Blucher Mechanical Engineering Proceedings
May 2014, vol. 1 , num. 1
www.proceedings.blucher.com.br/evento/10wccm

10th World Congress on Computational Mechanics
8-13 July 2012 • São Paulo • Brazil

METAMODELING TECHNIQUES APPLIED TO AIRCRAFT NOISE PREDICTION

B. S. Aflalo[1], D. B. T. P. A. Ferrari[1]

[1]Department of Mechanical Engineering, Technological Institute of Aeronautics

Abstract. *Given the increasing impact of aircraft noise for certification and commercialization purposes, a great effort has been made to decrease the acoustic emissions of newer aircraft. A great number of airports around the world limit noise emissions during the landing and takeoff phases, which contributes to increase the importance of controlling acoustic emissions of high-lift devices.*

Typically, simulations on aeroacustics are extremely expensive, because they involve fluid dynamics computations in the transient regime. Not rarely, a single run of the computer code may require dozens of thousands of CPUs. Thus, direct optimization in aeroacoustics is often unfeasible for most of practical applications.

A popular approach that has been currently adopted to overcome such difficulties involves creating a cheap-to-compute, yet accurate, surrogate (or metamodel) for the computation-intensive function on which optimization is to be performed.

This work aims at using and analyzing the performance of various metamodeling techniques in slat noise simulations, carried out using a 2D discrete vortices method.

Three classes of metamodels, were tested: (i) first and second order polynomial regressions, (ii) artificial neural networks, and (iii) Gaussian stochastic processes (GaSP). For the neural networks model, various models were created in order to verify which combinations of numbers of neurons and layers yielded better results. For the GaSP models, different covariance functions and base functions were tested. All metamodels were constructed based on data generated according to a Latin hypercube design with 28 points in a four-dimensional input space consisting in angle of attack, slat deflection angle and the two dimensions of slat position.

The performance of each model was assessed with respect to the root mean square cross-validation prediction error. Results for the performed simulations indicated that the Gaussian stochastic process with the power exponential covariance function and a first order base function produced the best metamodel.

Keywords: *Metamodeling Techniques, GaSP, Aeroacoustics, high-lift noise.*

1. INTRODUCTION

The noise limits requirements for commercial aircrafts, both for certification and airport operations purposes, are becoming more and more restrictive. Despite the aircraft noise, specially the engine noise, has decreased considerably in the past decades (with the use of high by pass ratio, swept fan blades, specials acoustic treatments), the community noise annoyance is increasing, because of the rapid air traffic increase. Since 2006, commercial aircrafts shall meet the so called “Stage 4”, according to ICAO [1]. Figure 1 shows the trend of such requirement in the last years and the noise certificated of various known aircrafts. It is possible that in the next ten years, a new certification limit, called Stage 5, could come into effect.

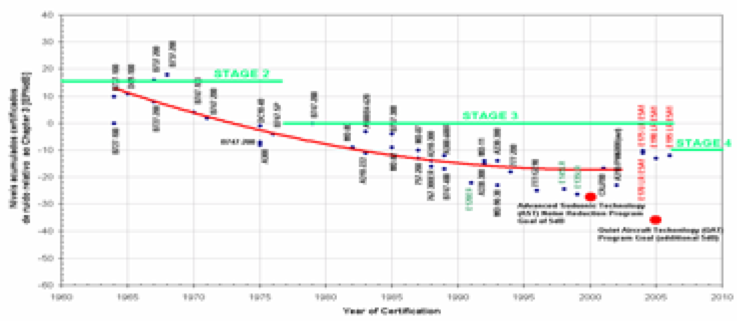


Figure 1 – Trend of certification limit, known as stages.

The noise components breakdown varies from aircraft to aircraft and during the takeoff and landing phases. Such breakdown can roughly be estimated as shown in Figure 2. It is worth noting the importance of the airframe noise (the noise produced by the flow around the aerodynamic surfaces – the green bar in the Figure 2) during the landing. In takeoff, the major noise contributions are from engines, both fan (strong directivity in the forward direction) and jet noise (strong directivity in the backward direction).

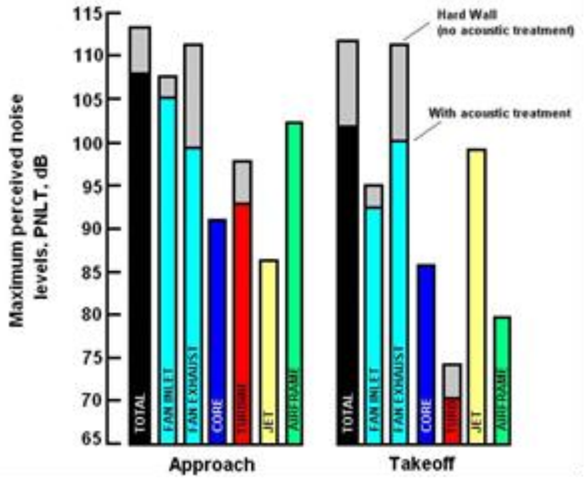


Figure 2: Aircraft Noise breakdown during takeoff and landing phases.

The airframe noise during the landing phase consists mainly of the noise produced by the devices deployed during this flight phase: the main and nose landing gears and the high-lift devices both in the wing leading edge (slats) and wing trailing edges (flaps). These high-lift devices are deployed to increase the aircraft maximum lift and, thus, reduce the speed at approach and the necessary landing field length.

Leading edge high lift devices have various noise generation mechanisms, including Kelvin-Helmholtz instabilities in the shear layer, vortex pairing, vortex collisions, vortex acceleration nearby the slat gap and alternated vortex shedding in slat trailing edge. These mechanisms are illustrated in the following figure [2].

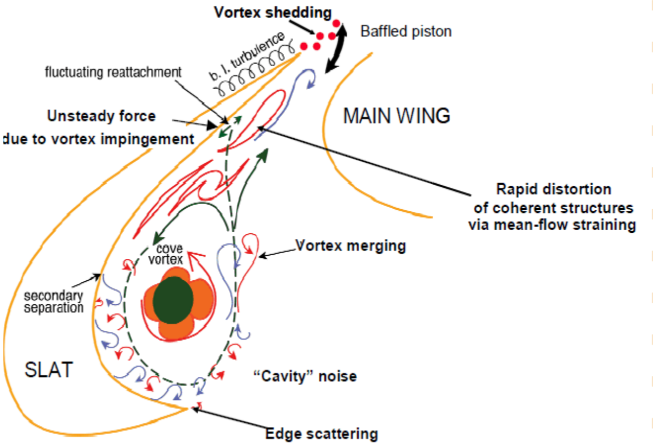


Figure 3: Slat Noise Generation Mechanisms [2].

Compute slat noise with all its geometric and topological complexities requires very complex transient computational aeroacoustics simulations (CAA) such as DES (detached eddy simulation) and LES (large eddy simulation). The computational effort to accomplish such simulations could be as higher as tens of thousands of CPU hours [3] for relatively simple, small scale cases. For real industrial applications, this kind of simulation, during the product development time-frame, could be unfeasible.

This work aims at testing different metamodeling techniques to model the acoustic power radiated by a 2D slat geometry, calculated using the discrete vortex method. Using the open-source R software, the techniques used were Polynomial regression using Least Squares, Stochastic Gaussian Process (GaSP) and Neural Networks. A total of 30 simulations were carried out, varying:

- Slat deflection angle (hereafter referred to as SlatDef);
- Airfoil angle of attack (hereafter referred to as AOA);
- x Offset of slat from main element, as shown in Figure 4 (hereafter referred to as Dx)
- y Offset of slat from main element, as shown in Figure 4 (hereafter referred to as Dy)

These parameters, inputs of the metamodels, are highlighted in green in Figure 4. The output, highlighted in blue, is the acoustic power integrated between emission angles 110º and 150º (in which the 0º is pointing toward –x).

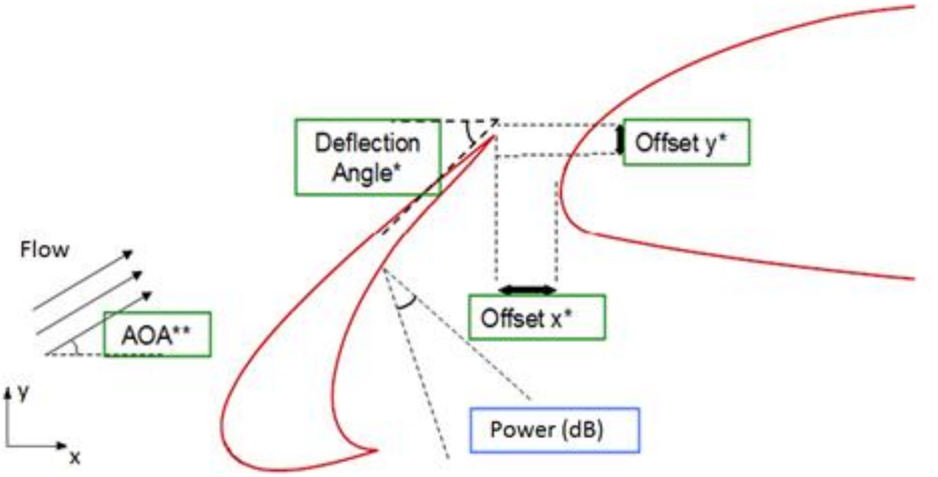


Figure 4: Metamodels inputs and outputs.

2. DISCRETE VORTEX METHOD

This methodology was used to provide, with relatively low computational cost, information about how the slat noise is affected by flow and geometry modification [4]. The code was implemented in Matlab and it assumes a two dimensional flow around the high-lift.

The method sheds, from each geometry trailing edge and the slat cusp, one vortex at each time step [4]. When a vortex shed occurs, its strength is determined by the Kutta condition and by the vortex conservation law for each one of the three elements (flap, slat and main element). The potential equation, written in complex form, is given by:

$$W = zUe^{-i\alpha} + \frac{1}{2\pi}\sum_{n=1}^{Ne}\int_{Sn}\left(\sigma - i\gamma_n\right)\ln\left(z - z_s\right)ds + \frac{1}{2\pi i}\sum_{k=1}^{Ny}\Gamma_k \ln\left(z - z_k\right) \qquad (1)$$

In which $z_k(t)$ and Γ_k are the position and the vortex strength in the flow, z e z_s are, respectively, a point inside the flow and a point on the high-lift surface, S_n is the surface of nth element, U is the freestream flow, α is the airfoil angle of attack, Ne is the number of elements (3, in this case), $\sigma(s,t)$ represents the source distribution on the elements surfaces and $\gamma_n(t)$ is the circulation around each of the elements.

The instantaneous speed at any point of the flow can be obtained by applying the gradient operator in the complex potential:

$$u(x,y,t) = \{u,v\} = \nabla\phi(x,y,t) \qquad (2)$$

The boundary conditions at profile surfaces can be obtained applying the non-porosity condition:

$$\vec{u}\cdot\vec{n} = \mathrm{Im}\left(e^{i\alpha} dW/dz\right) = 0 \qquad \text{over } S_n \qquad (3)$$

The vortex dynamics are computed assuming that each vortex velocity is equal to the regular part of the local instantaneous flow.

$$\frac{dz_l}{dt} = \lim_{z \to zl}\left(\frac{dW}{dz} - \frac{\Gamma_l}{2\pi i}\frac{1}{z - z_l}\right)^{*} \qquad (4)$$

In which the symbol * indicates conjugate complex. Once the near flow is computed, the noise is propagated using a frequency domain implementation of the solution of Lighthill equations proposed by Ffowcs Williams/Hawkings [5]. All the time to frequency domain transformation were carried out using a Welch periodogram with hanning windowing in each subsample. A total of 2000 time steps were carried out for each simulation shown.

This methodology was applied to slat noise computation in many works available in the literature [6]. As a two dimensional simulation and because the potential method do not contains all the physics involved in this complex problem, this method is not the most suitable for the determination of an absolute noise level, but rather for a sensitivity analysis. For this reason, in this work, the output variable, the integrated noise power, will be normalized in such a way that the maximum value will be 0 dB.

3. DOE AND SIMULATION

A total of 30 simulations were carried out for the slat noise computation using the discrete vortex method. The script, implemented in Matlab, takes 2,5 hours to run the 2000 timesteps of one case in a IntelCore 2Duo CPU 7500 @ 2.93GHz, 8GB RAM, using version v7.11-R2010B-X64 of software. The data was generated according to a Latin hypercube design with the criterion of maximize the minimum distance of the 30 points in a four-dimensional input space consisting in angle of attack, slat deflection angle and the two dimensions of slat position. Before further analysis, the data was verified [7] for:

- Input data errors
- Missing values
- Outliers
- Clustering

Two simulations presented non-physical results. In the following figure, one can see the typical vortex distribution around a slat, calculated with a 2d transient analysis (left) and one of the non-physical results obtained (right). After this check, only 28 valid points were used to assess the metamodels, that is, 7 points per input dimension.

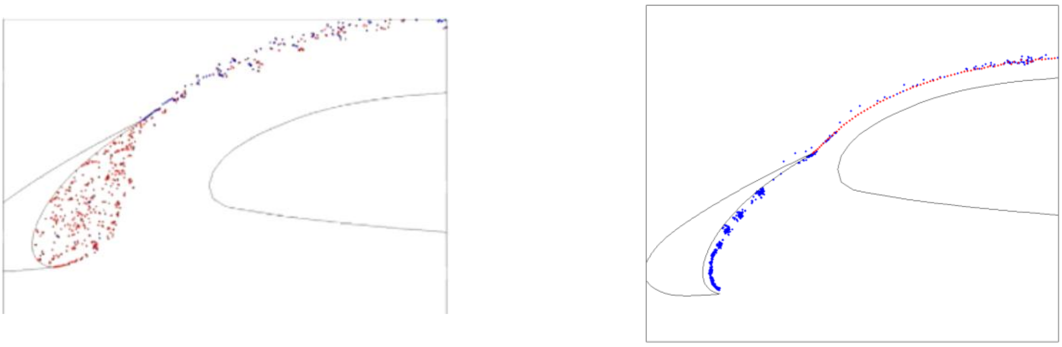


Figure 5: Comparison between a good (left) and a bad (right) simulation.

Follow bellow the Q-Q normal distribution and the histogram of the output variables, which indicate how close the noise power emission outputs are from the normal distribution.

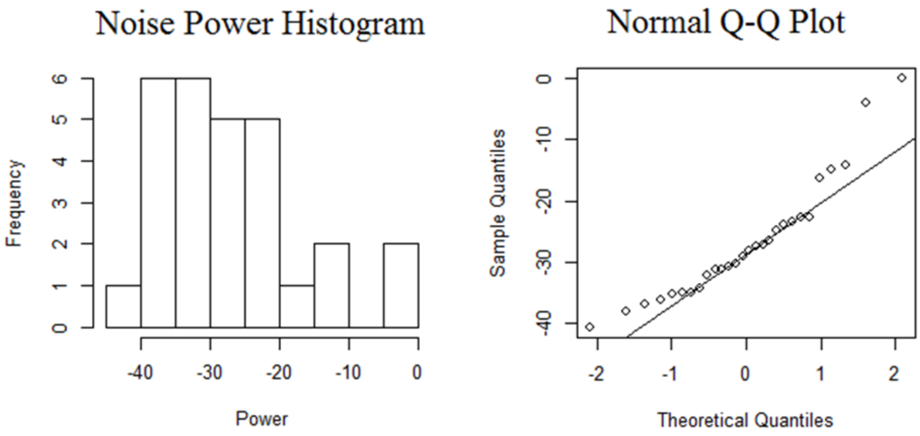


Figure 6: Histogram and Q-Q normal plot.

4. METAMODELING TECHNIQUES

4.1 Least Squares Polynomial Regression

This technique aims at adjusting a polynomial of order n to a data set in such a way that the quadratic error between the adjusted polynomial and the sample points are minimized [8]. In the least squares fitting, unknown variables are the polynomial coefficients. Defining the quadratic sum of residuals as:

$$S = \sum_{i=1}^{n} r_i^2 \tag{5}$$

In which the residual is defined as $r_i = y_i - f(x_i, \beta)$. For the linear model, the polynomial can be written as:

$$f(x_i,\beta)=\sum_{j=1}^{m}\beta_j\phi_j(x_i) \tag{6}$$

Taking the derivative of equation 5 and set to zero leads to the expression for the linear coefficients:

$$\hat{\beta}=(X^TX)^{-1}X^Ty,\text{ in which } X_{ij}=\frac{\partial f(x_i,\beta)}{\partial\beta_j}=\phi_j(x_i) \tag{7}$$

In most of application, including the present work, the number of unknown of the resulted linear system is lower than the number of equations. This means that not necessarily (and in most cases, it doesn't!) this technique will interpolate the sample points. For the metamodeling of deterministic process, this is a highly desirable characteristic, because we don't have any measurements or random errors associated with each point.

4.2 Artificial Neural Networks

The artificial neural networks technique is based on how the biological systems like brains process information [9]. The goal is to, given a set of inputs and outputs (the training set), train a highly connected elements, known as artificial neurons, to reproduce the outputs of these sample points. Then, this network can be used to estimate points not included in the training set and can deal with highly non-linear data.

A MLP neural network (multi-layer perceptron) consists in the input, the output and one or more hidden layers. The function connecting the input layer and the hidden layer is given by:

$$v_j=\sum_{i=1}^{s}w_{ji}x_i+w_{j0} \tag{8}$$

In which j is the hidden layer neuron index and i is the input layer neuron index. In the aforementioned equation, w represents the weight between the two layers and is the object to be optimized in the learning process. The objective function is a measured of residue between the value estimated by the network and the real value, in a validation set of data. In an analogous way, the function relating the hidden layer and the output layer is:

$$\hat{y}=\sum_{j=1}^{d}\beta_jb_j(v_j)+\beta_0 \tag{9}$$

In which d is the number of hidden layers, β_j is the weight which relates the hidden layer and the output layer and b_j is:

$$b_j(v_j)=\frac{1}{1+e^{-\lambda v_j}},\text{ ou } b_j(v_j)=\tanh(\lambda v_j) \tag{10}$$

v_j is the output of the jth neuron of hidden layer. For the learning process, it is recommended to standardize the input variables by subtracting the mean value and dividing by the standard deviation. This helps avoiding scale disparities between different input dimensions, harmful for the learning phase. The number of hidden layers and the number of neurons in each layer was subject of study in the present work. Too much neuron could result in an overfitting problem, while a small number of neurons could lead in an underfitting problem.

4.3. Gaussian Stochastic Process (GaSP)

Developed by the geologist D. G. Kridge to analyze mine data, the GaSP assumes a realization of a stochastic process Y(x). For a linear base function [10]:

$$Y(x) = \sum_{k=1}^{K} \beta_k f_k(x) + Z(x) \tag{11}$$

In which *$f_k(x)$* are known linear base function and β_k are the coefficients to be determined. The summation in the right hand side can be thought as a generalized least squares approximation and, hence, do not guarantee interpolation of the sample points. The term Z(x) is added to assure the sample point interpolation, according to a correlation function with a pre-defined shape. Between two sample points, the covariance between them can be written as:

$$\mathrm{cov} = [Z(x_1), Z(x_2)] = \sigma_Z^2 R(x_1, x_2) \tag{12}$$

In which $R(x_1, x_2)$, in the present work, is a function of the Euclidian distance between these two points. Two different shapes of the $R(x_1, x_2)$ functions were tested. The first, known as “power exponential", given by:

$$R(x_{1,} x_2) = \prod_{j=1}^{d} \exp(-\theta_j \left| x_{1j} - x_{2j} \right|^{p_j}) \tag{13}$$

and gaussian functions, particular cases of equation 13 when *$P_j=2$*. Only in 1989, Sacks [11] used this technique to model deterministic computer experiments and, since then, it is being used in many different applications with this purpose.

5. RESULTS

In the following subsections, results for each one of the metamodels assessed will be shown. In order to compare the performance of each one, the root mean square error (RMSE) calculated using the cross validation, or “leave-one-out” technique.

5.1. Least Squares Polynomial Regression

These analyzes were carried out using the function "lm" of software R. First and second order polynomial, completes and reduced were used to model the sample points. Reduced polynomials shall mean a polynomial without the terms with low relevance for the metamodel. These low relevance terms are withdrawn from model to increase the degree of freedom and the quality of the fitting. The following figure shows the residuals, the leverage graphic and the Cook distance for the complete first order polynomial.

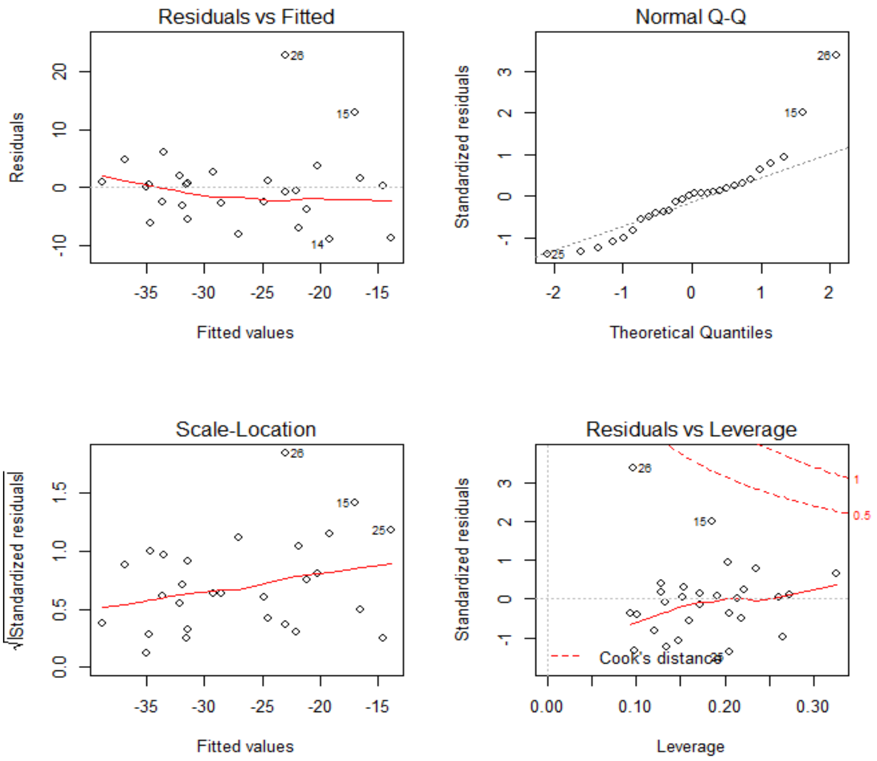


Figure 7: Residue, standardized residue, Leverage and Cook Distance for 1st order polynomial.

Points 15 and 26 are the ones with the higher standardized residuals, exceeding 2. The "Cook" distance [12] indicates the influence of an individual point for the regression. It also shows sample points to be more carefully analyzed and even regions to insert more sample points. The absolute residuals are as high as 10 dBs, too high to be used in an aircraft preliminary design phase. The cross validation RSME, for this model, is 6,98. An ANOVA (analysis of variance) of the first order polynomial resulted in the following table:

Table 1: ANOVA of the first order polynomial

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
AOA	1	847.77	847.77	16.5858	0.0004697
Dx	1	1.15	1.15	0.0224	0.8822527
Dy	1	281.67	281.67	5.5106	0.0278768
SlatDef	1	280.91	280.91	5.4957	0.0280669
Residuals	23	1175.63	51.11		

Note that variable Dx doesn't play an important role in the model, because the variance explained by it is much lower than the total variance. This variable was, therefore, withdrawn from the polynomial, leading to the "reduced 1st order polynomial". This new model has a RMSE of 6,83. The complete 2nd order polynomial residuals can be visualized in the following figure.

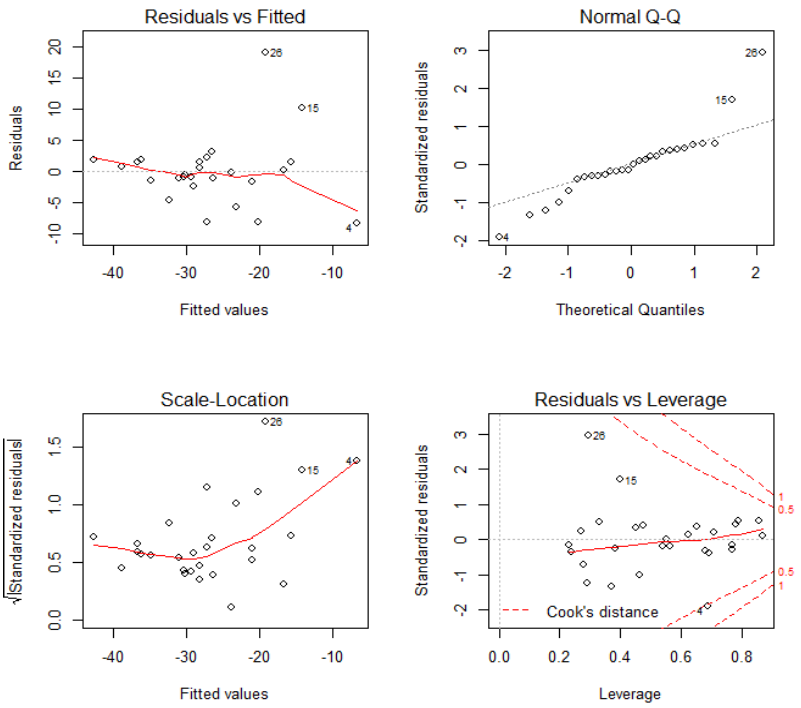


Figure 8: Residue, standardized residue, Leverage and Cook Distance for 2nd order polynomial.

Even with the second order polynomial, points 15 and 26 still present high standardized residuals. This metamodel better represents the sample points, because the cross validation RMSE was 5,26. An ANOVA of this surface shows that a number of terms don't play important roles, such as Dx, Dx^2, Dy^2, $SlatDef^2$, AOA * Dy, Dx * SlatDef, Dy * SlatDef. The reduced second order method had, thus, these terms removed and resulted in a RMSE of 5,91. The following table summarizes the results of the four least squares polynomial fitting used.

Table 2: RMSEcv of polynomial regression

Polinômio	RMSE (cross validation)
1st order complete	6,98
1st order reduced	6,83
2nd order complete	5,26
2nd order reduced	5,91

5.2. Artificial Neural Network

The neural network metamodel was carried out using package AMORE of R software [13]. With this package, it is possible to specify different activation functions, modify the learn rate, number of neurons in each layer, number of training iterations and others.

The input variables were standardized by using (x-xmean)/σx, in which σx is the standard deviation of input variable x. In general, for the cases analyzed, subtle differences were noticed by using the standardized variables or not.

Due to the random initialization of initial weights and to the presence of local minimum in the optimization process, each neural network run, keeping all the parameters constant, leads to different results. This is why each result presented in this section is, actually, the mean of ten runs with the same inputs and parameters. It is worth noting that the mean had been calculated with the final result, not with the network weights.

The activation function of hidden layer had a sigmoidal shape while, in the output, it was employed a linear function. Different ANNs was tested:

- For the one hidden layer networks: number of neurons varying from 1 to 30;
- For the two hidden layers networks: number of neurons varying from 1 to 25 in each layer.

Figures 9 and 10 show how the cross validation RMSE changes with the number of neurons for one and two hidden layers, respectively. For the ease of visualization, Figure 10 abscissa represents the index of the network created, in such a way that index = n1+25(n2-1), in which n1 and n2 are the number of neurons in first and second hidden layers, respectively.

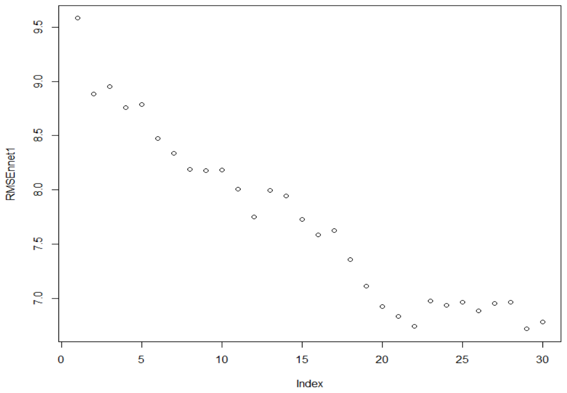


Figure 9: RMSEcv vs number of neurons – one hidden layer.

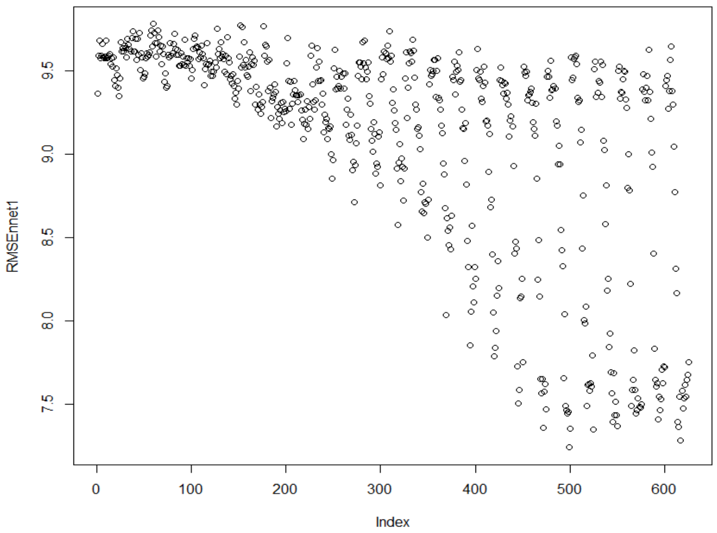


Figure 10: RMSEcv vs number of neurons – two hidden layer.

Among all network tested, the best result was obtained with 22 neurons in just one layer, leading to a RMSEcv = 6,75. The following graphic shows the simulated acoustic power against the cross validation prediction of this network.

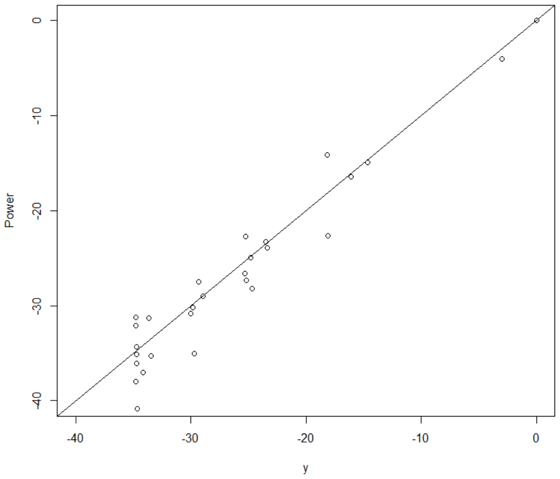


Figure 11: Exact values vs cross validation fitted results with an ANN with a 22 neurons.

5.3. Gaussian Stochastic Process (GaSP)

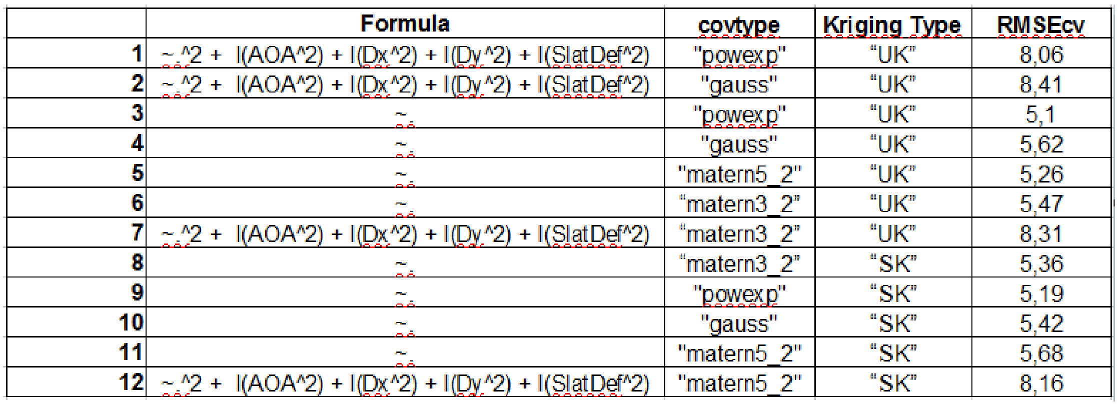
Different covariance functions, kriging types (universal and simple kriging) and base functions μ(x), available in Dicekriging R package, were tested. The covariance functions tested are [14]:

- Gaussian
- Matérn v = 5/2

- Matérn v = 3/2
- Power-Exponential

Universal Kriging is the more general type, because it uses, as base function, a linear model of type $\mu(x)=\sum_{k=1}^{K}\beta_k f_k(x)$, while the simple kriging uses a constant base function in the form $\mu(x)=K$. Table 3 shows the RMSE for the different GaSP settings.

Table 3: Dicekriging RMSEcv with different parameters



	Formula	covtype	Kriging Type	RMSEcv
1	~.^2 + I(AOA^2) + I(Dx^2) + I(Dy^2) + I(SlatDef^2)	"powexp"	“UK”	8,06
2	~.^2 + I(AOA^2) + I(Dx^2) + I(Dy^2) + I(SlatDef^2)	"gauss"	“UK”	8,41
3	~.	"powexp"	“UK”	5,1
4	~.	"gauss"	“UK”	5,62
5	~.	"matern5_2"	“UK”	5,26
6	~.	“matern3_2”	“UK”	5,47
7	~.^2 + I(AOA^2) + I(Dx^2) + I(Dy^2) + I(SlatDef^2)	“matern3_2”	“UK”	8,31
8	~.	“matern3_2”	“SK”	5,36
9	~.	"powexp"	“SK”	5,19
10	~.	"gauss"	“SK”	5,42
11	~.	"matern5_2"	“SK”	5,68
12	~.^2 + I(AOA^2) + I(Dx^2) + I(Dy^2) + I(SlatDef^2)	"matern5_2"	“SK”	8,16

Symbol “~” indicates a first order model and “~.^2 + I(AOA^2) + I(Dx^2) + I(Dy^2) + I(SlatDef^2) ” indicates a second order model for μ(x). The best result was obtained with a universal kriging, a “power exponential” function and a first order surface for μ(x). Figure 12 shows graphics “leaves-one-out”, standardized residuals and QQ-normal plot of the standardized residuals for this model. As in the case of the neural network computations, the final result was, actually, the mean value of 10 calculated RMSEcv, to reduce statistical variability. Both θ and p are calculated in the Dicekriging package. That is, not always the p equals 2, which leads to surfaces not infinitely differentiable.

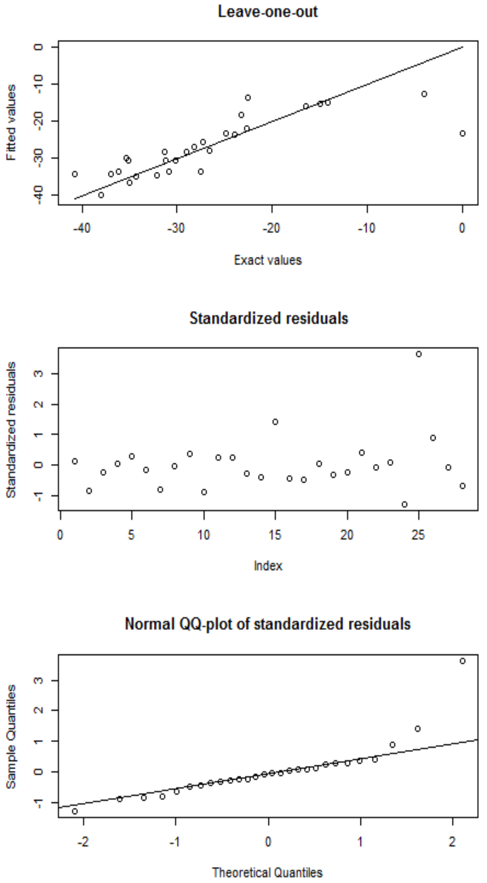


Figure 12: Dicekriging Results.

6. CONCLUSIONS AND NEXT STEPS

Least Squares Polynomial Regression, Artificial Neural Networks and Gaussian Stochastic Process were applied to a high-lift aeroacoustic simulation. For each one of the techniques, a study was carried out to understand the influence of each model parameter in the final result. For example, differences of 45% with the artificial neural network were obtained, just varying a small number of neurons. In the same way, significant differences (near 65% of RMSEcv) were obtained by changing the covariance and base functions in the GaSP model.

For the cases analyzed, the best result, a RMSEcv of 5,1, was achieved using the GaSP with a first order base function and "power exponential" correlation. The following table summarizes the best RMSEcv for each metamodel:

Table 4: RMSEcv calculated with the different metamodels

Metamodeling Techniques	**RMSEcv**
2nd order polynomial regression	5,3
Neural Network with 22 neurons in a single hidden layer, sigmoidal activation function	6,8
GaSP with first order base function and a "power exponential" correlation	5,1

Indeed, residuals between 5 and 7 dBs are considered high for a noise estimative. An error of such magnitude could drastically impact decisions in a preliminary aircraft design phase. Most likely, if more points were added to the sample points used in this work, better results could be obtained. It is believed that, for a four-dimensional input data, a minimum of 40 points would lead to a better estimative. As aforementioned, due to the high computational costs associated with aeroacoustics simulations, a metamodel that gives good results even with a relatively small number of input variables is a must. Further studies can be carried out towards the improvement of current results, such as:

- Models aerodynamic parameters together with the aeroacoustics results, in order to substantiate design decisions;
- Analyze how RMSEcv changes with more and less input points;
- Assess the metamodels performance in an optimization process.

Acknowledgements

The authors thank "Aeronave Silenciosa" project, a partnership between Embraer, FAPESP (Fundação de Apoio à Pesquisa do Estado de São Paulo) and Universities USP, ITA, UnB, UFSC and University of Uberlândia. Special thanks to André Bonatto, for providing the discrete vortex method code.

7. REFERENCES

[1] ICAO - International Civil Aviation Organization. Disponível em: http://www.icao.int/environmental-protection/Pages/default.aspx. Acessado em: 25/01/2012.

[2] Choudhari, M. M., Khorrami, M. R., Effect of Three-Dimensional Shear-Layer Structures on Slat Cove Unsteadiness, AIAA Journal Vol 45, No.9, September 2007.

[3] Aflalo, B. S., Simoes, L. G. C., Silva, R. G., Medeiros, M. A. F., “Comparative Analysis of Turbulence Models for Slat Noise Source Calculations Employing Unstructured Meshes”, 16th AIAA/CEAS Aeroacoustics Conference, Stocolmo, Suécia. AIAA-2010-3838.

[4] Guo, Y., "A Discrete Vortex Model for Slat Noise Prediction", 7th AIAA/CEAS Aeroacoustics Conference, Maastricht, Netherlands. AIAA-2001-2157.

[5] Ffowcs Williams, J. E., and Hawkings, D. L., "Sound Generated by Turbulence and Surfaces in Arbitrary Motion", Philosophical Transactions of the Royal Society, Vol. A264, 1969, pp. 321-342.

[6] Rackl, R. G., Miller, G., Guo, Y., Yamamoto, K., "Airframe Noise Studies - Review and Future Direction".NASA/CR-2005-213767.

[7] Ferrari, D. & Head, T., Regression in R Part I :Simple Linear Regression, UCLA Department of Statistics - Statistical Consulting Center, 2010.

[8] Least Squares. Disponível em: http://en.wikipedia.org/wiki/Least_squares. Acessado em: 1º de dezembro de2011.

[9] Stergiou, C.; Siganos, D., Neural Networks. Disponível em: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#. What is a Neural Network. Acessado em: 07 de setembro de 2010.

[10] Ferrari, D., B., T., P., A., "Multi-fidelity Data Fusion in Statistical Metamodeling of Aero-dynamic Functions". PhD thesis in Statistics, University of California, Los Angeles, 2010.

[11] Sacks, J., Schiller, S. B. and Welch, W. J. (1989), “Designs for computer experiments”, Technometrics 31, 41–47.

[12] Cook's Distance, Wikipedia, the Free Encyclopedia. Acessado em 2 de dezembro de 2011, Disponível em http://en.wikipedia.org/wiki/Cook's_distance.

[13] The AMORE package: A MORE flexible neural network package. Disponível em: http://rwiki.sciviews.org/doku.php?id=packages:cran:amore. Acessado em: 12/12/2011.

[14] Roustant, O., Ginsbourger, D., Deville, Y., "DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization". HAL-00495766, versão 1 - 28 Jun 2010.