Amin Houari[1], Salah Amrouone[3], Kouider Madani[1], Kouider Bendine[3], Samir Lecheb[2], Khalissa Saada[3]*

OPTIMISATION METHODS APPLIED TO THE DISTRIBUTION OF ROTOR BLADES IN A TURBOFAN ENGINE

PRIMENA METODA OPTIMIZACIJE NA RASPODELJENOST ROTORSKIH LOPATICA U TURBOMLAZNOM MOTORU

Originalni naučni rad / Original scientific paper
Rad primljen / Paper received: 18.6.2024
https://doi.org/10.69644/ivk-2025-01-0072

Adresa autora / Author's address:
[1] Laboratoire de Matériaux, et Mécanique des Structures (LMMS), Université SBA. Sidi Bel Abesse, Algérie
A. Houari https://orcid.org/0009-0004-2617-2182
K. Madani https://orcid.org/0000-0003-3277-1187
[2] Faculty of Technology, Mechanical Engineering Department, University M'Hamed Bougara of Boumerdes, Algeria
S. Lecheb https://orcid.org/0000-0003-1237-0220
[3] Laboratoire de Matériaux, et Mécanique des Structures (LMMS), Université de M'sila. Algérie :
S. Amroune https://orcid.org/0000-0002-9565-1935
K. Saada https://orcid.org/0000-0002-3025-1287
*email: salah.amroune@univ-msila.dz

Keywords
- blade
- distribution
- static balancing
- gas turbine
- MATLAB®

Abstract

The primary aim of this research is to enhance the distribution of rotor blades in a turbofan engine to achieve superior static balancing. To this end, an advanced optimisation method is proposed to ensure that the mass imbalance correction stays within the allowed tolerance limits, even after the rotor blades' positions are altered during maintenance activities. This proposed method leverages a genetic algorithm that is refined through the integration of various optimisation techniques. The use of MATLAB® code plays a crucial role in this process, providing the computational power and flexibility needed to implement and test the optimisation algorithm effectively. In practical terms, the method involves adjusting the positions of the rotor blades to minimise the static imbalance, which can significantly impact the performance and longevity of the engine. By maintaining the mass correction within specified tolerances, the engine can operate more smoothly and efficiently, reducing wear and tear and potentially extending its operational lifespan. The integration of MATLAB® allows for precise calculations and simulations, ensuring that the proposed adjustments are both feasible and effective in real-world scenarios. This approach not only improves the immediate balance of the rotor but also ensures that the balance is maintained over time, even with the positional changes that occur during routine maintenance.

Ključne reči
- lopatica
- raspodela
- statička uravnoteženost
- gasna turbina
- MATLAB®

Izvod

Primarni cilj ovog istraživanja je poboljšanje raspodeljenosti rotorskih lopatica turbomlaznog motora, kako bi se postigla superiorna statička uravnoteženost. U tu svrhu je predložena napredna metoda optimizacije, kako bi se u granicama tolerancije osigurala korekcija neizbalansiranosti mase, čak i pri promeni položaja rotorskih lopatica prilikom aktivnosti održavanja. Preložena metoda sadrži genetski algoritam koji je dopunjen integrisanjem različitim metodama optimizacije. Upotreba programa MATLAB® igra ključnu ulogu u ovom postupku, obezbeđujući računarsku snagu i prilagodljivost, potrebnih za efikasnu implementaciju i testiranje algoritma optimizacije. U praksi, metoda obuhvata prilagođavanje položaja lopatica rotora, kako bi se minimizirala statička neuravnoteženost, koja bi imala značajnog uticaja na performanse i vek motora. Održavanjem korekcija mase unutar preciziranih tolerancija, motor funkcioniše ravnomernije i efikasnije, smanjuju se trošenje i oštećenja uz potencijalno produžavanje radnog veka. Integracijom MATLAB® dobija se tačan proračun i simulacije, čime se postiže da su predložena izvođenja prilagodljiva i efikasna u realnim uslovima. Ovakav pristup ne samo da poboljšava neposrednu uravnoteženost rotora, već obezbeđuje da se ta uravnoteženost održava tokom vremena, čak i pri promenama položaja koje se dešavaju u toku rutinskog održavanja.

INTRODUCTION

Rotating machinery, particularly turbomachinery, plays a crucial role in the field of aeronautics, powering aircraft through the skies with remarkable power and efficiency. Among these machines, turbofan engines hold a prominent position. Designed with cutting-edge engineering and advanced technology, turbofan engines are the preferred choice for many commercial and military aircraft.

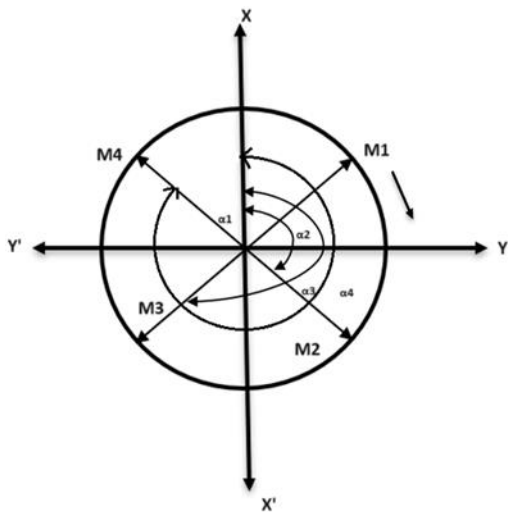
Vibrations in rotating machinery are minimised by addressing the primary cause of vibrations: imbalance in the rotor blades. An imbalance in the fan blades can cause excessive vibrations that may damage the engine and shorten its lifespan. This imbalance can result from differences in blade weights or from damage affecting their symmetry. While it is impossible to completely eliminate vibrations, it is generally acceptable to reduce them to a level below the prescribed threshold for a certain quality class of machinery. Balancing the rotor blades extends the bearing life, reduces vibrations, audible noise, and losses.

Numerous researchers have conducted studies in this field. For instance, Amroun et al. /1/ developed a computer-aided blade distribution technique for reducing rotor imbalance in rotating machinery. Yuan /2/ explored optimising blade arrangement to absorb vibrations in aircraft engines using an artificial ant colony algorithm. Li et al. /3/ applied the ant colony algorithm to blade arrangement, successfully reducing turbine imbalance to 1 % of the standard value over a year, achieving a reduction in turbine imbalance from over 99 % to 1 % of the official value. Sun et al. /4/ proposed a blade sorting method based on the Cloud Adaptive Genetic Algorithm (CAGA) to optimise the imbalance of asymmetrical rotor blades in aircraft engines. Huiqun et al. /5/ developed an optimisation model using the CUDA framework to increase computation speed, which also helped reduce the amplitude of forced vibration responses in bladed disk systems, bringing them within permissible engineering ranges. Dai et al. /6/ applied this approach to the optimal arrangement of turbine blades using a genetic algorithm, addressing the issue of lengthy computation times and the difficulty of obtaining optimal results in previous exhaustive optimisations of large numbers of blades. Rahimi /7/ used a genetic algorithm to solve an optimisation problem aimed at finding the least unfavourable response of a bladed disk assembly.

The primary objective of our optimisation programme is to achieve a distribution of turbofan engine blades such that the stage imbalance remains within acceptable tolerance limits, or at the very least, minimises it.

HYPOTHESIS OF STATIC BALANCING OF ROTORS

An imbalance is the mass added or removed to make a correction during a balancing operation. For calculating the imbalance and the correction angle, there are two methods. To demonstrate the relationship that allows for calculating the imbalance and the correction angle, we will consider four masses, M_1, M_2, M_3, and M_4, which are located at distances r_1, r_2, r_3, and r_4 from the centre of rotation, with angles α_1, α_2, α_3, and α_4 (where angle α_i indicates the position of the mass relative to the rotating reference frame). As shown in Fig. 1.



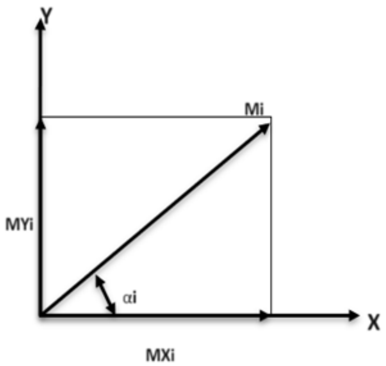


Figure 1. Description of imbalance of a disc-shaped rotor in the same plane.

The simplest case is represented by a disk-shaped rotor mounted perpendicular to the shaft axis. If the rotor rotates with a continuous angular speed of ω (rad/s), each elementary mass m_i generates on its radius $\vec{r}_i$, an inertial force $\vec{F}_i$,

$$\vec{F}_i = M_i \vec{r}_i \omega^2 \,. \tag{1}$$

The opposite force of the same intensity, the force of inertia, is called centrifugal force, its expression is:

$$\vec{F}_c = M_c \vec{r}_c \omega^2 \,. \tag{2}$$

The vector sum of the centrifugal forces of all elements is the centrifugal force which acts on the bearings, and it is expressed by:

$$\vec{F} = M_1\vec{r}_1\omega^2 + M_2\vec{r}_2\omega^2 + M_3\vec{r}_3\omega^2 + M_4\vec{r}_4\omega^2 + \dots \tag{3}$$

So:

$$\vec{F}_r = \sum_{i=k}^{n} M_i \vec{r}_i \omega^2 \quad \text{(N)} \,. \tag{4}$$

To ensure that a rotor becomes perfectly balanced, the equilibrium condition is: $\vec{F}_r = 0$. Now, the question is how to best express the unbalance. We can consider the residual centrifugal force as arising from an unbalance m_c, r_c (Eq.(5)) and then simplify the influence of the rotational speed on both sides (Eq.(6)):

$$\sum_{i=k}^{n} M_i \vec{r}_i \omega^2 = M_c \vec{r}_c \omega^2 \quad \text{(N)} \,, \tag{5}$$

$$\sum_{i=k}^{n} m_i \vec{r}_i = \vec{U} \quad (\text{g}\cdot\text{mm}) \,. \tag{6}$$

To solve these equations mathematically, divide each force into its x and y components:

$$\sum_{i=k}^{n} M_i r_i \cos\alpha_i = U \cos\alpha \,, \tag{7}$$

$$\sum_{i=k}^{n} M_i r_i \sin\alpha_i = U \sin\alpha \,. \tag{8}$$

The addition of square roots of the two equations above, Eq.(6) and Eq.(7), gives:

$$U = \sqrt{\left(\sum_{i=k}^{n} M_i r_i \cos\alpha_i\right)^2 + \left(\sum_{i=k}^{n} M_i r_i \sin\alpha_i\right)^2} \,, \tag{9}$$

$$\tan\alpha_c = \frac{\sum_{i=k}^{n} M_i r_i \sin\alpha_i}{\sum_{i=k}^{n} M_i r_i \cos\alpha_i} \,. \tag{10}$$

The analytical method was used to determine the correction mass (magnitude) and its angle (phase):

magnitude: $U = \sqrt{\left(\sum_{r=0}^{n} M_{xi}\right)^2 + \left(\sum_{r=0}^{n} M_{yi}\right)^2}$, (10)

phase (α):
$$\alpha = \arctan \frac{\sum_{r=0}^{n} M_{yi}}{\sum_{r=0}^{n} M_{xi}}. \qquad (11)$$

GRAPHICAL METHOD

The graphical method involves choosing a well-defined scale. Vectors representing the moments of each blade are drawn, and the resulting parallelogram must close. In our case, we use the analytical method because it is very simple to integrate mathematical formulas into a computer programme. However, if we use the graphical method with 4 blades (Fig. 2), the representation would be complicated.

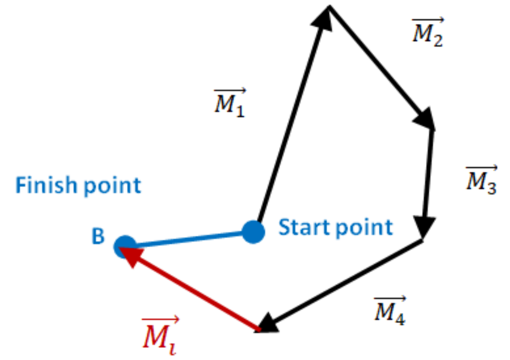


Figure 2. Vector diagram for balancing multiple rotating masses in the same plane.

METHOD FOR OPTIMISING THE DISTRIBUTION OF BLADES

In our problem, we will use genetic algorithms because they are known for their robustness and flexibility, allowing them to find optimal solutions even when the objective function is complex. They have the ability to avoid local minima by globally exploring the solution space. Additionally, these algorithms can be parallelised, enabling faster calculations for different scenarios simultaneously. To optimise the balance and correction angles in the fan with blades in our case, the best optimisation method depends on the specific characteristics of the problem, its nonlinearity, dimensionality, and constraints. In order to mitigate the influence of imbalance at the rotor level, the imbalance localisation module is proposed as follows:

$$F = \sum_{1}^{N} \min(U) \cdot \alpha$$

where: U is a one-dimensional vector composed of the magnitude of each blade; min(U) is the minimum value; and α is the blade localisation factor. To achieve this, we used MATLAB® code in our programme, utilising the genetic algorithm based on the preselections.

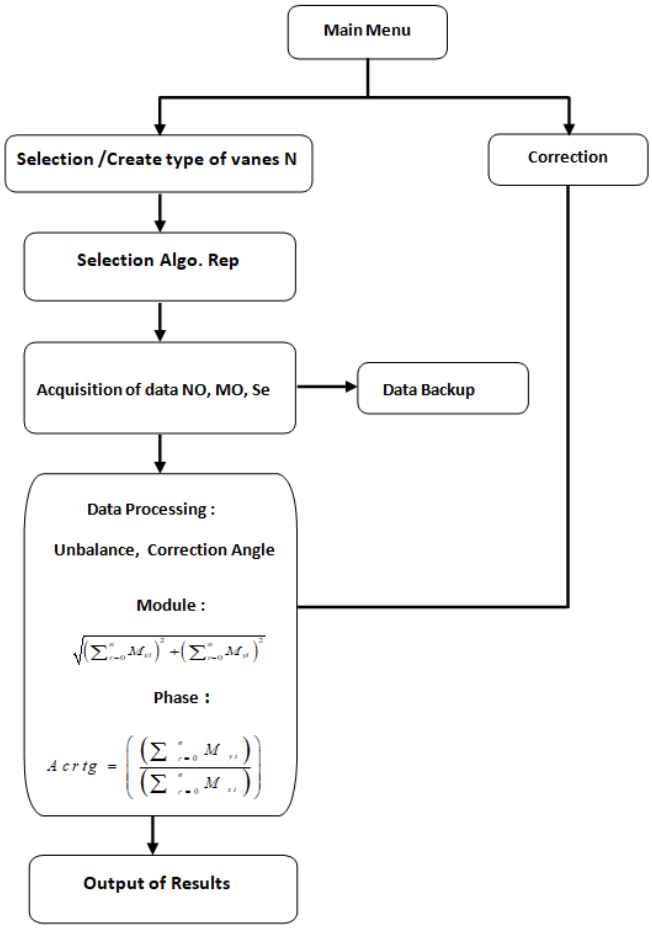


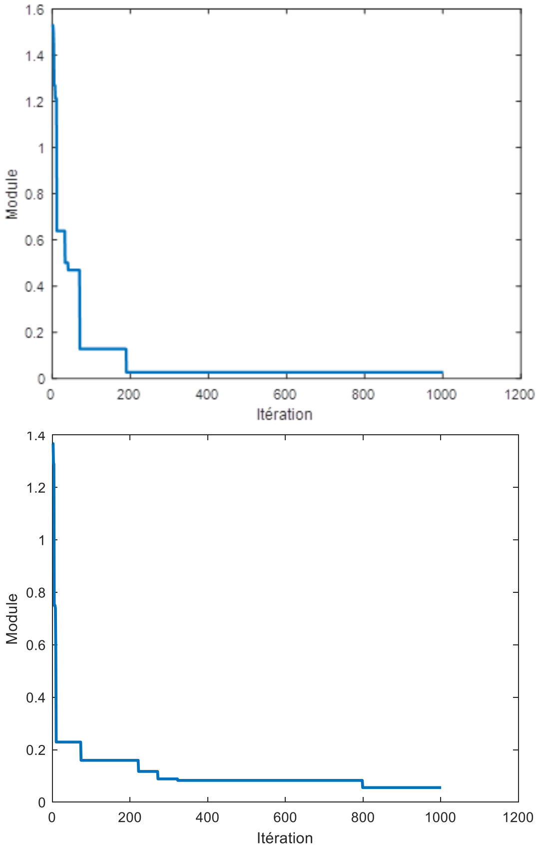
Figure 3. Blade distribution algorithm MATLAB® subroutine.

RESULTS AND VALIDATION

To visualise the effect of the optimisation method and the computational speed of the genetic algorithm, we selected the first stage blades of the fan rotor for a good distribution in order to optimise the algorithm assembly. Figure 4 shows an arrangement of the blades on a rotor disk following the best distribution.

Figure 4. Fan module in stage, /7/.

The convergence curve of the optimisation is presented in Fig. 5. It is evident that the genetic algorithm converges rapidly, reaching the optimal solution in only 190 iterations. We observed that horizontal lines appear at various points in the resolution process each time our program is run, indicating new convergence with a different number of iterations in the mass correction range between 0.02 and 0.05.



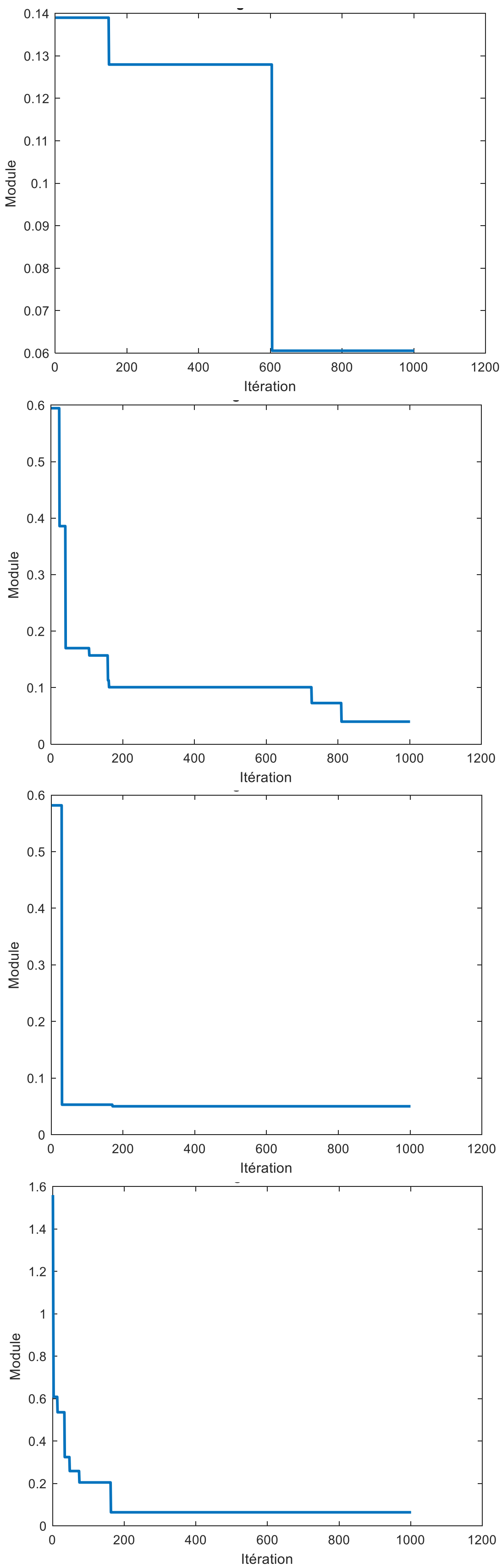
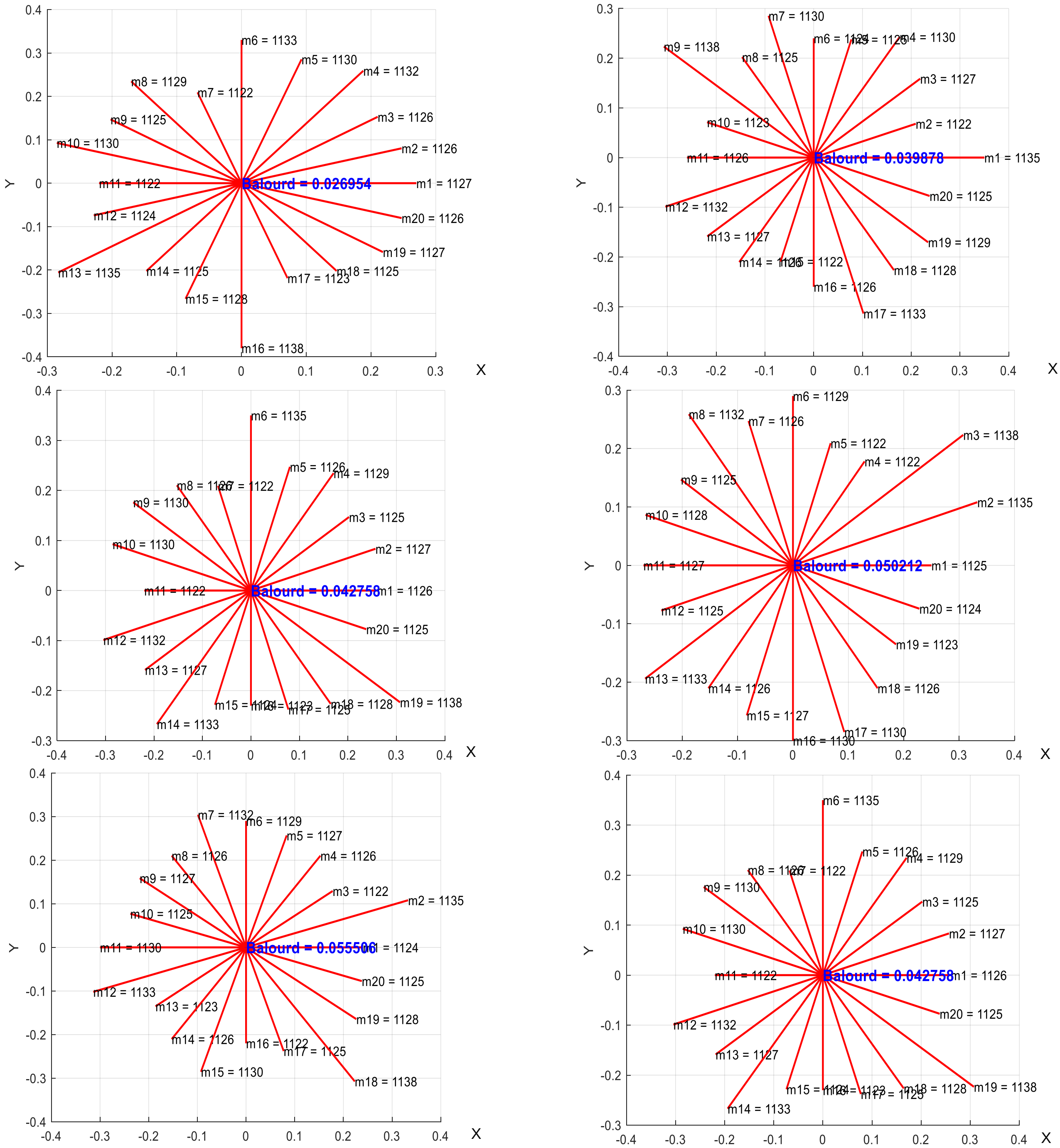


Figure 5. Comparison of optimisation convergence curves for each run.

According to Fig. 6 which presents the polar distribution of blade mass vectors, we observe that with each run, a new case of blade distribution appears.



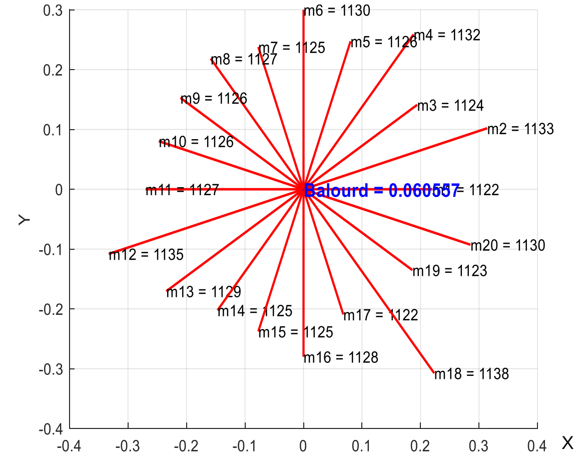
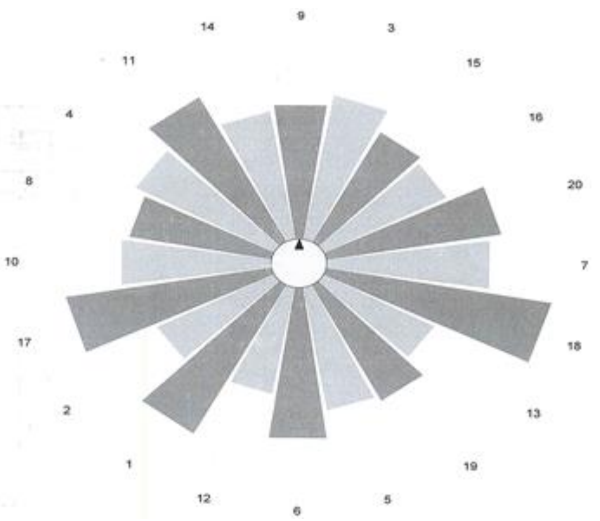


Figure 6. Results of the graphical simulation for each run.

Furthermore, the best permutation among the distributions is indicated in Fig. 7a, consistent with the results mentioned in the BLADIS.NET Proceedings, /8/.

From reading Table 1, we observe that the first run was the ideal solution after 180 iterations and was more accurate compared to the experimental results.

a)



b)

Pre-distribution model:	**7-weighing order**
Calculation unbalance:	**1 gmm**
Angular position:	**0 deg**
Vane clearance status:	**In tolerance (1gmm)**

Figure 7. Images recorded: a) simulation results; b) results of the machines mentioned in the BLADIS.NET proceedings, /8/.

Table 1. Evaluation of initial run.

Optimal value	Module	Phase	Iterations
first launch	0.0270	62.3282	190
second launch	0.0555	104.0375	800
third try	0.0606	-49.6506	600
fourth try	0.0399	78.5749	800
fifth try	0.0502	96.1141	180
sixth try	0.0640	-175.7489	180
experimental	1	90	6

CONCLUSION

The contribution of the optimisation method for blade distribution in a rotating machine based on genetic algorithms has been studied. This study shows that:
- adequate blade distribution can effectively minimise the amplitude of vibrations induced by imbalance;
- the optimisation method in this article offers advantages such as fast convergence in a wide search space, strong local search capability, and optimal results;
- at the same time, it has been demonstrated that the method presented is particularly effective in solving large-scale distribution problems;
- the genetic algorithm optimisation provides a cost-effective calculation method, high precision, and efficiency when computation time is limited, resulting in good distribution.

The findings of this study are of significant guidance importance for blade distribution in a rotor, therefore, this article proposes applicable methods for blade distribution based on genetic algorithms.

REFERENCES

1. Amroune, S., Belaadi, A., Menasri, N., et al. (2019), *New approach for computer-aided static balancing of turbines rotors*, Diagnostyka, 20(4): 95-101. doi: 10.29354/diag/114621
2. Yuan, H.-Q., Zhang, L., Han, Q.-K., Song, L. (2012), *Optimization of mistuning blades arrangement for vibration absorption in an aero-engine based on artificial ant colony algorithm*, J Vibr. Shock, 31(11): 169-172.
3. Li, Y., Yuan, H.-Q., Yang, S. (2013), *Optimization on mistuning blades arrangement of vibration absorption based on genetic particle swarm algorithm in aero-engine*, Adv. Mater. Res. 655: 481-485. doi: 10.4028/www.scientific.net/amr.655-657.481
4. Sun, C., Xiao, P., Wang, X., Liu, Y. (2021), *Blade sorting method for unbalance minimization of an aeroengine concentric rotor*, 13(5): 832. doi: 10.3390/sym13050832
5. Yuan, H., Zhao, T., Yang, W., Pan, H. (2015), *Annealing evolutionary parallel algorithm analysis of optimization arrangement on mistuned blades with non-linear friction*, J Vibroeng. 17(8): 4078-4095.
6. Dai, C.-H., Zhu, Y.-F., Chen, W.-R. (2006), *Adaptive probabilities of crossover and mutation in genetic algorithms based on cloud model*, In: 2006 IEEE Information Theory Workshop - ITW '06 Chengdu, China, 2006, IEEE, pp.710-713. doi: 10.1109/ITW2.2006.323754
7. Rahimi, M., Ziaei-Rad, S. (2010), *Uncertainty treatment in forced response calculation of mistuned bladed disk*, Math. Comp. Simul. 80(8): 1746-1757. doi: 10.1016/j.matcom.2009.07.002
8. Houari, A. (2018), *Contribution to the study of static balancing of turbine rotors (Contribution à l'étude de l'équilibrage statique des rotors turbines)*, Master Thesis, Faculty of Technology, Department of Mechanical Engineering, Mohamed Boudiaf University - M'sila, Algeria, https://repository.univ-msila.dz/items/4e627587-803f-4231-a187-5c481633b199/full

Appendix

Programme

```
% Initialiser les moments de masse
m = [1127 1129 1125 1124 1130 1128 1138 1122 1126 1125 1130 1122
1133 1123 1135 1126 1125 1127 1132 1126 ];
num_iterations = 1000; % Augmenter le nombre d'itérations de la
recherche aléatoire
ObjModule = zeros(1, num_iterations + 1);
ObjPhase = zeros(1, num_iterations + 1);
ObjE = zeros(2, length(m), num_iterations + 1);
[Module, Phase, E] = ObjecF(m);
ObjModule(1) = Module;
ObjPhase(1) = Phase;
ObjE(:, :, 1) = E;
 % Boucle de recherche aléatoire pour minimiser le balourd
for i = 1:num_iterations
  [Module, Phase, E] = ObjecF(m);
 ObjModule(i + 1) = Module;
 ObjPhase(i + 1) = Phase;
 ObjE(:, :, i + 1) = E;
 if ObjModule(i + 1) <= ObjModule(i)
   ObjModule(i + 1) = ObjModule(i + 1);
   ObjPhase(i + 1) = ObjPhase(i + 1);
   ObjE(:, :, i + 1) = ObjE(:, :, i + 1);
 else
   ObjModule(i + 1) = ObjModule(i);
   ObjPhase(i + 1) = ObjPhase(i);
   ObjE(:, :, i + 1) = ObjE(:, :, i);
 end
end
 % Afficher la valeur minimale du balourd dans la fenêtre de commande
disp('Valeur minimale du balourd :')
disp(ObjModule(end))
 % Tracer la ligne de correction de masse optimisée finale
figure
line([0, ObjModule(end) * cosd(ObjPhase(end))], [0, ObjModule(end) *
sind(ObjPhase(end))], 'Color', 'Blue', 'LineWidth', 2.5)
title('Vecteur de Correction de Masse')
xlabel('X')
```

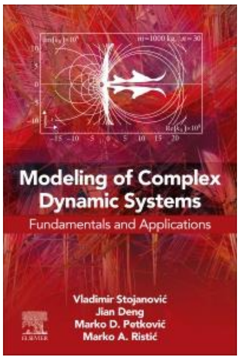
```
ylabel('Y')
grid on
% Tracer la convergence du Module
figure('Color', 'w')
set(gca, 'FontSize', 20, 'FontName', 'Times New Roman')
plot(ObjModule, 'LineWidth', 2)
ylabel('Module')
xlabel('Itération')
title('Convergence du Module')
% Tracer la convergence de la Phase
figure('Color', 'w')
set(gca, 'FontSize', 20, 'FontName', 'Times New Roman')
plot(ObjPhase, 'LineWidth', 2)
ylabel('Phase (degrés)')
xlabel('Itération')
title('Convergence de la Phase')
% Afficher les valeurs optimales
disp('Valeurs optimales de la distribution de masse :')
disp(ObjE(:, :, end))
disp('Valeur optimale du Module :')
disp(ObjModule(end))
disp('Valeur optimale de la Phase :')
disp(ObjPhase(end))
 % Tracer la distribution polaire des vecteurs de masse
figure('Color', 'w')
hold on
% Générer les angles pour chaque élément
theta = linspace(0, 2 * pi, length(m) + 1); % +1 pour compléter le cercle
theta(end) = []; % Supprimer le dernier élément pour éviter la duplication
 % Tracer les lignes rouges pour chaque masse et ajouter des annotations
for k = 1:length(m)
  x_end = (ObjE(1, k, end) - 1100) * cosd(ObjE(2, k, end));
  y_end = (ObjE(1, k, end) - 1100) * sind(ObjE(2, k, end));
  line([0, x_end * 0.01], [0, y_end * 0.01], 'Color', 'Red', 'LineWidth', 1.5);
  text(x_end * 0.01, y_end * 0.01, ['m' num2str(k) ' = ' num2str(ObjE(1, k,
end))], 'FontSize', 10, 'Color', 'black'); % Ajouter des annotations pour
chaque masse
end

% Tracer le vecteur de balourd en bleu
x_balourd = ObjModule(end) * cosd(ObjPhase(end)) * 0.01;
y_balourd = ObjModule(end) * sind(ObjPhase(end)) * 0.01;
line([0, x_balourd], [0, y_balourd], 'Color', 'Blue', 'LineWidth', 2.5);
% Ajouter l'annotation pour le balourd
text(x_balourd, y_balourd, ['Balourd = ' num2str(ObjModule(end))],
'FontSize', 12, 'Color', 'blue', 'FontWeight', 'bold');
 % Ajouter un titre et des labels
title('Distribution Polaire des Vecteurs de Masse')
xlabel('X')
ylabel('Y')
grid on
hold off
% Fonction objectif
function [Module, Phase, E] = ObjecF(m)
  % Créer le vecteur d'angles correspondant
  Alfa = linspace(0, 360, length(m) + 1); % Inclure 0 et 360 degrés
  Alfa(end) = []; % Supprimer le point redondant de 360 degrés
   % Mélanger aléatoirement les moments de masse
  E = zeros(2, length(m));
  shuffledIndices = randperm(length(m)); % Générer une permutation
aléatoire
  E(1, :) = m(shuffledIndices); % Appliquer la permutation aux masses
  E(2, :) = Alfa; % Attribuer les angles correspondants
   % Afficher les numéros des masses
  for k = 1:length(m)
    fprintf('m%d = %.0f\n', k, E(1, k));
  end

  % Calculer le vecteur résultant
  t = 0.1; % Facteur d'échelle pour la visualisation
  Mx = E(1, :) * t .* cosd(E(2, :));
  My = E(1, :) * t .* sind(E(2, :));
  Module = sqrt(sum(My)^2 + sum(Mx)^2); % Calcul du module
  Phase = atan2d(sum(My), sum(Mx)); % Calcul de la phase
end
```

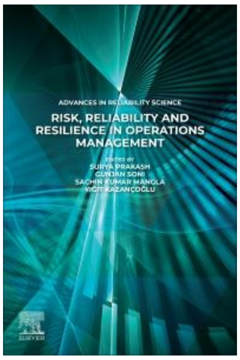
© 2025 The Author. Structural Integrity and Life, Published by DIVK (The Society for Structural Integrity and Life 'Prof. Dr Stojan Sedmak') (http://divk.inovacionicentar.rs/ivk/home.html). This is an open access article distributed under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License

New Elsevier Book Titles – Woodhead Publishing – Academic Press – Butterworth-Heinemann – …

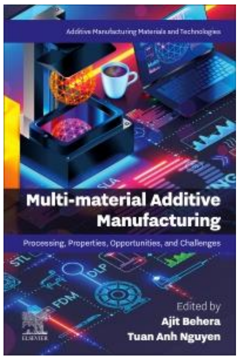


Modeling of Complex Dynamic Systems Fundamentals and Applications, 1st Edition
Vladimir Stojanović, Jian Deng, Marko D. Petković, Marko A. Ristić
Elsevier, April 2025
ISBN: 9780443239427
EISBN: 9780443239434

Additive Manufacturing of Continuous Fiber Reinforced Polymer Composites, 1st Edition
Xiaoyong Tian, Tengfei Liu, Zhanghao Hou, Lingling Wu
Elsevier, April 2025
ISBN: 9780443135996
EISBN: 9780443136009



Risk, Reliability and Resilience in Operations Management, 1st Edition
Vahid Monfared
Elsevier, April 2025
ISBN: 9780443298127
EISBN: 9780443298134



Multi-material Additive Manufacturing Processing, Properties, Opportunities, and Challenges, 1st Edition
Editors: Ajit Behera, Tuan Anh Nguyen
Elsevier, April 2025
ISBN: 9780443292286
EISBN: 9780443292293