MULTIDISCIPLINARY DESIGN, ANALYSIS, AND OPTIMIZATION TOOL DEVELOPMENT USING A GENETIC ALGORITHM

Chan-gi Pak and Wesley Li
NASA Dryden Flight Research Center

Keywords: *flutter, genetic algorithm, MDO, multidisciplinary, optimization*

Abstract

Multidisciplinary design, analysis, and optimization using a genetic algorithm is being developed at the National Aeronautics and Space Administration Dryden Flight Research Center to automate analysis and design process by leveraging existing tools such as NASTRAN, ZAERO and CFD codes to enable true multidisciplinary optimization in the preliminary design stage of subsonic, transonic, supersonic, and hypersonic aircraft. This is a promising technology, but faces many challenges in large-scale, real-world application. This paper describes current approaches, recent results, and challenges for MDAO as demonstrated by our experience with the Ikhana fire pod design.

Nomenclature

AIC	aerodynamic influence coefficient
DC	DOT optimizer with continuous design variable
DFRC	Dryden Flight Research Center
DOT	design optimization tool
GA	genetic algorithm
GC	genetic optimizer with continuous design variable
GD	genetic optimizer with discrete design variable
KEAS	knots equivalent airspeed
MDA	multidisciplinary design analysis
MDAO	multidisciplinary design, analysis, and optimization
MDO	multidisciplinary design optimization
NASA	National Aeronautics and Space Administration
P_1, P_2	loading values
V-g	speed versus damping
V-ω	speed versus frequency
X_1, X_2, X_3	design variables

1 Introduction

Supporting the Aeronautics Research Mission Directorate [ARMD] guidelines, the National Aeronautics and Space Administration [NASA] Dryden Flight Research Center [DFRC] is developing a multidisciplinary design, analysis, and optimization [MDAO] tool. This tool will leverage existing tools and practices, and allow the easy integration and adoption of new state-of-the-art software.

Optimization has made its way into many mainstream applications. For example, MSC/ NASTRAN (MSC.Software Corporation, Santa Ana, California, USA) has developed solution sequence 200 for design optimization [1], and MATLAB (The MathWorks, Natick, Massachusetts, USA) has developed an optimization toolbox [2]. Other applications, such as the ZAERO (ZONA Technology Inc., Scottsdale, Arizona, USA) aeroelastic panel code [3] and the CFL3D Navier-Stokes solver [4] do not include a built-in optimizer.

Current commercial MDAO tools are codes that are limited to certain disciplines and are largely based on a single fidelity approach. Aircraft designs using nonlinear analyses such as transonic aeroelasticity are predominantly developed using the manual trial-and-error approach; this considerably slows the entire design process. The development of an MDAO tool to automate the existing manual trial-and-error approach will enable true multidisciplinary optimization early in the design process.

The primary and long-term objective of the development of the MDAO tool is to generate a 'central executive' capable of using disparate software packages in a cross-platform network environment so as to quickly perform optimization and design tasks in a cohesive streamlined manner. This object-oriented framework will integrate the analysis codes for multiple disciplines, instead of relying on one code to perform the analysis for all disciplines. Optimization can then take place within each individual tool, or in a loop between the executive and the tool, or both. Figure 1 shows a typical set of tools and their relation to the central executive.

2 Background and Analysis Modules

The heart of the MDAO framework is the central executive module. This is the module the user will use to choose input files, solution modules, and output files; to determine the status of executing jobs; and to select modules for the viewing and filtering of outputs. It is a graphical user interface [GUI] that provides a single point of control for applications that run on the user's own computer, or for code that may reside on remote workstations or a computational cluster. At this stage, most of the code is written in FORTRAN on a UNIX platform. The GUI will be added later in the development process.

An interesting aspect of the central executive application is user-selectable resolution. An engineer engaged in preliminary design will probably prefer quick analysis turnaround over absolute accuracy. In aeroelastic analysis, for example, this may mean using a simple strip theory. As the design effort proceeds, the designer may segue into using a more accurate aeroelastic panel code such as ZAERO, and then perhaps an unsteady Euler- or Navier-Stokes-type of solver. The interface and process, however, to the extent practicable, should be the same for each solution type. The user also can easily plug in his own analyzer and script. This could be considered a form of high-level object-oriented solution. Only the interface is exposed to the user. The interface variables can be total weight, safety factors, frequencies, drag, noise level, flutter speed, et cetera. Internally, one solution module could be entirely different from another, but the solution module is hidden from the user.

Two external optimizers are included in the MDAO tool: design optimization tools [DOTs] [5, 6] based on a gradient-based algorithm; and the genetic algorithm [GA] [7]. One reason to use an external optimizer is to use a different type of optimizer. NASTRAN, for example, uses a gradient-based approach to optimization [8]; a drawback to this approach is the necessity to compute finite difference or analytical sensitivity values and perform the search, and there is often a need for a prior input with experience in defining the problem and search directions.

The DOT is a commercial optimization code that can be used to solve a wide variety of nonlinear optimization problems. When the DOT requires the values of the objective and constraint functions corresponding to a proposed design, it returns control to the user's program. The user's program calls the DOT again to obtain the next design point; this process is repeated until the DOT returns a parameter to indicate that the optimum objective function is reached.

Genetic algorithms do not require gradient calculations and can be started with random seeds, eliminating some of the need for user input and allowing for solutions that may not be readily apparent to even experienced designers [9]. In the case of multiple local minima problems, genetic algorithms are able to find the global optimum results while gradient-based algorithms may converge to the local optimum value.

2.1 MDAO Framework

Currently, the central executive MDAO framework can handle structural optimization problems. The framework of the MDAO is object oriented; users can either use the built-in pre- and post-processor to convert design variables to structural parameters and generate objective functions or use their own analyzer for the optimization analysis. The MDAO framework process is presented in the flowchart in Fig. 2. The MDAO tool has been tested and validated in several test cases and applied to model update problems [10]. Basically, the current MDAO framework developed at NASA DFRC contains the five modules described immediately below.

The input module is used to read the user input data which includes model type, analysis type, constraints, responses, optimizer setting, et cetera.

The optimizer module is used to select any of the optimizers described in this paragraph, based on user input. Optimizers include a genetic algorithm [GA] with a continuous design variable [GC]; a gradient-based algorithm, that is, DOT, with a continuous design variable [DC]; a genetic algorithm with a discrete design variable [GD]; GC + DC; and GC + DC + GD. Additional optimizers can be added in this module if they are needed in the future.

The pre-processor module is used to create and update input files based on the design variable values before executing the analyzer module.

The analyzer module is used to execute analyzer and scripts automatically. Users can use a script file to execute a series of analyses in sequential order.

The post-processor module is used to post-process the analyzer output file and to analyze the results automatically. Since the output file format from NASTRAN is different than that from ZAERO or CFL3D, several sub-modules have been developed. More sub-modules will be added as needed.

2.2 Structural Analysis Discipline

The structural analysis discipline has been developed into the MDAO framework; other disciplines are planned to be included in this framework in the future. The structural analysis discipline determines the structural sizing required for a vehicle to be of sufficient strength to withstand the load conditions imposed during operation. The structural design of an airframe in MDAO is determined by multidisciplinary criteria which include stress, deflection, buckling, and margins of safety for sizing optimization under given loading conditions; flutter and divergence requirements; and weight. This is a complex effort, which involves loads analysis, structural dynamics, aeroelasticity, and structural optimization. The main outputs from the structural analysis discipline are structural weight, mass properties, safety factors, divergence speeds, flutter speeds, flutter frequencies, natural frequencies, and mode shapes.

Several sub-modules were developed in the post-processor module for the structural analysis discipline. These sub-modules automate data calculation, provide data output in useful formats, and provide information with which to compute the objective function and penalty function.

2.2.1 The Weight Calculation Sub-module

Weight calculation by this module is a straightforward process. MSC/NASTRAN with a weight generator is used to compute the model weight, moment of inertias, and location of the center of gravity [CG]. The weight will be saved as an objective function if defined so by the user.

2.2.2 The Deflection, Stress, Strain, and Buckling Sub-module

Deflection, stress, and strain are computed using the MSC/NASTRAN static analysis solution sequence 101. For overall linear buckling, the MSC/NASTRAN solution sequence 105 is used to calculate the eigenvalue associated with the first buckling mode for the structure. The critical buckling load can also be calculated for the structure.

2.2.3 The Modal Analysis Sub-module

Natural frequencies and mode shapes are computed using MSC/NASTRAN solution sequence 103. These data are generated and saved for the ZAERO flutter analysis.

2.2.4 The Flutter Sub-module

For higher accuracy, subsonic and supersonic flutter analyses are performed using the ZAERO code. Since the ZAERO code requires structure modal data, the modal analysis sub-module must execute first.

2.2.5 The Divergence Sub-module

Divergence analysis is performed using an in-house code based on eigenvalue problem. The theory is based on the Rayleigh-Ritz Method and Galerkin's Method [11]. Input data to this code is a generalized aerodynamic influence coefficient [AIC] matrix from the ZAERO code, and generalized stiffness matrix data from the MSC/NASTRAN code. The output is the divergence speed.

3 Applications

3.1 Three-Bar Truss Stress/Strain Analysis Model

The preliminary application of the MDAO tool was the optimal design of a three-bar truss subjected to an external load [8] as shown in Fig. 3. In this problem the objective is to minimize the total weight of the structure. In the figure, the design variables X_1 and X_2 correspond to the cross-sectional areas of member 1 (and 3) and member 2, respectively. The area of member 3 is "linked" to be the same as member 1 for symmetry. The constraints are tensile and compressive stress constraints in member 1 and member 2 under loading P_1 and P_2. The loading P_1 and P_2 are applied separately and the material specific weight is 0.1 lb/in^3. The allowable stress of the tension and compression in the member is 20,000 psi and -15,000 psi, respectively. This problem can be solved by using a finite element model and an analytical model with a closed-form solution. Table 1 shows the comparison of results between the closed-form solution, MSC/NASTRAN using solution 200, the current MDAO tool with the DOT optimizer, and the current MDAO tool with the GA optimizer. All of the results agree with the analytical results. The optimization history of the objective function using the GA optimizer is shown in Fig. 4.

3.2 Ikhana with the Fire Pod

3.2 1 Overview

The three-bar truss is a straightforward application. A more interesting application and one of interest to NASA DFRC is the analysis and optimization of aeroelastic surfaces. The NASA DFRC is NASA's premier flight research center and is often the location for flights of highly modified vehicles intended to explore highly unusual flight regimes. When a vehicle has been modified elsewhere, it is often Dryden's task to analyze its aeroelastic characteristics for flight safety validation, and the same is true when the aircraft is designed or modified 'in house.' Aircraft must be free from aeroelastic instabilities such as flutter and divergence to ensure safe operation.

Recently, NASA DFRC acquired a Predator B (General Atomics Aeronautical Systems, Inc., San Diego, California, USA) unmanned aircraft system for civilian missions. The Ikhana, as the vehicle is named, is shown in Figs. 5 and 6 and will carry a 'fire pod' that will transmit images of remote areas of the western United States down from the aircraft to a ground station. The fire pod is located under the wing, near the left wing root, and can alter the flutter characteristics of the baseline aircraft. The flutter flight envelope prediction of the Ikhana using the current MDAO design process is the second optimization problem of this study.

The objective of the second optimization problem is to maximize the flutter and divergence speeds of the structure by varying the chordwise location of the fire pod +/-30 in. chordwise from the baseline location while constraining the allowable tensile and compressive stresses within an acceptable range. The chordwise location of the fire pod will be the design variable for this application. A negative chordwise location means to move the fire pod forward from the baseline location.

Below are the steps of the MDAO flutter and divergence optimization for the Ikhana problem. Several structural analysis disciplines will execute in sequential order within the

MDAO analyzer module for this optimization problem.

Step 1: update the finite element model and the aerodynamic model based on the design variable in the pre-processor module.

Step 2: run the modal analysis using MSC/NASTRAN solution 103 to compute stiffness, mass, frequency and mode shape in the analyzer module.

Step 3: run the ZAERO trim analysis and MSC/NASTRAN solution 101 to compute the stresses in the analyzer module.

Step 4: run the ZAERO to compute flutter speed in the analyzer module.

Step 5: run the in-house divergence code to compute divergence speed in the analyzer module.

Step 6: run the post-processor to collect and compute flutter and divergence speeds and constraints.

Step 7: return the objective function, constraints, and penalty function to the optimizer.

Step 8: repeat from step 1 until the optimum result is reached.

3.2.2 Challenge

A flutter analysis determines the dynamic stability of an aeroelastic system. As with static aeroelastic analysis, flutter analysis presupposes a structural model, an aerodynamic model, and their interconnection by splines. Therefore, modification of the fire pod location affects both the structural finite element model and the unsteady aerodynamic model in this design optimization process. To achieve the true optimum result, new MSC/NASTRAN and ZAERO analyses must be executed for each optimization iteration with any design variable update. The challenge of the Ikhana example is in the size of the AIC matrices and the complexity of the problem. With current computing resources at NASA Dryden, and using the ZAERO code to generate the AIC matrix, computation for a single case, one Mach number with 16 reduced frequencies, takes an average of 20 hr. Even using an existing AIC matrix, it still takes approximately 25 min. to complete the flutter analysis. Using the GA optimizer requires at least hundreds and perhaps thousands of NASTRAN and ZAERO executions. This is not at all practical as a timely process.

3.2.3 Approach

As mentioned above, because of computing resource limitations and the excessive real time required for this large optimization problem, several approaches have been investigated to expedite the flutter and divergence calculations and to avoid computing a new AIC matrix for each design variable update.

The first approach computes an AIC matrix that corresponds to the desired fire pod location using the cubic spline from some precomputed AIC matrixes based on a predefined fire pod location that has been previously investigated. A matrix named ‘AJJ’ that contains the unsteady pressure coefficients [Cp] on each aerodynamic box for each Mach number and reduced frequency can be output from ZAERO. Matrix AJJ is the fundamental matrix for AIC matrix calculation. Since matrix AJJ is very large, it is very time-consuming to cubic-spline each element in the matrix. In addition, ZAERO does not allow users to modify or replace matrix AJJ. The ZAERO does allow the user to replace matrix QHH, which is the generalized aerodynamic forces matrix due to mode shapes. Since QHH is a generalized matrix, the QHH matrix is quite small compared to matrix AJJ. Although matrix QHH can be calculated from matrix AJJ [3], computing matrix QHH from matrix AJJ requires much computing memory and power. The first approach was not chosen for Ikhana problem due to constrained computing resources and excessive computational time.

The second approach is to perform the approximation based on the matrix QHH. The decision was made to cubic-spline each element in the generalized aerodynamic forces matrix QHH (instead of AJJ) in the most efficient and effective way. The matrix QHH calculation was carried out explicitly for seven fire pod chordwise locations with one Mach number and 16 reduced frequencies. These generalized (modal) aerodynamic force coefficient matrices were then interpolated to any new fire pod chordwise location for the flutter analysis. The flutter analysis is performed in modal coordinates. The analysis does not provide accurate estimates of the AIC for the desired fire pod location. Using this second approach, the time required for computing the divergence speed and the flutter speed calculation using ZAERO is still time-consuming, requiring 40 min. Because of the difficulty of using matrix AJJ or QHH, a third approach was developed and the decision was made to perform the approximation based on the third approach.

The third approach, since the technical goal in this research is to demonstrate the aeroelastic optimization problem, is to determine the interpolated flutter and divergence speeds from some precalculated flutter and divergence speeds computed from ZAERO and in-house codes based on some predefined fire pod locations without running NASTRAN, ZAERO, and in-house codes for each design variable update.

Thirteen ZAERO flutter and in-house divergence analyses with chordwise fire pod locations +/-30 in. with 5-in. increments from the baseline model were analyzed before optimization. The flutter and divergence speed are summarized and shown in Fig. 7 and Table 2 with case 7 being the baseline model. According to these results at thirteen chordwise pod locations, with 2% damping, in cases 9, 11 and 13, the first flutter speed decreased 57.7 knots equivalent airspeed [KEAS], 67.6 KEAS, and 173.1 KEAS while moving the fire pod 10 in. , 20 in. and 30 in. aft, respectively. In case 5, when the fire pod was moved 10 in. forward of the baseline, first the flutter mode became a hump mode then disappeared, divergence was the only instability. Based on Fig. 7 the divergence is critical for Ikhana with the fire pod between 25 and -30 in.. This approach does provide fair estimates of the critical divergence speed for the desired fire pod location. This third approach enables performing the optimization using both the gradient-based optimizer and genetic algorithm.

4 Results and Discussion

The Ikhana with fire pod optimization problem has been performed using both the gradient-based optimizer DOT and the genetic algorithm with the third approach.

Because of the complexity of this application and the presence of some local minima and maxima, the gradient-based optimizer works, although inefficiently, to reach the optimum solution for this problem. With different initial values, the optimizer can only reach different local maximum, not the global maximum. See Table 3 for the results of the final design variable and the objective value using DOT.

Multidisciplinary optimization of maximum flutter speed and divergence speed with no constraints using the genetic algorithm optimizer is exercised and the results have been presented in this paper. The fire pod chordwise location is limited to -20 and 30 in. When the optimization was carried out with the third approach and the genetic algorithm optimizer with 20 populations, the optimum fire pod location was 14.4 in. aft of the baseline model and the divergence speed was 465.53 KEAS. Although there is no significant changes in the value of divergence speed at each fire pod the locations, MDAO are able to find the optimum fire pod location in this exercise. Table 4 shows the comparison of the critical speed before and after optimization. The optimization result moved the V-g and V-ω curves from the baseline configuration in Figs. 8 and 9 to the optimized configuration in Figs. 10 and 11. Only modes with frequency below 20 Hz are plotted.

5 Conclusion

A MDAO tool using gradient-based optimizers and a genetic algorithm has been developed at the National Aeronautics and Space Administration Dryden Flight Research Center to automate the design and analysis process and leverage existing tools such as NASTRAN, ZAERO and CFD codes to enable true multidisciplinary optimization in the preliminary design stage of subsonic, transonic, supersonic and hypersonic aircraft. The first stage of implementing and applying MDAO techniques to the fire-pod-equipped Ikhana has been successfully completed at DFRC; the MDAO shows much promise for improving the Ikhana with fire pod optimization. Studies have also shown the potential of the MDAO tool for increasing the flutter margin of safety in flight test envelope expansion. The MDAO process produces a much-desired design for flight envelope expansion. Applying the MDAO tool in modern and realistic aircraft design is efficient; once all of the tools for pre- and post-processing were in place, all design conditions were taken into account.

The challenge of integrating higher-fidelity codes, such as ZAERO, with the MDAO tool is in the time required for computation. High fidelity would become practical if high-fidelity analysis executed more quickly. All of the individual analysis components, as well as parts of the integrated procedure, have been numerically tested, with computing resource upgrades to improve ZAERO computational time being carried out. Once the structural analysis discipline is finalized and integrated completely into the MDAO process, other disciplines such as aerodynamics and flight controls will be integrated.

References

[1] *MSC.NASTRAN 2005 quick reference guide*, MSC.Software Corporation, 2004.

[2] *Matlab 7 getting started guide*. The MathWorks, Inc., 2008.

[3] *ZAERO version 8.0 user's manual*, ZONA Technology, Inc., 2007.

[4] Krist S, Biedron R and Rumsey, C. "CFL3D User's Manual (Version 5.0)," NASA-TM-1998-208444, 1998.

[5] *DOT design optimization tools user's manual version 5.0*. Vanderplaats Research & Development, Inc., 2001.

[6] Vanderplaats G. *Numerical optimization techniques for engineering design*. McGraw-Hill Book Company, 1984.

[7] Charbonneau P. and Knapp B. *A user's guide to PIKAIA 1.0*. National Center for Atmospheric Research, 1995.

[8] Moore, G. *MSC/NASTRAN design sensitivity and optimization user's guide*. The MacNeal-Schwendler Corporation, 1993.

[9] Goldberg, D. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional, 1989.

[10] Lung, S and Pak C. Structural model tuning capability in an object-oriented multidisplinary design, analysis, and optimization tool. To be presented at the 26th Congress of the International Council of the Aeronautical Sciences (ICAS), Anchorage, Alaska, 2008.

[11] Rao, S. *The finite element method in engineering*, Pergamon Press, 1982.

Tables

Bar Area	Closed form solution	MSC/NASTRAN	MDAO with DOT	MDAO with GA
Bar 1	.788675	.77142	.78798	.72093
Bar 2	.408249	.45185	.40999	.66093
Bar 3	.788675	.77142	.78798	.72093
Total weight	2.63896	2.6338	2.6388	2.7003
Number of iterations	N/A	5	7	50 generation and 50 propulsion

Table 1. Results of optimization on a three-bar truss.

Case number	Fire pod location from baseline, in.	Divergence speed, KEAS	Flutter speed, KEAS	Flutter Frequency, Hz
1	-30	466.5	N/A	N/A
2	-25	465.2	N/A	N/A
3	-20	464.0	N/A	N/A
4	-15	463.2	N/A	N/A
5	-10	463.3	N/A	N/A
6	-5	463.7	558.4	14.6
7	0	464.2	554.2	14.7
8	5	464.5	522.0	14.6
9	10	465.1	496.5	13.8
10	15	466.5	465.6	13.3
11	20	465.1	486.6	12.2
12	25	464.7	456.0	10.8
13	30	464.7	381.1	10.4

Table 2. Ikhana with fire pod divergence and flutter speed.

Design variable Chordwise location from baseline, in.		Objective value Critical speed, KEAS
Initial value	Final value	
-20.0	-19.80	463.95
0.0	0.0001	464.20
10.0	14.31	465.53
20.0	14.27	465.53

Table 3. Ikhana with fire pod optimization using design optimization tools.

	Design variable Chordwise location from baseline, in.	Objective value Critical speed, KEAS
Before	0.00	464.20
After	14.37	465.53

Table 4. Summaries of critical speeds before and after optimization using genetic algorithm.

Figures

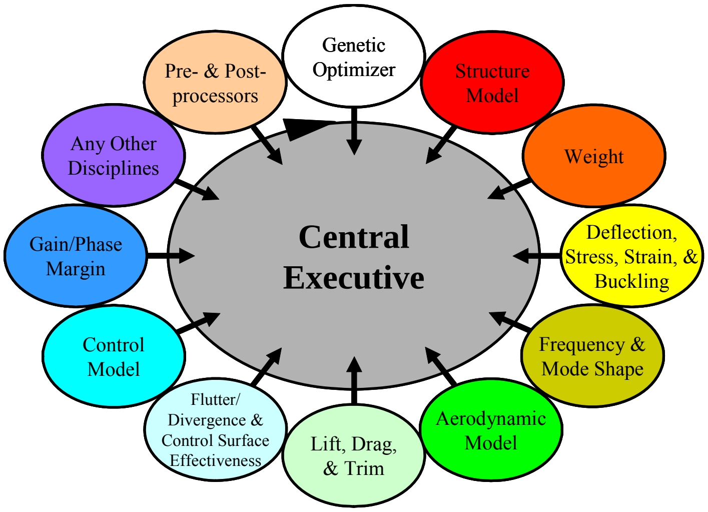

Fig. 1. Central executive overview.

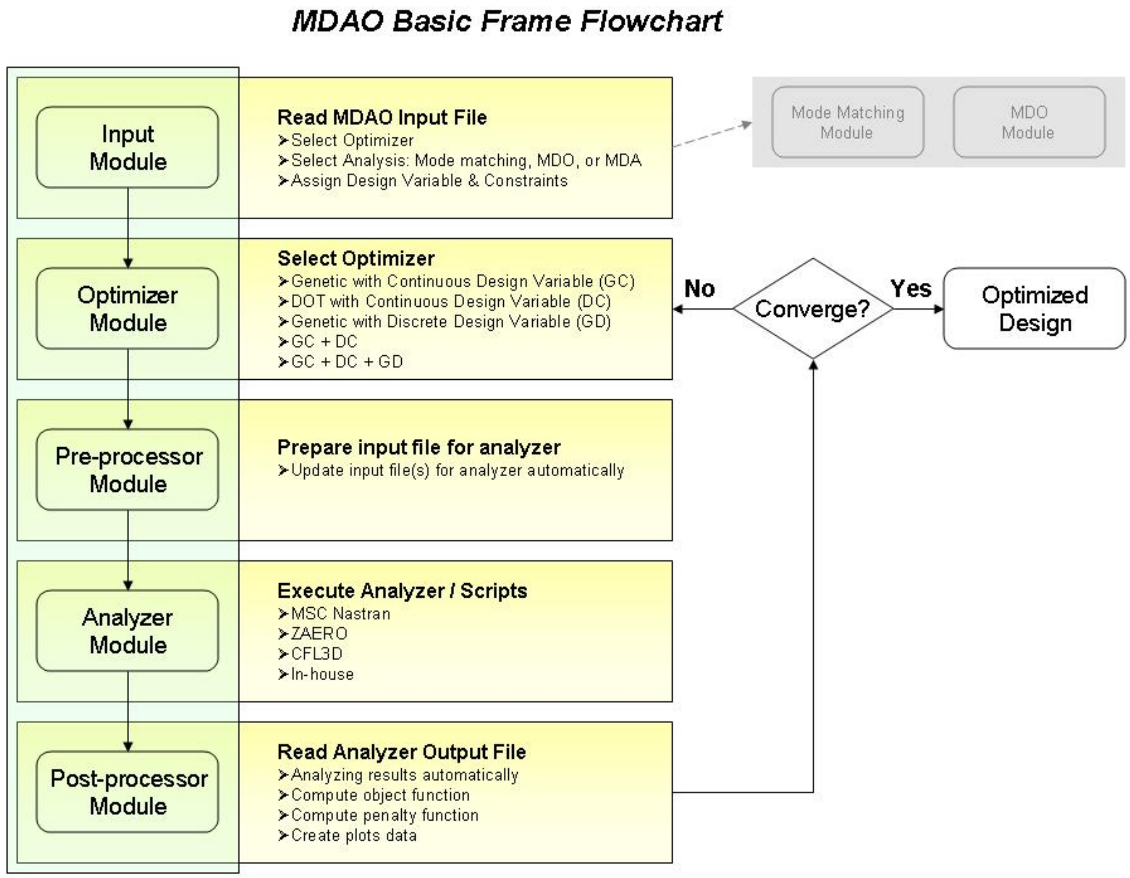

Fig. 2. The basic framework of the multidisciplinary design, analysis, and optimization tool central executive.

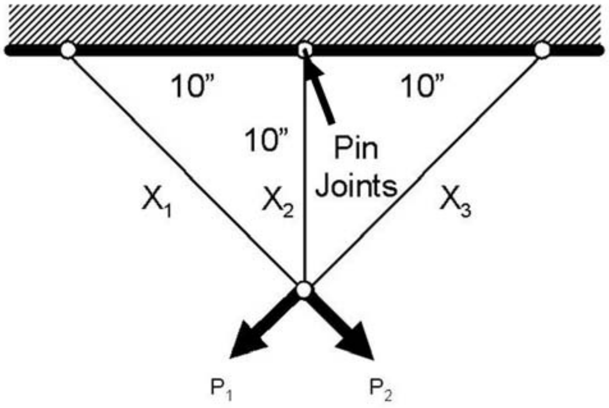

Fig. 3. Three-bar truss load conditions.

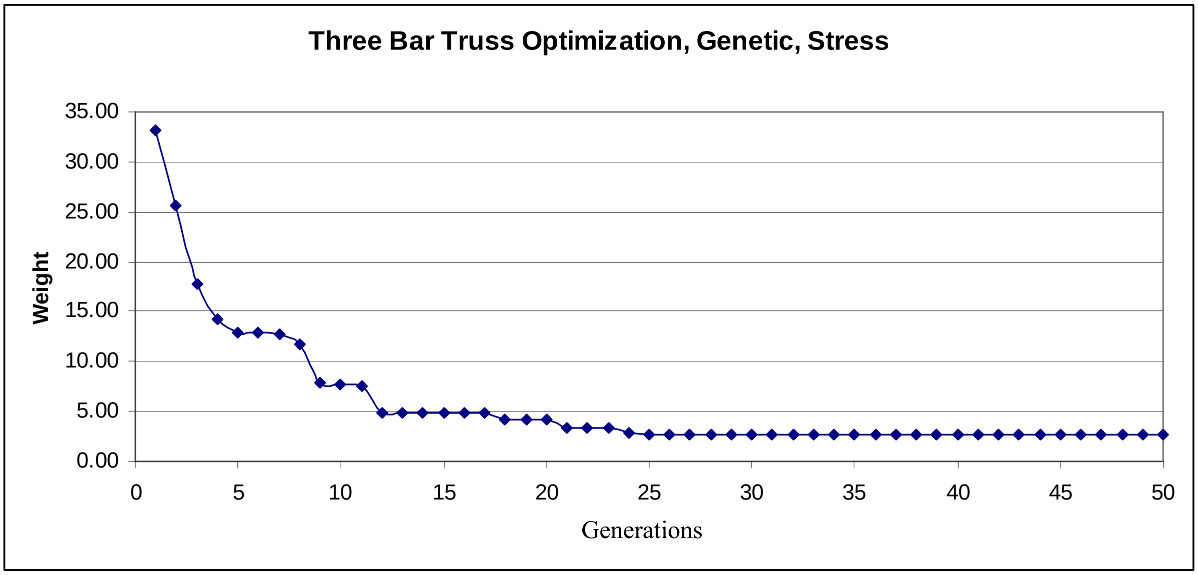

Fig. 4. History of design optimization of the three-bar truss: objective function weight and constraint stress using the genetic optimizer.

Fig. 5. The Ikhana baseline aircraft.

Fig. 6. The Ikhana aircraft carrying the fire pod.

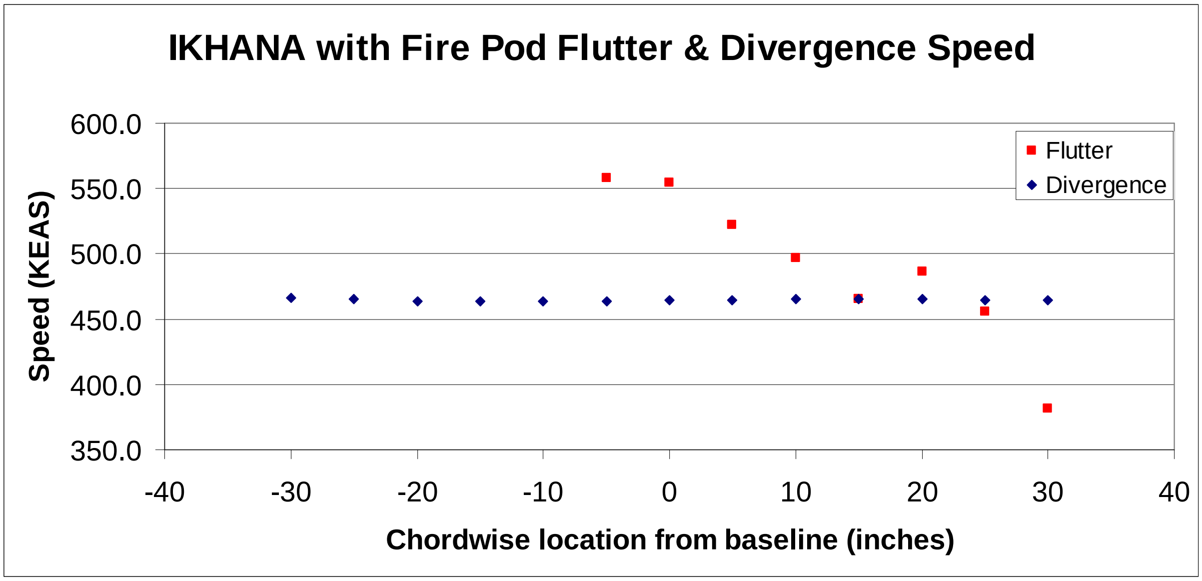

Fig. 7. Ikhana with fire pod divergence and flutter speed.

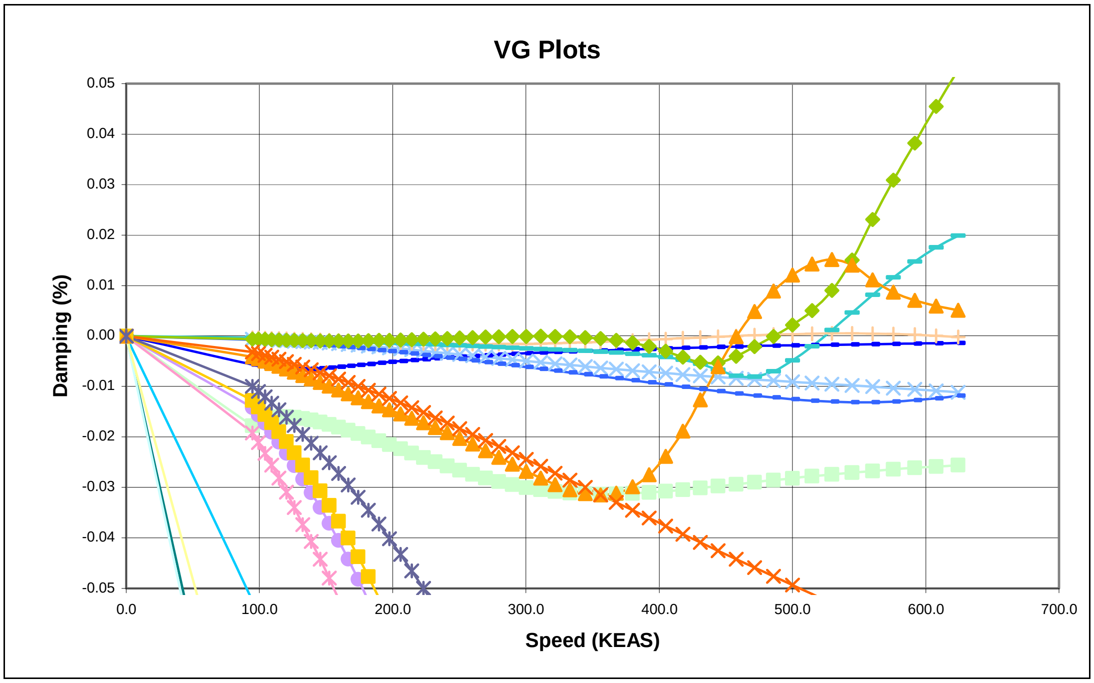

Fig. 8. The V-g curves before optimization.

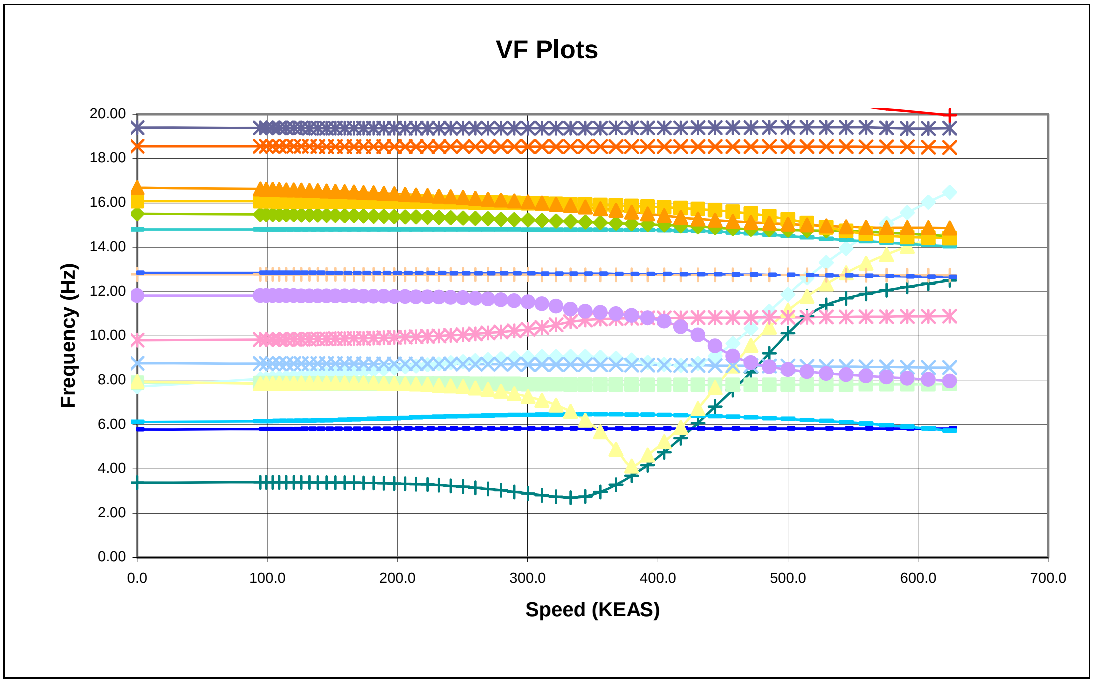

Fig. 9. The V-ω curves before optimization.

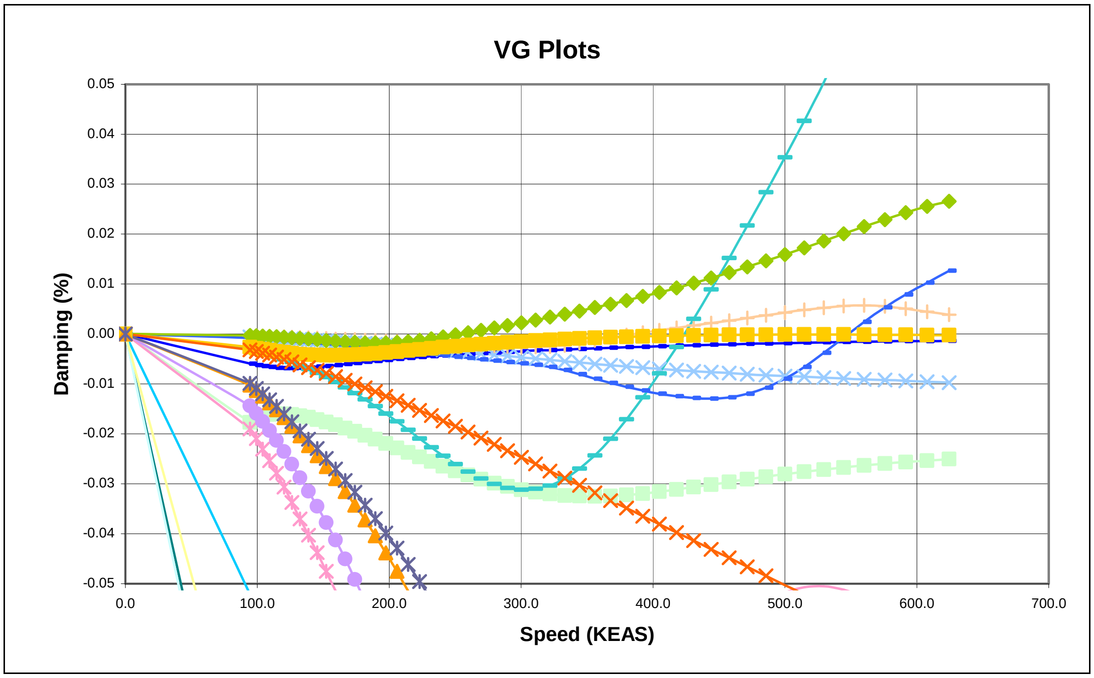

Fig. 10. The V-g curves after optimization.

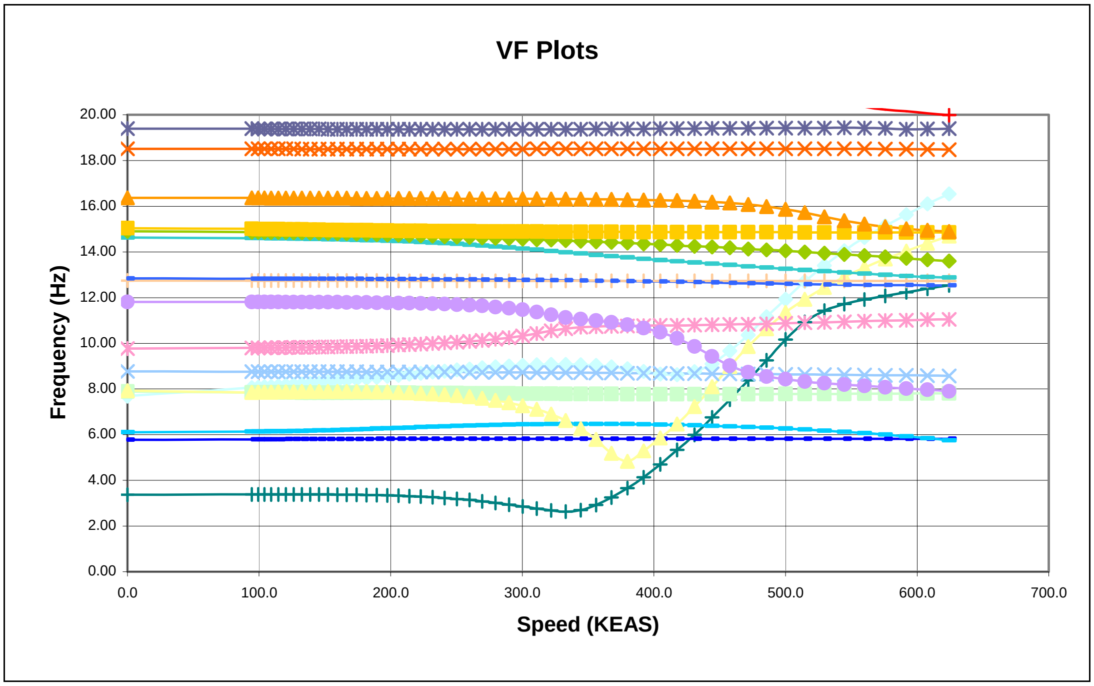

Fig. 11. The V-ω curves after optimization.

Copyright Statement