BALANCING OF FLEXIBLE ROTORS WITHOUT TRIAL WEIGHTS BY USING OPTIMIZATION TECHNIQUES

Manuel Villafañe Saldarriaga
Universidade Federal de Uberlândia
Faculdade de Engenharia Mecânica
Campus Santa Mônica – Bloco 1M
Uberlândia – MG – 38400-902
mavisa@mecanica.ufu.br

Valder Steffen Junior
Universidade Federal de Uberlândia
Faculdade de Engenharia Mecânica
Campus Santa Mônica – Bloco 1M
Uberlândia – MG – 38400-902
vsteffen@mecanica.ufu.br

The present work presents two flexible-rotor balancing procedures, which do not use trial weights. The goal of this approach is to help in situations found in the industrial context, in which trial weights based techniques cannot be applied. A few reasons can be mentioned: the time consumed to stop the machine to install the weights and start-up again to accelerate the rotor to the balancing rototation for several times is prohibitive; technical reasons may make difficult the work of installing the trial weights. The actual balancing technique is based on pseudo-random optimization methods. In this research work two methods were explored, namely, the genetic algorithms and artificial neural networks. The basic idea is to obtain the flexible rotor unbalance response, which is then mimicked by using a FEM model in which the unbalance masses and their corresponding angular positions are the design variables in the optimization run. This way, an inverse problem is solved and the masses obtained are installed in previously chosen balancing planes at inverted angular positions. Numerical simulations show the efficiency and limitations of the methodology developed.

rotodynamics, balancing, flexible rotor, inverse problems, optimization, neural networks, genetic algorithms

1. Introduction

Normally, the problem of balancing flexible rotors has been resolved using the influence coefficients method and the classic modal method, also it is common to use hybrid method based on a combination of these two techniques (Foiles et al, 1998).

The influence coefficients balancing method uses known trial weights to determine experimentally the response sensitivity of a rotor-bearing system with respect to the trial weights and subsequently calculates a set of discrete correction masses which will minimize whirl responses. In conventional procedures a trial mass is first applied to one of the balancing planes and the rotor responses are measured. This process is repeated for all the other balancing planes. Then an influence coefficient matrix is obtained from these data. Using this influence coefficient matrix is possible obtain a set of correction weights which will minimize the rotor vibrations (Lacerda, 1990).

The classic modal method, initially proposed by Bishop et al (1959) requires information about the flexural mode shapes. By means of the prior knowledge of the mode shapes and by measuring the vibration response close to the critical speeds it is possible to identify the generalized modal unbalances. The generalized unbalance corresponding to a given mode can be corrected by a set of discrete balancing moments. Then, it is possible to determine a set of orthogonal correction weights that are to be installed in the correction planes for balancing the rotor at prescribed critical speeds (Gnielka, 1983).

Xu et al (2001) proposed a rotor balancing method by using optimization techniques, which does not need trial weights. In the present work a FEM rotor model is used and two different optimization procedures are implemented, namely the Genetic Algorithms and Artificial Neural Networks. The goal is to determine a set of unbalanced weights, which installed in the mathematical model and reproduce an unbalance response close to the experimental one.

The procedure proposed in the present paper allows to reduce some disadvantages founds in the classical methods. For example, it is possible to balance the rotor even in the presence of large modal damping and large modal density cases, in which the modal balancing method fails (Vaqueiro,1989). Others disadvantages of the modal method such as those that arise in rotors with hydrodynamic journal bearings also can be minimized. In the case of the influence coefficient method , test weights are needed and the procedure impose many starts and stops which are extremely time-consuming. The runs with trial weights can therefore easily take up half, or more of the time involved in the whole balancing procedure. The method reported in this paper does not use trial weights, thus, it is possible a significant reduction in the time consumed in the balancing procedure.

In the remainder, a review of the rotor FEM model is presented, the concepts about Genetic Algorithms and Neural Networks are discussed together with the optimization procedures applied to the problem of rotor balancing. Then, characteristics of he experimental test rig used in the validation of he procedure are shown. Finally, the simulation results are compared with the experimental ones.

2. FEM Model

The mathematical model used to calculate the unbalance forces, natural frequencies and vibration mode shapes is based on the finite element method and the Lagrange equations. The discrete rotor model is represented by symmetric rigid discs elements, symmetric Timoshenko beam elements, non-symmetric coupling elements, and non-symmetric viscous damped bearings. The complete rotor model is represented by the following matrix differential equation:

$$[M]\cdot\ddot{\delta}+[C(\Omega)]\cdot\dot{\delta}+[K]\cdot\delta=[F(t)] \tag{1}$$

Natural frequencies and mode shapes are obtained from equation (1) for the homogeneous case, through the calculation of the system eigenvalues and eigenvectors.

The unbalance form is obtained from the kinetic energy expression of the unbalance mass, as follows:

$$T_u=\frac{1}{2}\cdot V^T\cdot m_u\cdot V \tag{2}$$

where:
m_u : unbalance mass
V : velocity vector of the unbalance mass

By using the Lagrange equations it is possible to obtain the mathematical model of the rotor system affected only by unbalance forces:

$$[M]\cdot\ddot{\delta}+[C(\Omega)]\cdot\dot{\delta}+[K]\cdot\delta=F_1\cdot\cos(\Omega\cdot t)+F_1\cdot\sin(\Omega\cdot t) \tag{3}$$

where F_1 and F_2 depends on the unbalance eccentricity, and the solutions are given by:

$$\delta=\Delta_1\cdot\cos(\Omega\cdot t)+\Delta_2\cdot\sin(\Omega\cdot t) \tag{4}$$

The modal base of the associated non-gyroscopic system is used to reduce the number of degrees-of-freedom in order to calculate the system eigenvalues and eigenvectors, and to obtain the unbalance response (Steffen and Lepore, 1983).

3. Optimization Strategies

The classical non-linear constrained optimization problem can be written mathematically as (Vanderplatts, 1983):

Minimize:

$F(X)$ Objective Function (5)

Subject to:

$g_j(X)\leq 0 \quad (j=1,M)$ Inequality Constraints (6)

$h_k(X)\leq 0 \quad (K=1,L)$ Equality Constraints (7)

$X_i^l\leq X_i\leq X_i^u \quad (i=1,N)$ Side Constraints (8)

In general the objective function (5), as well as the constraints functions (6) and (7), are non-linear implicit functions with respect to the design variables, as balancing can be understood as an inverse problem, the classical optimization methods could have difficulties in these cases due to local minimum (Assis, 1998). Most classical method do not have the global perspective and often get converged to locally optimal solutions. Another difficulty is their inability to be used in parallel computing environment efficiently.

Over the years, a number of search and optimization algorithms, which are drastically different in principle from de classical methods, are getting increasingly more attention. These method mimic a particular phenomenon to solve search and optimization problems. In this work, two of these method are used for to determine the correction mass in order to balance an unbalanced rotor, these are: Genetic Algorithms and Neural Network.

3.1 Neural Networks

Studies on neural networks have been motivated to imitate the way that the brain operates. A network is described in terms of the individual neurons, the network connectivity, the weights associated with various interconnections between neurons, and the activation function for each neuron (Haykin, 1998). The network maps an input vector from one space to another. The mapping is not specified, but is learned. The network is presented with a given set of inputs and their associated outputs. The learning process is used to determine proper interconnection weights and the network is trained to make proper associations between the inputs and their corresponding outputs. Once trained, the network provides rapid mapping of a given input into the desired output quantities. This, in turn, can be used to enhance the efficiency of the design process.

Consider a single neuron. This neuron receives a set of n inputs, x_i , i=1, 2 ,... n, from its neighboring neurons and a bias whose value is equal to one. Each of the inputs has a weight (gain) w_{ji} connecting between the ith and the jth units. The weighted sum of the inputs determines the activity of a neuron, and is given as:

$$net_j = \sum_{j=1}^{n} w_{ji} \cdot x_i \qquad (9)$$

A simple function is now used to provide a mapping from the n-dimensional space of the inputs into a one-dimensional space which comprises of an output value a neuron sends to its neighbors. The output of a neuron is a function of its activity:

$$y = f(net) \qquad (10)$$

Many types of neural networks have been proposed by changing the network topology, node characteristics, and learning procedures. In this study, we use a multi-layer feed-forward neural network topology with one hidden-layer as shown in Figure 5.

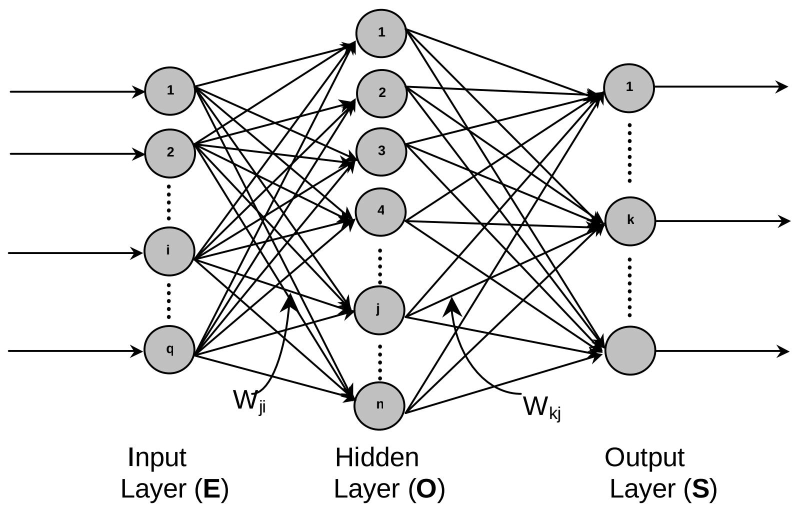

Figure 1 - Multi-layer feed-forward neural network

The training method used in this works is based to the Levenberg-Marquardt algorithm, this method is a modification of the Newton method, it uses second order terms to calculate the Hessian matrix (H) , and guarantees a convergence faster than the methods based on the descending gradient. However, the necessity to calculate the Hessian matrix can be an unsurmountable difficulty for some applications of Neural Networks (Masters, 1993).

To prevent these inconveniences methods have been developed that create an array which aproximates the H matrix. There methods are known as Quasi-Newton methods (Vanderplaats, 1993). In this case, the Levenverg-Marquardt method uses an approximation of the Hessian matrix based on the product of Jacobeans. Also, the Levenberg-Marquardt method minimizes the errors presented for ill-conditioning of the approximate Hessian matrix. The modified equation of the Newton method used in the Levenberg-Marquardt algorithm is:

$$X_{t+1} = X_t - \left[H(X_t) + \mu_t \cdot I\right]^{-1} \cdot \nabla F(X_t) \quad (11)$$

with:
X_t = vector of variables in the *t*-th iteration
$H(X_t)$ = Hessian matrix for X_t
$F(X_t)$ = Function to minimize for X_t
$\nabla F(X_t)$ = Gradient function of $F(X_t)$

In equation (11) I is the identity matrix. The value μ_t determines the trend of the algorithm, then, if μ_t is zero, equation (11) is reduced to the Newton method, but if μ_t is large, the value of $H(X_t)$ will be worthless with regard to $\mu_t \cdot I$. In this case the value: $-\left[H(X_t) + \mu_t \cdot I\right]^{-1} \cdot \nabla F(X_t)$ represents a small advance in the opposite direction of the gradient and the algorithm will have a behavior similar to the descending gradient method.

3.2 Genetic Algorithms

Genetic Algorithms constitute an iterative optimization procedure. Instead of working with a single solution in each iteration, a genetic algorithm works with a number of solutions (collectively know as a population). A flowchart of the basic principle of a simple Genetic Algorithm is shown in figure 2. In the absence of any knowledge of the problem domain, a Genetic Algorithm begins its search form a random population of solutions. As shown in the figure, a solution in Genetic Algorithms is presented by using a string vector of fixed length (it depends on the coding representation chosen). In each iteration, if the termination criterion is not satisfied, three different genetic operators (reproduction, crossover, and mutation) are applied to update the population strings. One iteration is called generation in the parlance of Genetic Algorithms. The representation of a solution in artificial Genetic Algorithms is similar to a natural chromosome and Genetic Algorithms operators try to mimic natural genetics. In the following subsection a brief description of the Genetic Algorithms is presented:

- *Reproduction*: is usually the first operator applied on the population. Reproduction selects good strings from a population and form a mating pool. There exists a number of reproduction operators in Genetic Algorithm literature, but the essential idea is that above-average strings are picked up form the current population, duplicates of them are inserted in the mating pool and these strings continue to the next generation. The commonly used reproduction operator is the proportionate selector operator, in which a string in the current population is selected with a probability proportional to the string's fitness.
- *Crossover*: is applied next to the springs of the mating pool. Similarly to the previous operator, there exists a number of crossover operator in the literature, but the main idea is that two springs are picked up from the mating pool and mixed, to produce two offsprings that keep some genetic characteristics from both ancestors.
- *Mutation*: This operator alter a small percentage of the strings. In the case of Binary Genetics Algorithms, it alters a small percentage of bits in the strings. It increases the algorithm's freedom to search outside the current region of parameters space. It also tends to distract the algorithm from converging to local minimum.

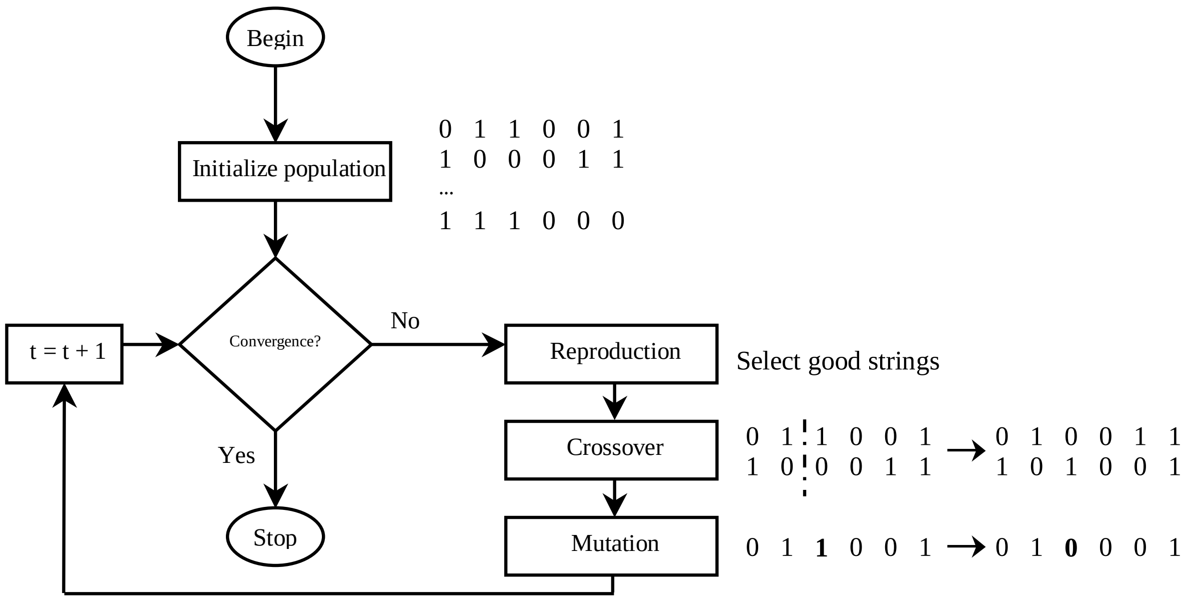

Figure 2 – Binary Genetic Algorithm Scheme

4. Identification of balancing rotor weights and their corresponding angular positions

In this research work the goal was to use optimization techniques to identify the unbalance condition of a flexible rotor. The method developed consists of using a rotor inverse model that permits the calculation of the rotor unbalance from the rotor displacements in he measure planes. For this purpose, two optimization procedures were developed, and are presented in the following subsections.

4.1 Using Neural Networks

In this case it is necessary a set of standard input for the neural network training. These inputs are obtained from the FEM model by introducing a set of random unbalance weights and collecting the vibration responses at the measure plane positions as calculated by the model. The process is shown in figure 3. The following steps are established:

- The experimental vibration response is measured and normalized.
- The FEM model is created.
- The input-output training sets are collected by using the FEM model and then are normalized. The normalization is necessary to use the neural networks. Besides the Neural Networks work better when the output values are in the polynomial form (the original form of the output of the net is Polar). It is necessary to transform this values in Cartesian coordinates.
  Thus, in the case of the displacements (input of the net) are normalized using a linear relation according to equation (16):

$$N(x,y) = 2 \cdot \frac{(R(x,y) - \min)}{(\max - \min)} - 1 \tag{12}$$

where:
N(x,y) = Normalized parameter corresponding to the *x*-th individual at the *y*-th measure plane.
R(x,y) = Parameter to be normalized corresponding to the *x*-th individual at *y* measure plane.
min = minimum value of the displacements.
max =maximum value of the displacements.

The output values of the net (weights and angles) are normalized by changing from replaced Polar to Cartesian coordinates, according to figure 3:

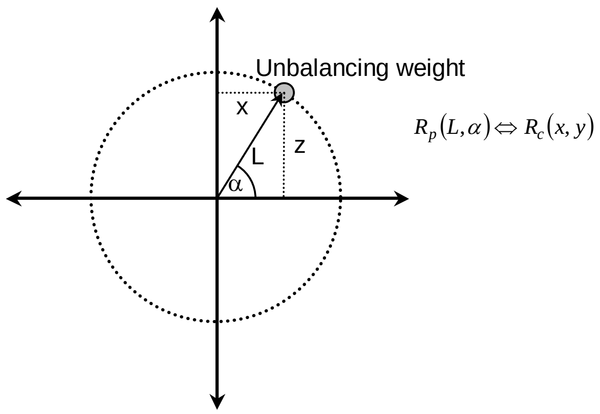

Figure 3 – Coordinate transform

where L is the unbalance moment associated to the concentrated unbalance weight and its corresponding eccentricity, and α is the phase angle.

The main steps to follow in the present case are depicted below:

- The net architecture is chosen by considering the different variables involved, such as the number of correction planes, measure planes and balancing speeds.
- The net is trained.
- The net is validated by using input-output training sets not used before in the training process.
- The experimental vibration response is introduced in the trained Neural Network to obtain a set of equivalent unbalance weights and angular positions.

- The set of balancing weights is determined, considering that the balancing weights are located at an angular position which is 180 degrees out of phase with respect to the set of determined equivalent unbalancing weights.
- The balancing weights are attached to the correction planes of the rotor
- The residual unbalance is checked

4.2 Using Genetic Algorithms

The basic idea of the procedure using Genetic Algorithms is similar to the one developed for Neural Networks. The procedure is presented as follows:

- The FEM model is created and fitted to the experimental data.
- The experimental response of the rotor is obtained for the balancing speeds.
- The objective function is defined as the difference between the displacements calculated in the measure planes of the model and those obtained from the test rig. That difference is associated with the strain energy (Lalanne and Ferraris, 1998), and the formulation of the objective function is:

$$F_{ob}(I_e) = \sum_{i=1}^{V} \sum_{j=1}^{2n} \left(F_{model}[I_e(i,j)]^2 - F_{exp}[i,j]^2 \right) \tag{13}$$

where:

$$F_{model}(I_e) = \begin{bmatrix} D_{11x} & D_{11z} & D_{21x} & D_{21z} & \cdots & D_{n1x} & D_{n1z} \\ D_{12x} & D_{12z} & D_{22x} & D_{22z} & & & \\ \vdots & & & & & & \\ D_{1Vx} & D_{1Vz} & & \cdots & & D_{nVx} & D_{nVz} \end{bmatrix} \tag{14}$$

and,

$$F_{exp} = \begin{bmatrix} R_{11x} & R_{11z} & R_{21x} & R_{21z} & \cdots & R_{n1x} & R_{n1z} \\ R_{12x} & R_{12z} & R_{22x} & R_{22z} & & & \\ \vdots & & & & & & \\ R_{1Vx} & R_{1Vz} & & \cdots & & R_{nVx} & R_{nVz} \end{bmatrix} \tag{15}$$

In equation (13), F_{ob} is the objective function, F_{model} is the rotor model response and F_{exp} are the vibrations measured in the rotor test rig. In equation (15), D_{jid} are the displacements calculated by the FEM model, in the *i*-th balancing speed at the *j*-th measure plane, and *d* corresponds to the acquisition direction. R_{jid} corresponds to the displacements experimentally acquired under the same conditions. I_e is the vector of design variables that is related to the *e*-th individual.

- The initial population is created, such a manner that each individual is a vector with $2\text{x}h$ elements, with h is the number of correction planes, thus:

$$I_e = [m_1, \alpha_1, m_2, \alpha_2, \cdots, m_h, \alpha_h]^T \tag{16}$$

where m_i are the unbalanced moments and α_i are their corresponding phase angles.

- The Genetic algorithm is executed until the stop criterion is satisfied.
- The set of unbalancing weights are determined using Genetic Algorithms. The balancing weights are located at an angular position which is 180 degrees out of phase with respect to the set of determined equivalent unbalancing weights.
- The balancing weights are attached to the correction planes of the rotor.
- The residual unbalance is checked

5. Numerical Simulations

For testing the balancing procedure presented in the previous section, both methods were simulated by using a rotor model shown in figure 4. The geometrical and physical proprieties of the elements used in the rotor discretization are presented in the tables 1,2, and 3. The material for shaft and disc elements is considered to be the steel ($E=2.067\text{x}10^{11}\text{N/m}^2$ and $\rho = 7800\ \text{kg/m}^3$).

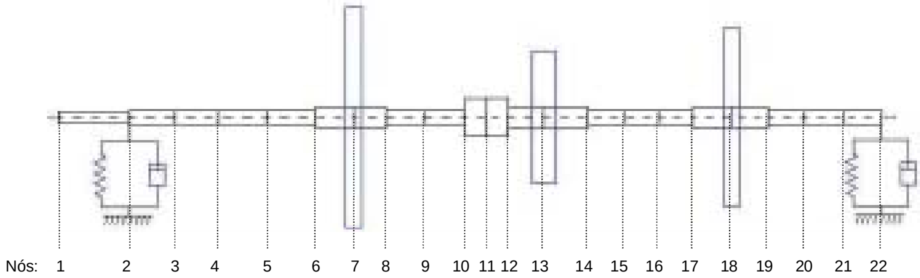

Figure 4 – Rotor Model

Table 1 - Shaft Elements

Number	Pos		Lenght [m]	Diameter [m]
1	1	2	0.048	0.004
2	2	3	0.03	0.005
3	3	4	0.029	0.005
4	4	5	0.033	0.005
5	5	6	0.032	0.005
6	6	7	0.026	0.007
7	7	8	0.022	0.007
8	8	9	0.025	0.005
9	9	10	0.027	0.005
10	10	11	0.015	0.0125
11	11	12	0.014	0.0125
12	12	13	0.024	0.007
13	13	14	0.029	0.007
14	14	15	0.025	0.005
15	15	16	0.023	0.005
16	16	17	0.023	0.005
17	17	18	0.026	0.007
18	18	19	0.024	0.007
19	19	20	0.025	0.005
20	20	21	0.026	0.005
21	21	22	0.026	0.005

Table 2 – Disc Elements

Number	Pos	Thickness [m]	Diameter [m]
1	7	0.0112	0.075
2	13	0.0157	0.045
3	18	0.0107	0.06

Table 3 – Bearing Elements

Number	pos	KX [N/m]	KZ [N/m]	CX [N.s/m]	CZ [N.s/m]
1	2	18750.0	10750.0	10.5	12.0
2	22	14770.0	24770.0	10.5	12.0

The critical speeds are shown in table 4

Table 4 – Critical Speeds

Modes	1	2	3	4	5	6
Critical Speeds [rpm]	639.54	696.05	1120.50	1214.20	2704.30	3029.30

5.1 Simulation using Neural Networks

The procedure using Neural Networks was applied to the rotor shown by figure 4 under two conditions: the first condition used two balancing planes and three measure planes; in the second three balancing planes and three measure planes were used. In both cases only one balancing speed (700 rpm) was considered. However, more balancing speeds could be used with the disadvantage of increasing the computational cost. The parameters used in the Neural Network are shown in table 6.

Table 6 – Neural Networks parameters

Layers	3
Training Algorithm	Levenverg-Marquart
Stop Criterion	RMS (<0.001)
Layer architecture	(6), (8), (6) for 3 balancing planes (6), (8), (4) for 2 balancing planes
Number of training sets	4000 for 3 balancing planes 2666 for 2 balancing planes

The first simulation required 50 epochs to achieve the stop criterion, while the second simulation used 120 epochs. The results between the weights attached in the rotor and the weights obtained using Neural Networks are shown in table 7.

Table 7 – Unbalance weights determined by using Neural Networks

Pos		Initial Unbalance	Identified unbalance (three balancing planes)	Identified unbalance (two balancing planes)
7	Mass [kg]	0.003	0.0034	0.0074
	Angle	150.00°	142.22°	127.58°
13	Mass [kg]	0.006	0.0056	0.0086
	Angle	30.00°	32.74°	334.06°
18	Mass [kg]	0.005	0.0048	----
	Angle	290.00°	292.91°	----
Amplitude Reduction			79.16%	89.37%

The unbalance response obtained at the disc number 2 is shown in the figure 5 for the different situations.

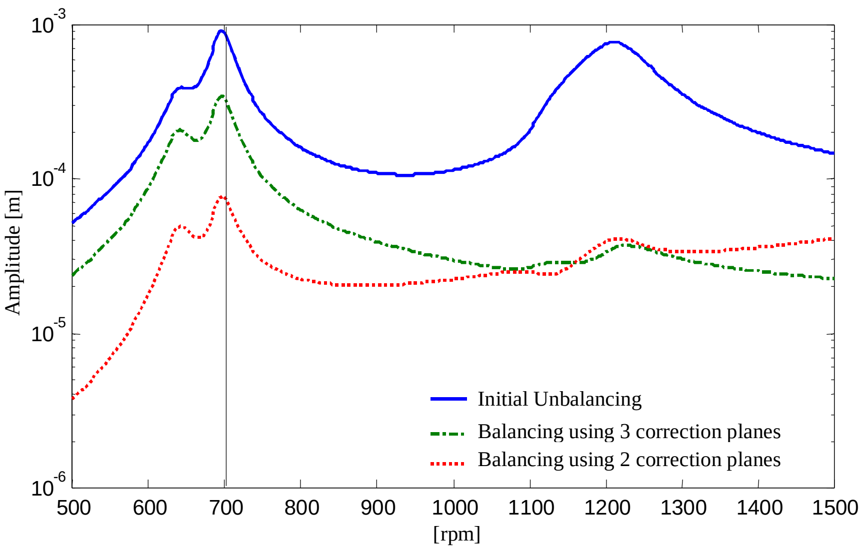

Figure 5 – Unbalance Response (before and after balancing)

5.2 Simulation using Genetic Algorithms

The same simulation procedure was used in the case of Genetic Algorithms, for which the parameter are shown in table 8.

Table 8 – Genetic Algorithm parameters

Coding Type	Real-Valued representation	
Stop Criterion	Number of Generations	50 Generations
Selection Function	Normalized geometric Selection	
Genetic Operators	Simple crossover	Crossover probability=0.6
	Uniform mutation	Mutation probability = 0.05
Initial Population	500	

ISO 11342 (1998), recommends for the case of balancing flexible rotors in bands enclosing *n* critical speeds, that at least *n* or, being possible, *(n+2)* balancing planes have to be used. Following these indications in the first case three balancing planes were used for the situation in which the balancing speeds are found in the band between 500 and 1200 rpm (enclosing three critical speeds). The vibration response was calculated at each 100 rpm interval. In the second the balancing speeds are in the range between 500 and 1000 rpm (enclosing two critical speeds). The result are shown in table 9.

Table 7 – Unbalance weights determined by using Genetic algorithms

Pos		Initial Unbalance	Identified unbalance (three balancing planes)	Identified unbalance (two balancing planes)
7	Mass [kg]	0.003	0.00311	0.0024
	Angle	150.00°	153.04°	70.29°
13	Mass [kg]	0.006	0.0068	0.0036
	Angle	30.00°	29.07°	286.50°
18	Mass [kg]	0.005	0.005	---
	Angle	290.00°	285.67°	---
Amplitude Reduction			95.20%	45.50%

The unbalance response obtained at disc number 2 is shown in the figure 6 for different situations.

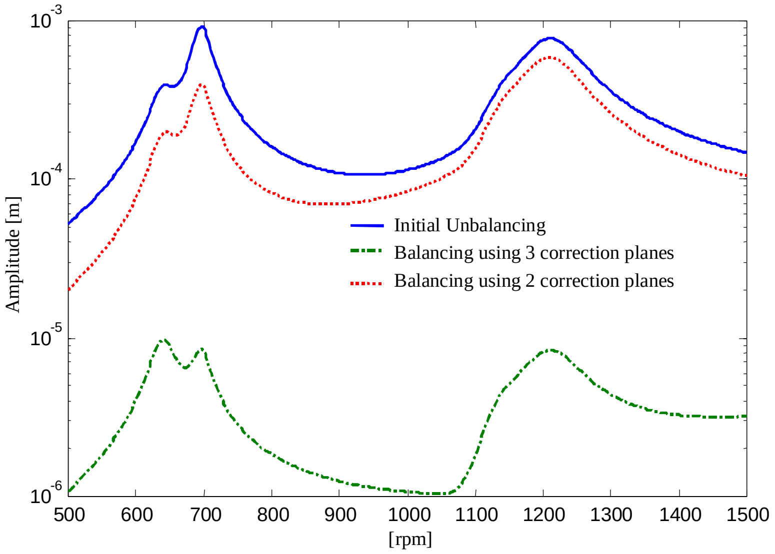

Figure 6 – Unbalance Response (before and after balancing)

6. Conclusions

The implemented procedures demonstrated efficiency for the two optimization techniques used (Artificial Neural Network and Genetic Algorithms). In both cases a good reduction in the level of vibration in the measure planes was obtained. The limitation in the case of the Neural Networks is represented by the difficulty of working with several balancing speeds simultaneously. In this way as more balancing speeds are considered, it is necessary to increase the complexity of the net and, consequently the computational time required for the training process together with the additional memory space are increased. In the cases studied in the present work only one speed of rotation was considered without meanwhile compromising the generality of the method. When more balancing speeds are considered, the training algorithm of Levenberg-Marquardt is no longer appropriated, because depending on the number of weights considered it can require a too expensive computational cost. It si worth to comment that selection of the data sets to train the network was the most time-consuming procedure. In uor case the training process itself was straightforward.

An advantage of the Neural Networks with respect to the Genetic Algorithms is the fact that the correction masses required and their respective angular positions are calculated automatically after introducing the vibrations measured in the rotor. The process of generating data and the training of the net takes a similar amount of time as for the Genetic Algorithms to determine the solution of the problem. Genetic algorithms, differently from the Artificial Neural Networks, offers the possibility of working in a band containing several speeds without adding a prohibitive computational cost. Besides, it allows to correct unbalancing for several critical speeds simultaneously. Then, it can be concluded that, in the case that it is desired to balance the rotor for a wide speed band, the procedure using Genetic Algorithms more adequate. However, in the case the objective is only to balance the rotor for its operating speed, the procedure involving Neural Network is more efficient. About the Genetic Algorithms the initial population was maintained constant (500) for all cases. Crossover probability was 5%. The maximum number of generations was fixed to 50, was reached a value of the objective function of $2 \cdot 10^{-8}$. Further investigation will use real experimental data to validate the methodology developed.

7. Acknowledgements

The first author is thankful to FAPEMIG for his doctorate scholarship.

8. References

Assis, E., 1999, "Uso De Técnicas De Otimização Para Auxiliar O Projeto E Identificar Parâmetros De Máquinas Rotativas", Tese de Doutorado, UFU, Uberlandia –MG, Brasil.

Bishop, R.E.D., Glandell, G.M.L., 1959, "the vibration and balancing of an unbalanced flexible rotor", Journal of mechanical Engineering Science, 1, pp. 66-77.

Foiles W.C., Allaire P.E., Guter E.J., 1998, "Review: Rotor Balancing", Shock and Vibration 5, pp. 325-336.

Gnielka, P., 1983, "Modal Balancing of Flexible Rotors Without Test Runs : An Experimental Investigation", Journal of Sound And Vibration, 90, pp. 157-172.

Haykin, S. S., 1998, "Neural Networks: A Comprehensive Foundation (2nd Edition)", Prentice Hall.

ISO 11342, 1998, International Organization For Standardization, "Methods And Criteria For The Mechanical Balancing Of Flexible Rotors".

Lacerda, H., 1990, "Racionalização do Método dos Coeficientes de influência Aplicado ao Balanceamento de Rotores Flexíveis", Dissertação de Mestrado, UFU, Uberlandia –MG, Brasil.

Masters, T., 1993, "Practical Neural Network Recipes in C++", Morgan Kaufmann Publishers.

Steffen , V., Lépore N. F. P., 1983, "Simplified Methods for Rotor Systems Frequencies Using Classical Perturbations Techniques ", Proc. Of the 9th Canadian Congress of Applied Mechanics, Askatoon, Canada.

Vanderplaats, G., 1998, "Numerical Optimization Techniques for Engineering Design: With Applications", McGraw Hill College Division.

Vaqueiro, J.; 1989, "Balanceamento de Rotores Flexíveis", Tese de Mestrado, UNICAMP, Campinas – SP, Brasil.

Xu, B, Qu, L. ,Sun, R., 2000, "The Optimization Technique-Based Balancing of Flexible Rotors Without Test Runs", Journal of Sound And Vibration, 238: pp. 877-892.