems to be much more sensitive to reductions in the amount of patterns, since RMSE is increased by a factor of 230 in this case.

3.2. Inverse airfoil design given subsonic flow conditions

In this section, the complete proposed approach given in Fig. 1 is applied to the preliminary optimization phase of airfoils. The flow conditions are set to Mach = 0.3, AoA = 2°, $Re = 10^6$ and the network is asked for the optimal airfoil in such design point. The objective function has been set as the efficiency (Cl/Cd) to be maximized. This optimization problem can be considered as an inverse design because the evolutionary approach returns an approximation to the optimal airfoil for such flow conditions. The precision of this approximation will be related to the precision of the SVMr used as the metamodel to estimate the objective function of the evolutionary algorithm (Cl/Cd). Comparing the airfoil provided by the SVMr model to those airfoils employed along the training phase and their Cl/Cd XFOIL values (the best of them are drawn in Fig. 9), only 0.15% of these were able to outperform the proposed profile. This point agrees with the distribution function of the training profiles and its corresponding cumulative distribution, which are shown in Fig.8. Both plots include a red arrow, indicating the value $Cl/Cd = 88.72$, which corresponds to the XFOIL computation of the airfoil obtained through the evolutionary method (the SVMr predicted value was $Cl/Cd = 96.60$). As can be seen, this value is placed at the end of the distribution tail, whereas there exist a

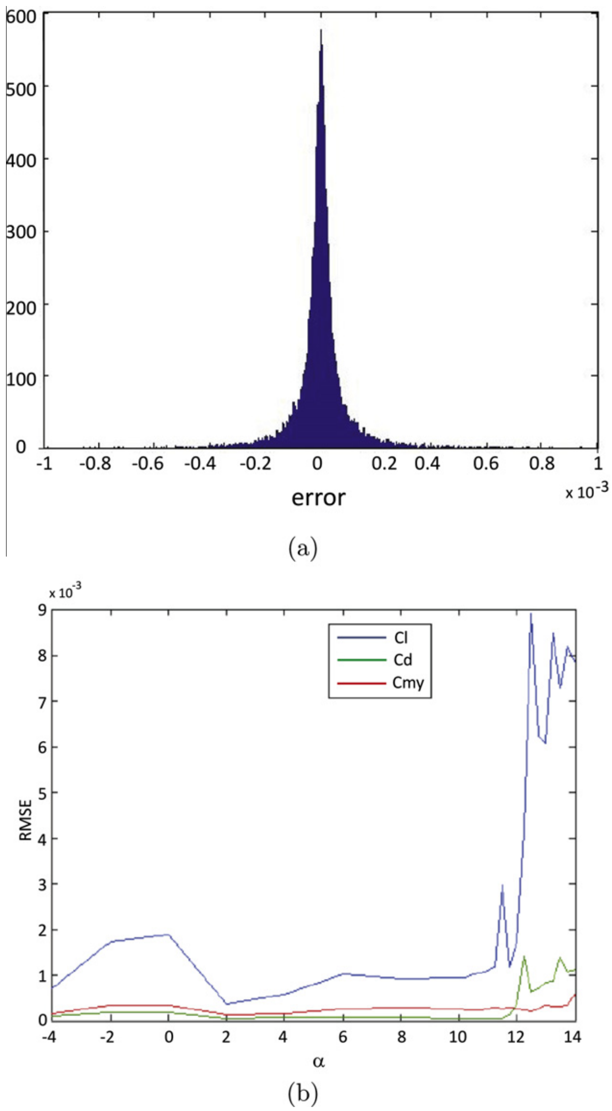

Fig. 7. Coefficients prediction characteristics; (a) histogram of the error in predicting the drag coefficient; (b) RMSE for the prediction of drag, lift and momentum coefficients when using different angles of attack.

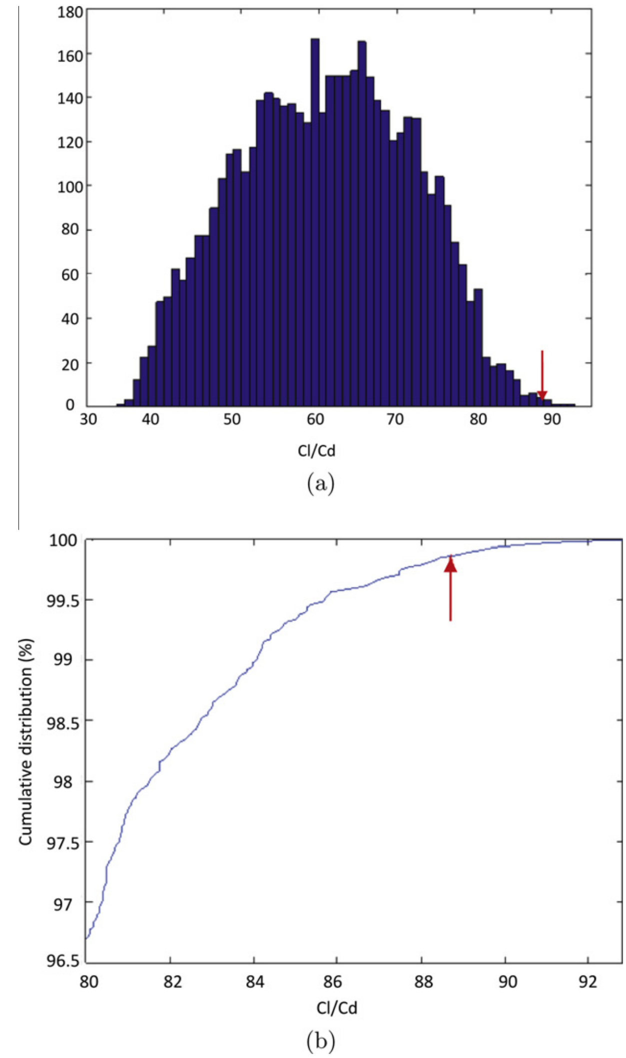

Fig. 8. Distribution function of the training profiles and its corresponding cumulative distribution; (a) distribution function; (b) cumulative function.

99.85% of training patterns providing lower performance, when considering XFOIL computations.

A further flow analysis, around the optimal geometry, returned by SVMr, provides the results in Tables 4. Fig. 9 shows the best 5 geometries for the considered target (including the one returned from the SVM model) and their XFOIL *Cl*/*Cd* value. These geometries have the same value for all the design parameters, except for the trailing edge thickness line angle, leading edge camber line angle and position for maximum camber which differ slightly. Therefore, the returned airfoil, whose geometric parameters are displayed in Table 5, could be a good starting point, close to the optimal solution, for detailed design.

This approach allows extensively exploring the design space, without any dependence on an initial solution and expensive CFD computations, but as it uses a metamodel to estimate the aerodynamic coefficients, the return geometry is also an approximation to the optimal geometry, and therefore, a further detailed design using gradient-basted methods should be applied. For completeness purpose, the flow conditions are set to Mach = 0.3, $Re = 10^6$, as in the previous case, but now the angle of attack is varying from 0° to 10°, and the network is asked for the optimal

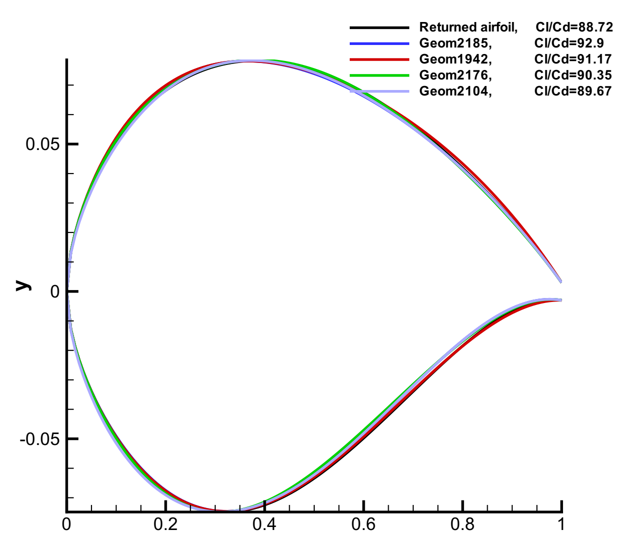

Fig. 9. Best 5 geometries obtained in the optimization process (including the one returned from the SVMr model) and their XFOIL *Cl*/*Cd* value.

Table 4
Aerodynamic results for the returned airfoil after the optimization process.

Cl	Cd	Cm	L/D	Trans. x/c upper part	Trans. x/c lower part
0.495	0.0056	−0.0748	88.7	0.2889	0.4941

Table 5
Geometric variables for the returned airfoil after the optimization process.

xtth	*ytth*	*atte*	*acle*	*ycth*	*xcmc*	*ycmc*	*acte*
0.347	0.152	−6.166	1.384	0.001	0.770	0.011	−7.446

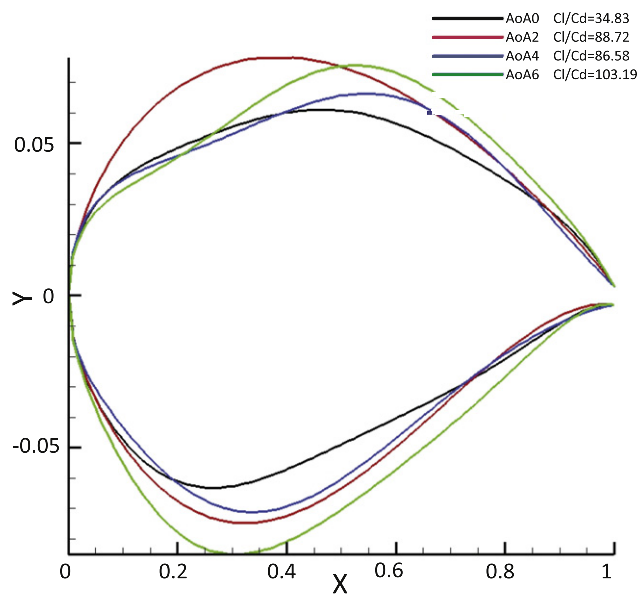

Fig. 10. Several returned airfoils for different angles of attack.

airfoil in each of the design points. The objective function was again the efficiency (*Cl*/*Cd*) to be maximized. Fig. 10 shows the returned airfoils for each of the angles of attack and their aerodynamic properties are displayed in Table 6. It can be observed that the returned airfoil for angles of attack 4° and 6°. Returned airfoils for 8° and 10° are quite similar to 4° and 6°, and due to the small differences with airfoils from previous angles of attack were not visible in this figure. Table 7 shows the geometric variables for these returned airfoils.

As it was mentioned previously, speed and broad exploration of the design space are more important than precision in a preliminary design phase. Therefore, it is necessary to evaluate the computational cost of the proposed approach, in order to support its applicability in such a design step. For the SVM network used as a metamodel to estimate the objective function, an initial training phase is necessary. It is important to remark that this phase is only performed at the beginning of the process and, once the network is trained, it can be used for optimization without any additional cost. Table 8 shows the computational time for each phase of the proposed approach. Note that once the network is trained, it is only 32 s to return the optimized geometry for a given condition.

Table 7
Geometric variables of several returned airfoils for different angles of attack.

AoA	*xtth*	*ytth*	*atte*	*acle*	*ycth*	*xcmc*	*ycmc*	*acte*
0°	0.338	0.118	−8.625	−1.477	−0.001	0.724	0.008	−9.266
2°	0.347	0.152	−6.166	1.384	0.001	0.770	0.011	−7.446
4°	0.412	0.129	−5.874	1.875	−0.003	0.724	0.012	−3.936
6°	0.412	0.129	−5.874	1.874	−0.003	0.725	0.012	−3.936
8°	0.412	0.150	−8.576	−4.374	−0.004	0.765	0.010	−9.978
10°	0.412	0.150	−8.576	−4.374	−0.004	0.765	0.010	−9.978

Table 8
Computational time of each phase.

Phase	Time
Initial training of the SVM network	11,300 s
Evaluation of the objective function (Cl/Cd)	1.42 ms
Optimization	32 s

3.2.1. A short note on predicting the aerodynamic coefficients in transonic configurations

In order to provide a complete overview of the prediction capabilities of the network, the analysis could not be limited to subsonic cases, and therefore, transonic configurations were also preliminary considered. For that purpose, a training data set of around 100 samples was obtained using the DLR TAU code for the NACA0012 profile with Mach ranging from 0.5 to 0.8, angle of attack ranging from 0 to 12 degrees and Reynolds number of 6×10^6. Given the short number of samples considered, a 5-fold cross validation procedure was applied over this set to ensure a meaningful error value. This method consists in splitting the available data in 5 sets, after having randomly shuffled them. Each of these sets is employed once to test the performance of the prediction model that is obtained from the other 4 sets. Then, the final error is obtained by averaging the error values related to each of the 5 models.

Table 9 shows the RMSE and the maximum error for each aerodynamic coefficient prediction. The error measures are one order of magnitude higher compared to the previous case. This is mainly due to the training data set size and these errors could be lowered by increasing the size or redistributing the initial samples. An RMSE of 16 lift counts or 37 drag counts could be reasonable or

Table 6
Aerodynamic results of several returned airfoils for different angles of attack.

AoA	Cl	Cd	Cm	L/D	Trans. x/c upper part	Trans. x/c lower part
0°	0.209	0.006	−0.069	34.83	0.1813	0.3517
2°	0.495	0.005	−0.074	88.7	0.2889	0.4941
4°	0.642	0.007	−0.061	86.58	0.0255	0.5200
6°	0.881	0.008	−0.061	103.02	0.0175	0.5554
8°	1.106	0.010	−0.070	103.19	0.0124	0.5865
10°	1.305	0.013	−0.056	99.71	0.0077	0.7262

Table 9
Maximum error and RMSE for lift, drag and momentum coefficients in transonic configurations.

	RMSE	Maximum error
CL	0.0166	0.0562
CD	0.0037	0.0093
CM	0.0056	0.0137

not, depending on the particular application and its requirement. If computational time constraints are much higher than accuracy constraints, as actually occurs in the preliminary design, then the prediction could be reasonable and accepted. The maximum error is lower than previous case due to the geometry is kept constant.

4. Conclusions

An initial feasibility analysis on the application of evolutionary optimization in the preliminary aerodynamic design has been performed showing promising results. The evolutionary algorithm and the Support Vector Regression algorithm used in the process have been detailed. It has been shown the good performance of the complete approach in different experiments in subsonic and transonic configurations. Future work will address the design of 3D transonic configurations, through the use of the proposed combined approach where the savings in computational time would be relevant. In addition, a further sensitivity study to analyze the influence of the training dataset size and distribution in the prediction accuracy will be addressed.

Furthermore, future activities will also consider other disciplines rather than aerodynamic, in order to exploit the potential of metamodel assisted evolutionary techniques to perform multidisciplinary optimizations (MDO).

Acknowledgement

The research described in this paper made by INTA and UAH researchers has been supported under the INTA activity "Termofluidodinámica" (IGB99001). This work is also partially supported by Spanish Ministry of Science and Innovation, under a project number ECO2010–22065-C03–02. The authors would also like to thank Pablo Bitrián for providing the training geometries.

References

ACARE Advisory Council for Aeronautics Research in Europe. (2010)."Aeronautics and Air Transport: Beyond vision 2020 (towards 2050)", <http://www.acare4europe.org/docs/Towards2050.pdf>.
Argüelles, P., Bischoff, M. et al. (2001). European Aeronautics: A vision for 2020, ACARE <http://www.acare4europe.com/docs/Vision2020.pdf>.
Asouti, V. G., & Giannakoglou, K. C. (2009). A grid-enabled asynchronous metamodel-assisted EA for aerodynamic optimization. *Genetic Programming and Evolvable Machines, 10*(4), 373–389.
Bäck, T., & Schwefel, H. P. (1993). An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation, 1*, 1–23.
Bompard, M., Peter, J., & Desideri, J. (2010). Surrogate models based on function and derivative values for aerodynamic global optimization. *V European Conference on Computational Fluid Dynamics ECCOMAS CFD*.
Cheng, C. S., Chen, P. W., & Huang, K. K. (2011). Estimating the shift size in the process mean with support vector regression and neural networks. *Expert Systems with Applications, 38*(8), 10624–10630.
Cohen, K., Siegel, Stefan, Seidel, J., Aradag, S., & McLaughlin, T. (2012). Nonlinear estimation of transient flow field low dimensional states using artificial neural nets. *Expert Systems with Applications, 39*(1), 1264–1272.
Di Stefano, P., & Di Angelo, L. (2003). Neural network based geometric primitive for airfoil design. *Shape Modeling International*.
Drela, M. (1989). *Xfoil: an analysis and design system for low Reynolds number airfoils. Conference on low Reynolds Numbers Airfoil Aerodynamics*. University of Notre Dame.
Fogel, D. B. (1994). An introduction to simulated evolution. *IEEE Transactions Neural Networks, 5*, 3–14.
Giannakoglou, K. C. (2002). Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences, 38*, 43–76.
Giannakoglou, K. C., Papadimitriou, D. I., & Kampolis, I. C. (2006). Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels. *Computational Methods in Applied Mechanics Engineering, 195*, 6312–6329.
Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning*. Reading, MA: Addison-Wesley.
Hossain, A., Rahmanb, A., Hossen c, J., Iqbal, A. K., & Zahirul, M. I. (2011). Prediction of aerodynamic characteristics of an aircraft model with and without winglet using fuzzy logic technique. *Aerospace Science and Technology, 15*(8), 595–605.
Jahangirian, A., & Shahrokhi, A. (2011). Aerodynamic shape optimization using efficient evolutionary algorithms and unstructured CFD solver. *Computers & Fluids, 46*, 270–276.
Jiang, H., & He, W. (2012). Grey relational grade in local support vector regression for financial time series prediction. *Expert Systems with Applications, 39*(3), 2256–2262.
Jin, Y. (2005). A comprenhensive survey of fitness approximation in evolutionary computation. *Soft Computing, 9*(1), 3–12.
Kampolis, I. C., & Giannakoglou, K. C. (2008). A multilevel approach to single and multiobjective aerodynamic optimization. *Computational Methods in Applied Mechanics Engineering, 197*, 2963–2975.
Kampolis, I. C., Trompoukis, X. S., Asouti, V. G., & Giannakoglou, K. C. (2010). CFD-based analysis and two-level aerodynamic optimization on graphics processing units. *Computational Methods in Applied Mechanics Engineering, 199*, 712–722.
Khurana, M., Winarto, H., & Sinha, A. (2009). Airfoil optimization by swarm algorithms with mutation and artificial neural networks. *American Institute of Aeronautics and Astronautics Conference*, 278–1289.
Kim, J. E., Rao, V. N., Koomullil, R. P., Ross, D. H., Soni, B. K., & Shih, A. M. (2009). Development of an efficient aerodynamic shape optimization framework. *Mathematics and Computers in Simulation, 79*, 2373–2384.
Liakopoulos, P., Kampolis, I. C., Giannakoglou, K. C., & enabled, Grid (2008). hierarchical distributed metamodel-assisted evolutionary algorithms for aerodynamic shape optimization. *Future Computer Systems, 24*, 701–708.
Lian, Y., Oyama, A., & Liou, M. S. (2010). Progress in design optimization using evolutionary algorithms for aerodynamic problems. *Progress in Aerospace Sciences, 46*, 199–223.
Li, P., Seebass, A. R., & Sobieczky, H. (1998). Manual aerodynamic optimization of an oblique flying wing. *American Institute of Aeronautics and Astronautics Conference*, 98–108.
Newman, J. C., Taylor, A. C., Barnwell, R. W., Newman, P. A., & Hou, G. J. III, (1999). Overview of sensitivity analysis and shape optimization for complex aerodynamic configuration. *Journal of Aircraft, 36*(1), 87–96.
Ortiz-García, E., Salcedo-Sanz, S., Pérez-Bellido, A., & Portilla-Figueras, J. A. (2009). Improving the training time of support vector regression algorithms through novel hyper-parameters search space reductions. *Neurocomputing, 72*, 3683–3691.
Praveen, C., & Duvigneau, R. (2009). Low cost PSO using metamodels and inexact pre-evaluation: Application to aerodynamic shape design. *Computational Methods in Applied Mechanics Engineering, 198*, 1087–1096.
Salcedo-Sanz, S., Ortiz-García, E. G., Pérez-Bellido, A. M., Portilla-Figueras, A., & Prieto, L. (2011). Short term wind speed prediction based on evolutionary support vector regression algorithms. *Expert Systems with Applications, 38*(4), 4052–4057.
Santos, M., De Mattos, B., & Girardi, R. (2008). Aerodynamic coefficient prediction of airfoils using neural networks. *American Institute of Aeronautics and Astronautics Conference*, 887–897.
Smola, A. J., Murata, N., Scholkopf, B., & Muller, K. (1998). Asymptotically optimal choice of ϵ-loss for support vector machines. In *Proc. of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing*.
Smola, A. J., & Schölkopf, B. (1998). A tutorial on support vector regression. *Statistics and Computing*.
Yao, X., Liu, Y., & Lin, G. (1999). Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation, 3*, 82–102.
Zhong-Hua, H., Zimmermann, R., & Görtz, S. (2010). A new cokriging method for variable-fidelity surrogate modeling of aerodynamic data. *American Institute of Aeronautics and Astronautics Conference*.