IJRAME
ISSN (ONLINE): 2321-3051

INTERNATIONAL JOURNAL OF RESEARCH IN AERONAUTICAL AND MECHANICAL ENGINEERING

Flutter prediction using neural networks-based hybrid optimization scheme

Prabhu L[1], Dr J Srinivas[2]

[1]Graduate student, Department of Mechanical Engineering, NIT Rourkela-769008, prabhulakshmananr@gmail.com
[2] Associate Professor, Department of Mechanical Engineering, NIT Rourkela-769008, srin07@yahoo.co.in

Abstract

In this paper, an unsteady aerodynamic model is used to study the influence of geometrical parameters on the flutter velocity of an airfoil section. The most affecting geometric variable on flutter speed is predicted using theory of ANOVA. A single hidden layer back-propagation neural network model is trained with different sets of geometric parameters and flutter velocities. The optimal parameters of the section corresponding to a known flutter velocity are obtained through the trained neural network in combination with hybrid optimization scheme formulated based on micro genetic algorithm (µGA) with micro differential evolution (µDE). Results are presented with a benchmark airfoil geometry.

Keywords: *Airfoil, Flutter, Artificial neural network, Hybrid optimization, Aeroelasticity*

1. Introduction

The study of the effect of aerodynamic forces over elastic bodies is known as aeroelasticity. The aeroelastic problems are different from theory of elasticity. The theory of elasticity is concerned about the stress and deformation of an elastic body subjected to external loading, which is independent of deformation of the body. In aeroelasticity problems the aerodynamic loading purely depends on posture of the body relative to flow; unless the value of elastic deformation is not known the aerodynamic loading cannot be predicted. Aeroelastic problems are solved mainly to get the stability regions. The flutter of an airfoil is a dynamic aeroelastic instability which involves the interactions of aerodynamic, elastic and inertia forces. When an airfoil is subjected to the air flow whose speed increase gradually, the rate of damping of the oscillation of airfoil initially increases and then rapidly falls down. At particular speed where, the oscillation of airfoil could be maintained with steady amplitudes is known as critical flutter speed. Even for small disturbances of airfoil above the critical speed leads to violent oscillation and this state is known as flutter. The effect of airfoil parameters on flutter speed were empirically given by Theodorsen and Garrick [1] in early 1940s.

In order to predict and control the flutter conditions, different methods of aerodynamic and structural modelling have been used in practice. The finite element modelling is most commonly used approach [2].

While predicting flutter in subsonic wing, Moosavi et al. [3] employed a procedure based on Galerkin's method to determine the critical speed with and without considering compressibility effect using quasi-steady aerodynamic model. Haddapour et al. [4] compared the aeroelastic behaviour of wing structure using quasi-steady aerodynamic model with that of unsteady aerodynamic model to know the best one for analysis. Riccy [5] presented a numerical study of flutter by using Leishman's state space model under unsteady aerodynamic loads to analyze the stability region and the effectiveness was verified by comparing with Theodorsen and Garricks' approach. Brain et al. [6] discussed various tools to examine the aeroelastic stability of wind turbine blade and suggested a design tool. Park et al. [7] utilized such unstable oscillations and converted them as the useful energy in the form of an electromagnetic energy harvester. In more recent works, frequency-domain identification for prediction of flutter in cantilevered wing [8] and experimental study of flutter behavior of flexible airfoils supported with laminated composite plates [9] were also illustrated in this line.

In order to understand primarily the flutter concept, an implicit model relating the object geometry with flutter velocity is required. Neural networks and optimization techniques are being recently used in several applications of aerodynamics. The critical flutter load and boundary conditions of tapered beam were obtained by Takahashi [10] by multilayer perceptron network trained with the back propagation algorithm. Chen et al. [11] used two different back-propagation neural networks to identify the flutter derivatives. Mannarino and Mntegazza [12] extended ANN application to find the aeroelastic limit cycle oscillations. Use of neural networks for function approximation and utilize this further for optimizing the outputs is a usual task. A hybrid algorithm holds the best properties of various optimization techniques. Recently, several works proposed hybrid optimization schemes [13-16].

In this paper first part deals with identification of flutter speed using unsteady aerodynamic model and the influence of aeroelastic parameters on flutter speed is analyzed. In the second part a back-propagation neural network is trained with aeroelastic parameters and flutter speed and the weights are acquired to identify the relationship between flutter speed and airfoil parameters. In the last part, the parameters at a desired flutter speed are obtained by using hybrid optimization technique based on micro genetic algorithms and differential evolution schemes. The content of the paper is organized as follows: section 2 presents related equations of motion including lift and pitching moment expressions. Section 3 deals with parametric studies and section 4 explains training of neural network and hybrid optimization methodology along with results and discussions.

2. Equations of motion

A two-degree of freedom of airfoil is considered as shown in Fig.1.

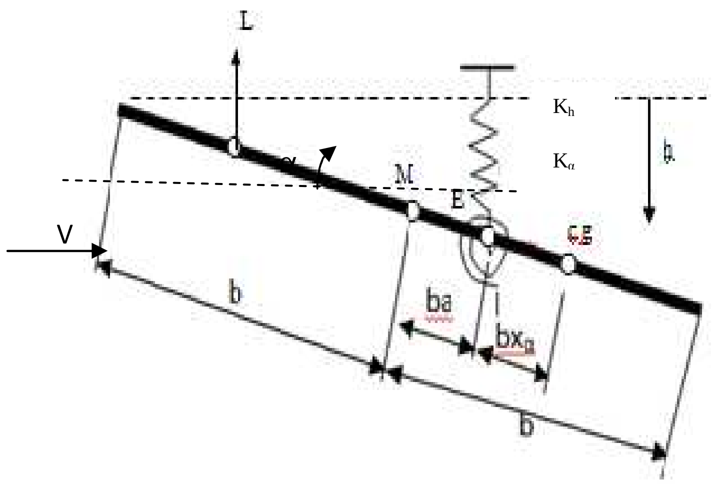

Fig. 1: Two degree of freedom model

Here, h is plunging translation and α is a pitching rotation about elastic axis. It is subjected to a lift force L and a pitching moment M_α. Here, E is elastic axis, M is mid chord, and c.g is centre of mass. The equations of motion are written as:

$$m\ddot{h} + S\ddot{\alpha} + hK_h = -L \qquad (1)$$

$$S\ddot{h} + I_\alpha \ddot{\alpha} + \alpha K_\alpha = M_\alpha \qquad (2)$$

where m, S, I_α, K_h and K_α are mass, static moment, mass moment of inertia, plunge coefficient and pitch coefficient respectively.

2.1. Aerodynamic model

In the unsteady aerodynamics, the lift and moment can be divided into two parts; one is apparent mass which is not associated with creation of vorticity and other is circulatory part, purely depends on the motion and influence of wake. The lift and moment expression for a sinusoidal oscillating airfoil is given by Theodorsen's unsteady aerodynamics as:

$$L = \pi\rho b^3\omega^2\left[\frac{h}{b}L_h + \alpha\left(L_\alpha - \left(\frac{1}{2}+a\right)L_h\right)\right] \qquad (3)$$

$$M_\alpha = \pi\rho b^4\omega^2\left[\left\{M_h - \left(\frac{1}{2}+a\right)L_h\right\}\frac{h}{b} + \left\{M_\alpha - \left(\frac{1}{2}+a\right)(L_\alpha + M_h) + \left(\frac{1}{2+a}\right)^2 L_h\right\}\alpha\right] \qquad (4)$$

where

$$L_h = 1 - i2C(k)\frac{1}{k} \qquad (5i)$$

$$L_\alpha = \frac{1}{2} - i\frac{1+2C(k)}{k} - \frac{2C(k)}{k^2} \qquad (5ii)$$

$$M_h = \frac{1}{2},\ M_\alpha = \frac{3}{8} - i\frac{1}{k} \qquad (5iii)$$

Here i denotes the imaginary part of complex number and k is the reduced frequency $k = \frac{\omega b}{V}$ with V as flow velocity and b as semi-chord length. By considering harmonic motion of airfoil, Eqs. (1-4) can be simplified and written in matrix form as a characteristic equation as follows:

$$\begin{bmatrix} \mu\left[1-\left(\frac{\omega_h}{\omega_\alpha}\right)^2\left(\frac{\omega_\alpha}{\omega}\right)^2\right] + L_h & \mu x_\alpha + L_\alpha - L_h\left(\frac{1}{2}+a\right) \\ \mu x_\alpha + M_h - L_h\left(\frac{1}{2}+a\right) & \mu r_\alpha^2\left[1-\left(\frac{\omega_\alpha}{\omega}\right)^2\right] + M_\alpha - (L_\alpha + M_h)\left(\frac{1}{2}+a\right) + L_h\left(\frac{1}{2}+a\right)^2 \end{bmatrix}\begin{bmatrix} \frac{h}{b} \\ \alpha \end{bmatrix} = 0 \qquad (6)$$

This has non-trivial roots if determinant is zero and it results in two sets of complex conjugate roots. By varying the flow velocity V over a range and obtaining different roots, we can plot real and imaginary terms. The intersection these terms occurs at a particular value of V (corresponding to 1/k) and is theoretically given by [17]:

$$U_{cr} = \frac{b\omega_\alpha}{k\sqrt{X}} \qquad (7)$$

3. Influence of geometric parameters on flutter velocity

3.1. Conventional method

The flutter velocity is strongly affected by distance between mid-chord and elastic axis (a_h), distance between elastic axis and center of mass (x_α) and bending torsion frequency ratio (ω_h/ω_α) as well as many other parameters. Table-1 shows the fixed parameters of airfoil employed in present work.

Table: 1 Aeroelastic parameters

Parameters	Value
μ	76
a_h	-0.15
x_α	0.25
r_α^2	0.388
b	0.127 m
ω_α	64.1 rad/s
ω_h	55.9 rad/s

In order to find the influence of each parameter on the flutter speed, three levels are considered as depicted in Table 2. By considering one of the parameters in level 1 as constant and combining with other parameters in all level results in 9 combinations, by doing in similar fashion it gives 27 combinations. Other parameters for aeroelastic analysis are considered as constant.

Table: 2 Aeroelastic parameter levels

Level	a_h	x_α	ω_h/ω_α
1	-0.2	0.1	0.4
2	-0.175	0.2	0.6
3	-0.15	0.25	0.87

Keeping a_h as constant, flutter velocities are obtained as shown in Fig 2. From Fig 2 it is clear that flutter speed is almost equal with very small variation for different a_h value (-0.2, -0.175, and -0.15) with different x_α and ω_h/ω_α combinations. Increase in x_α and ω_h/ω_α leads to an increase in flutter velocity. When ω_h/ω_α is 0.4 and 0.6 the flutter velocity decreases with corresponding to increase in x_α but when ω_h/ω_α is 0.87 the flutter velocity increases with corresponding to increase in x_α.

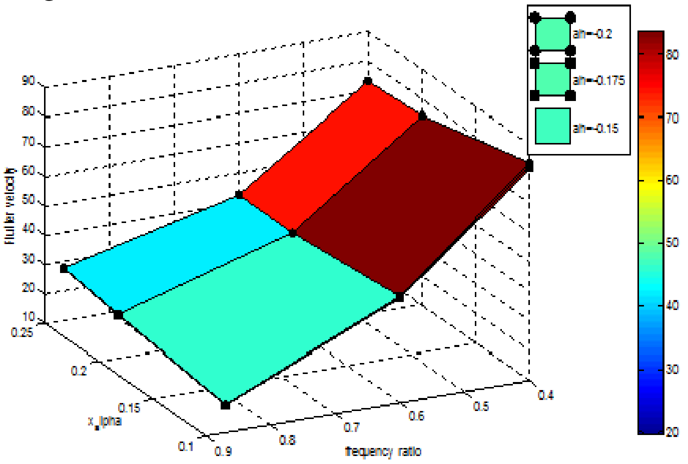

Fig. 2: Variation of flutter velocity with a_h as constant

Now x_α is taken as constant, then flutter velocity is plotted against a_h and ω_h/ω_α in Fig 3. Flutter velocity is varying with different x_α values (0.1, 0.2, and 0.25) can be seen from Fig 3. Increase in a_h (-0.15 to -0.2) and ω_h/ω_α gives increased value of flutter velocity. At x_α is 0.1, ω_h/ω_α is 0.4 and for various a_h the flutter velocity is high and it is low when ω_h/ω_α is 0.87 comparing to x_α (0.2 and 0.25). When ω_h/ω_α is 0.6 the combination of various a_h and $x_{\alpha,}$ the flutter velocities are with small difference.

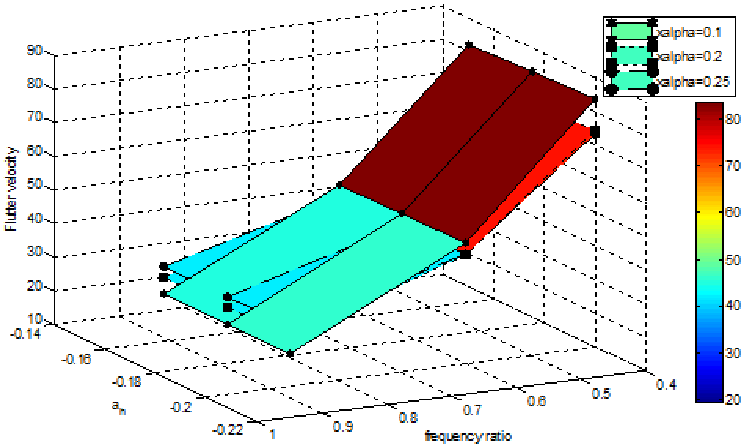

Fig. 3: Variation of flutter velocity with x_α as constant

At last ω_h/ω_α is taken as constant with various values and variation of flutter velocity is plotted in Fig 4. The increase in ω_h/ω_α results in decrease of flutter velocity. Decrease and increase of x_α and a_h (-0.15 to -0.2) leads to increase in flutter velocity. The combination of least values of ω_h/ω_α, x_α and a_h gives the high flutter velocity.

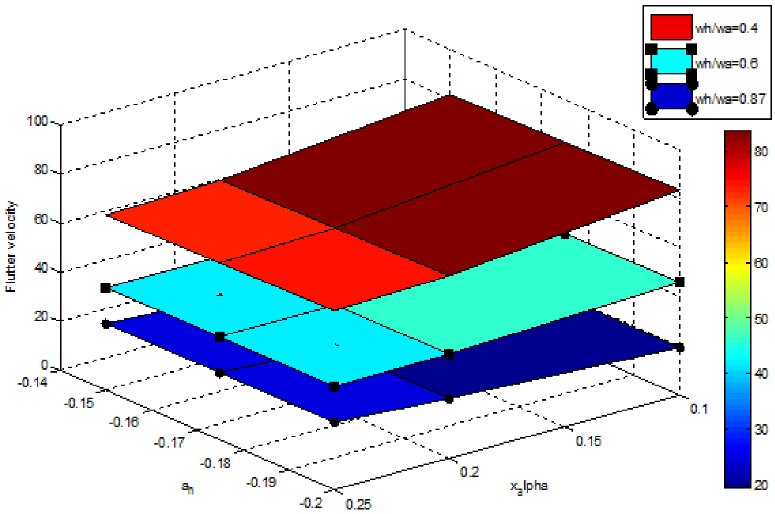

Fig. 4: Variation of flutter velocity with ω_h/ω_α as constant

3.2. Design of Experiments

In order to minimize the number of experiments, the method of design of experiments (DOE) is employed. The first step before using the DOE is to identify number of factors and levels. Once it is done orthogonal array is used to know the possible number of experiments. Taguchi method is employed for three level and three parameters given in Table 2, Taguchi L_9 (3*3) orthogonal array is used to predict the influence of parameters such as distance between mid-chord and elastic axis (a_h), distance between elastic axis and center of mass (x_α) and bending torsion frequency ratio (ω_h/ω_α) on the output factor namely flutter speed (U_{cr}). The experimental

Vol.2 Issue.4, April 2014. Pgs: 127-137

layout of an L_9 orthogonal array is shown in Table 3. As the essential objective is to maximize the flutter velocity, one has to predict the best level of each factor. Minitab 16 is used to carry out the Taguchi method; flutter speed of 9 combinations is predicted by using MATLAB code and fed into Minitab to get the Means graphs for analysis.

Table: 3: L9 Orthogonal array

Level	a_h	x_α	ω_h/ω_α	V_f
1	-0.2	0.1	0.4	85.31
2	-0.2	0.2	0.6	43.74
3	-0.2	0.25	0.87	27.83
4	-0.175	0.1	0.6	45.84
5	-0.175	0.2	0.87	25.19
6	-0.175	0.25	0.4	74.12
7	-0.15	0.1	0.87	19.72
8	-0.15	0.2	0.4	75.02
9	-0.15	0.25	0.6	42.37

Fig. 5 shows the variation of parameters against mean flutter speed. The parameter with higher value of Mean has very good influence over flutter speed. It graphically compares the level of an output variable for different input parameters, to gain an understanding of the effect of a parameter on the output. A relatively flat line shows it has little effect upon the output, while a line that has a lot of up and down movement indicates that as the factor changes; it has a greater mean effect. From Fig 5, it is observed that the parameter highly influencing the flutter velocity is frequency ratio ω_h/ω_α and other two have less influence. In the case of flutter speed (U_{cr}), the optimum parameters are a_h =-0.2, x_α= 0.10 and frequency ratio ω_h/ω_α=0.40 as shown in Fig. 5, this same as conventional one.

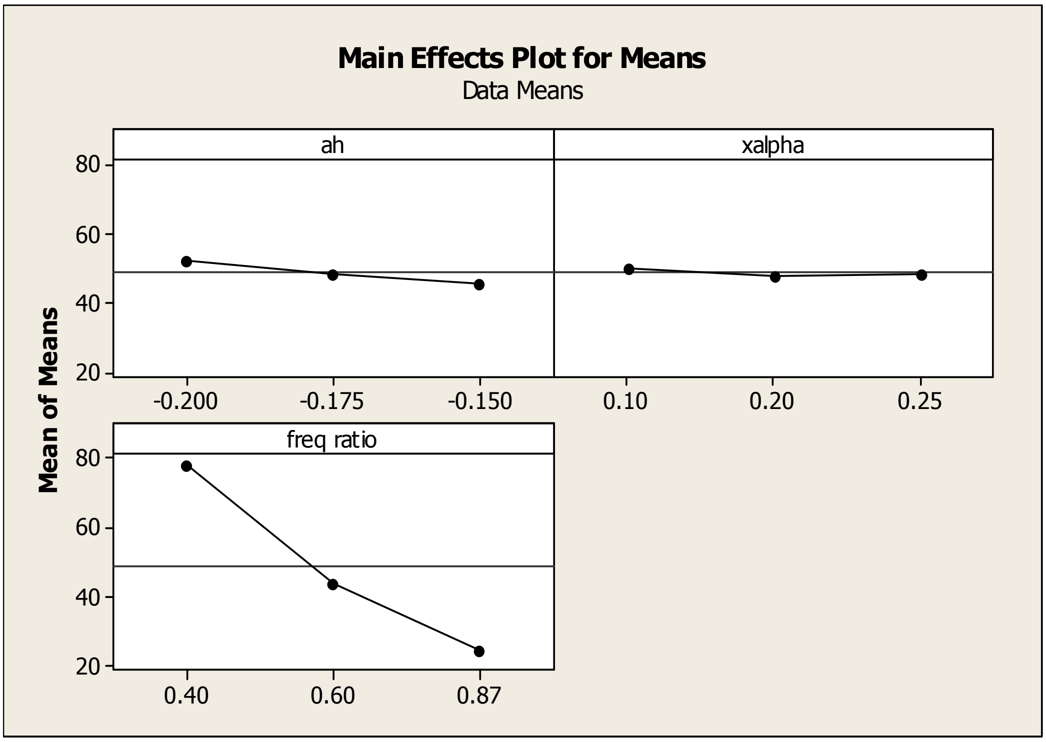

Fig. 5: Effect of parameters on flutter velocity

The aeroelastic system is statistically analyzed by Analysis of Variance (ANOVA). The output of ANOVA from Minitab is given in Table 4 and it also shows the frequency ratio has high influence (97.43%) compared to others.

Table: 4 ANOVA for aeroelastic analysis

Analysis of Variance for Means

Source	DF	Seq SS	Adj SS	Adj MS	F	P
ah	2	65.90	65.90	32.95	1.58	0.387
xalpha	2	10.10	10.10	5.05	0.24	0.805
freq ratio	2	4462.47	4462.47	2231.23	107.19	0.009
Residual Error	2	41.63	41.63	20.82		
Total	8	4580.10				

4. Artificial neural networks

The feed-forward back propagation (BP) network is a very popular model in neural networks and the architecture is shown in Fig. 6. As the name implies the computations are passed forward from the input to output layer, calculated errors are propagated back order to update the weights. Twenty seven training sets from three level parameters shown in Table 2 are used to train the 3-layer neural network with 4 neurons in hidden layer. Here a_h, x_α, and ω_h/ω_α are given as inputs and U_{cr} as output parameter. In-house code is written in MATLAB to train the network and to acquire the weights. Convergence pattern of error is shown in Fig. 7 and weights are shown in Table 5.

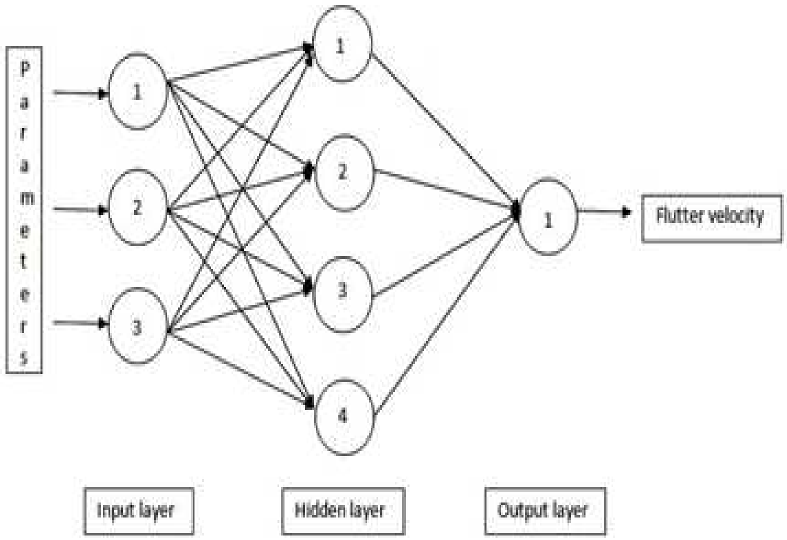

Fig. 6: Neural network structure

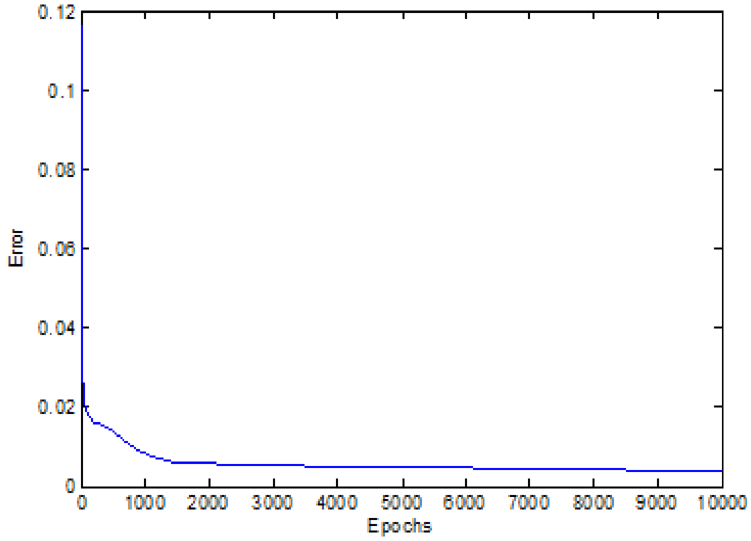

Fig. 7: Convergence of error

Table: 5 Weights

Weights	
Between input and hidden layer	-5.6978 0.0774 -1.5806 -1.9556
	-9.5137 -0.6191 -3.2522 -4.7738
	-5.4921 -4.8774 -4.9607 -0.5389
Between hidden and output layer	20.9597 12.7619 14.8446 -4.8813

4.1. Hybrid-optimization

4.1.1. Genetic algorithms

GA is a stochastic optimization technique based on principle of 'survival of the fittest'. GA has perfect balance between the robustness and the efficiency of survival in different environment. Applications of GA are increasing day by day in many engineering fields. The principle behind GA is Darwin's theory of evolution. The population members are strings similar to chromosomes, which are usually in binary representation of solution vectors. At the beginning, initial population is evaluated for fitness value. Starting with this information, the new generation of the population can be deduced using selection, crossover and mutation operators. This process is iterated up to convergence. Every gene (sub-string) in the chromosome consists of fixed number of binary digits and represents a single value.

Micro GA (µGA) operates on a population similar to GA as explained above. Unlike the GA, the mechanics of µGA allows for a very small population size with crossover alone. Though the population is very small the chances of becoming stagnant is high, once algorithm enters this situation reinitialization is the only solution to take back the algorithm in right way.

4.1.2. Hybrid µGA with µDE

In this paper a hybrid µGA is proposed which uses µDE for aeroelastic optimization studies. In this technique population initialization takes place randomly, the fitness values are evaluated and the ranking will be given, and new set of population is created by crossover. The new set of population is evaluated and if stagnant occurs, µDE is activated and it will be on till the best population is achieved comparing to population from µGA or twenty cycles, whichever is earlier. Once µDE is completed the improved populations are then passed to crossover. Crossover value taken is 0.99 and 0.05 for µGA and µDE correspondingly and mutation value for µDE is 0.8. Flowchart for hybrid µGA with µDE is shown in Fig. 8.

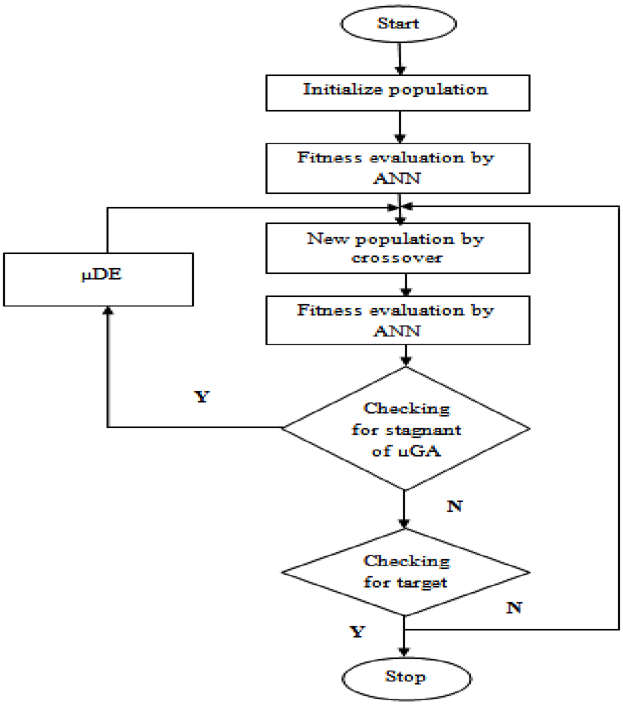

Fig. 8: Flowchart of hybrid µGA with µDE

The above said hybrid optimization technique is used to predict the influencing parameters for a flutter speed of 60 m/s. Each population consist of three parameters namely a_h, x_α, and ω_h/ω_α which are randomly initialized and the output U_{cr} is evaluated by the weights acquired through the neural network training. The fitness value of each population is obtained by:

$$Fitnessvalue = \sqrt{\left(U_{Target} - U_{Optim}(a_h, x_\alpha, \omega_h / \omega_\alpha)\right)^2} \quad (8)$$

The trends followed by the various optimization techniques are shown in Fig. 9 and the influencing parameters by optimization technique using ANN is shown along with the evaluation of parameters by weights and MATLAB code in Table 6.

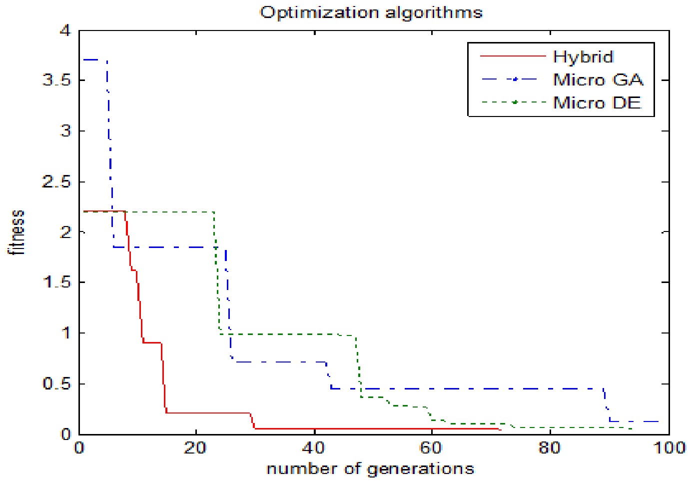

Fig. 9: Comparison of optimization techniques

Table: 6 Parameters by optimization

Parameters by optimization with ANN	**Flutter velocity** **ANN-Weights**	 **MATLAB code**
a_h=-0.1062 **x_α=0.2661** **ω_h/ω_α=0.4650**	59.98	57.50

5. Conclusion

This paper presented a parametric study based on unsteady aerodynamic model and various influencing parameters on flutter speed by Taguchi and ANOVA and the prediction is compared with conventional approach. It is observed that the frequency ratio is a highly influencing parameter. A 3-4-1 back-propagation neural network model was trained with twenty seven training sets to obtain the flutter velocities at unknown geometric data. Hybrid of µGA with µDE optimization tool was employed and it shows better performance comparing to µGA and µDE.

References

[1]. Theodorsen T, Garrick I E, 1940, Mechanism of flutter a theoretical and experimental investigation of the flutter problem, *NACA report 685*.

[2]. Mushim J J, Shokat A T, Tariq S T, 2012, Flutter speed limits of subsonic wings, *J Eng*, 18, 2, 163-183.

[3]. Moosavi M R, Naddaf Oskouei A, Khelli A, 2005, Flutter of subsonic wing, *Thin wall Struct*, 43, 4, 617-627.

[4]. Haddadpour H, Firouz –Abadi R D, 2006, Evaluation of quasi-steady aerodynamic modelling for flutter prediction of aircraft wings in incompressible flow, *Thin wall Struct*, 44, 9, 931-936.

[5]. Riccy K, 2013, Numerical study of flutter of a two dimensional aeroelastic system, In *Proc.World congress on engineering*, London, U.K.

[6]. Brian O C, Griffith D T, Brian R R, John E H, 2013, Impact of modelling approach on flutter predictions for very large wind turbine blade designs, In: *Proceedings AHS 69th annual forum*, phoenix., Arizona.

[7]. Park J, Morgenthal G, Kim K, Kwon S, Kincho L, Power evaluation for flutter-based electromagnetic energy harvester using CFD simulations, *J Intell Mat Syst Str* (In press) doi: 10.1177/1045389X14526954.

[8]. B.Gjerek, R.Drazumeric, F.Kosel, Flutter behavior of a flexible airfoil: Multiparameter experimental study, *Aerosp Sci Technol*, http://dx.doi.org/10.1016/j.ast.2014.04.002

[9]. J.Ertveldt, J.Lataire, R.Pintelon, S.Vanlanduit, Frequency-domain identification of time-varying systems for analysis and prediction of aeroelastic flutter, *Mech Syst Signal Pr*, http://dx.doi.org/10.1016/j.ymssp.2013.08.020.

[10]. Takahashi I, 1999, Identification for critical flutter load and Boundary conditions of a beam using neural network, *J Sound Vib*, 228, 4, 857-870.

[11]. Chen C H, Wu J C, Chen J H, 2008, Prediction of flutter derivatives by artificial neural networks, *J Wind Eng Ind Aerod*, 96, 925-1937.

[12]. Mannarino A, Mantegazza P, Nonlinear aeroelastic reduced order modeling by recurrent neural networks. *J Fluid Struct* (In Press), http://dx.doi.org/10.1016/j.jfluidstructs.2014.02.016i

[13]. Santana-Quintero L V, Hernandez-Diaz A G, Molina J, Coello C A, Caballero R, 2010, DEMORS: A hybrid multi-objective optimization algorithm using differential evolution and rough set theory for constrained problems, *Comput Oper Res*, 37, 470-480.

[14]. Zammer A, Mizra S M, Mizra N M, 2014, Core loading pattern optimization of a typical two-loop 300 MWe PWR using Simulated Annealing (SA) novel crossover Genetic Algorithms (GA) and hybrid GA(SA) schemes, *Ann Nucl Energy*, 65, 122-131.

[15]. Wang Y, 2014, Hybrid genetic algorithm with two local optimization strategies for travelling salesman problem, *Comput Ind Eng*, 14, 124-133.

[16]. Yuan X, Zhao J, Yang Y, Wang Y, 2014, Hybrid parallel chaos optimization algorithm with harmony search algorithm, *Appl Soft Comput*,17, 12-22.

[17]. Fung Y C, 1993, An introduction to the theory of Aeroelasticity, Dover publications. Inc, New York, Chap.6.