Abstract

Paper presents an overview of MADO, the Multidisciplinary Aircraft Design and Optimization software package. The package is prepared to link existing and modified tools to utilize them next, in an efficient way for optimization. The tools used in the first stage of development of MADO were presented. Examples of numerical computations are shown - airfoil and the new designed UAVs.

1 Introduction

Design of new aircraft is always a challenge. Today’s market demands highly efficient planes, fulfilling challenging missions and often contradicting expectations. Thus every project, especially as complex as new design of aircraft, is a set of compromises. Without meeting the goals from many scientific disciplines the aircraft simply won’t fly. If the design requirements are very demanding it is often hard to propose sufficient project not using numerical optimization techniques [1, 2]. More information on the early stage of the project helps to reduce costs and work time. This is done by improving the process of making confident decisions, which was considered in the SimSAC[3] project (Fig.1).

One of the main achievements of SimSAC project was development of CEASIOM[4] package, that links many different tools, from geometry definition, through aerodynamic computation, mass and inertia analysis, to stability and control issues. This system had to decrease time necessary to obtain wide knowledge about new designed aircraft. MADO goes a step beyond CEASIOM[5]. Collecting experience of SimSAC project authors present proposition to link current tools into global numerical optimization loop.

2 Main idea of MADO

Engineers have to deal with increasing amount of data, which forces the designers to focus on repeatable activities, not on creative work they are paid for. Most of these activities can be automated or semi automated depending on the complexity of the task and experience of the designer. Very often the disciplines (e.g. aerodynamics, stability and control, strength analysis, etc.) exist separately. The lack of interfaces between the tools, or only one way connections, cause that translating data from one solver to another con-
sumes most of the work time. Thus even simplest
optimization is not in use. Thanks to software
environment integration it is possible to improve
the design process. The key concept lies in con-
nection of main software analyzers (e.g. stabili-
ty analyzer, aerodynamic solver, etc.) into one
structure (Fig.2). The designer can make such
connections from one program to other manu-
ally, however it still could consume much time.
Depending on skills and experience user can add
scripts controlling data flow. The solution is de-
veloping easy to use, so-called wrappers, that link
tools used in design process into globally man-
aged system. Easy way of data transferring re-
lieves the designer and will allow to increase his
creativeness. Many starting points are possible to
realize the design task, either from CAD geometry
to numerical analysis, or in an opposite way.
Defining the complicated nonlinear geometry of
an aircraft can also be done by author’s innova-
tive interface by setting only few design param-
eters. This method can be treated as fast, high level
geometry definition. The package is thought to
extend capabilities of existing software by adding
additional components - commercial or free, used
in the actual institution and linking them into one
suite, which can be used for optimization.

2.1 Current development state of MADO

The collected software is capable of perform-
ing the aerodynamic, structural analyzes basing
on CAD geometry if desired. The package con-
tains the component for static and dynamic stabili-
ty analysis, which also allows to perform flight
simulation and estimate performance [6]. In this
way we can take into account many project de-
tails. All the unique features of presented pack-
age, which provide automation, allow to define
optimization task in a relatively easy way. Grad-
dient [7, 8], Monte Carlo [9], Genetic [10] and
Swarming [11] optimization algorithms are cur-
tently available in the suite. Designer works with
user friendly environment, spending less time on
repeatable activities, without sacrificing flexibil-
ity and complexity of the task he defines. Over
the time spent working with the software user
can increase complexity by adding next scientif-
cal disciplines and increasing number of design
variables. This procedure will lead to better opti-
mized solutions.

2.2 Main components of MADO

Current version of MADO contains several com-
ponents, that are the standalone applications.
However, they can also run in batch mode, which
is very useful to automate optimization process.
In the beginning the freeware (or ourself written)
applications were taken into account, due to pos-
sibility of making necessary changes to run iter-
ation without any prompting. Some commercial
applications, which were available for authors
were also initially tested with success (VSAERO,
VLAERO by AMI Inc.).

2.2.1 XFOIL

XFOIL [12] is a well known program for the de-
sign and analysis of subsonic isolated airfoils, de-
veloped by Mark Drela. Possibility to use XFOIL
in batch mode allowed for easy integration of the
software in the optimization loop.

2.2.2 PANUKL

PANUKL is a package for aerodynamic analysis
of an aircraft using low order panel method. The
package was born in mid nineties and was used
many times to compute aerodynamic characteris-
tics of an aircraft including stability derivatives.
The package (Fig.3) is being developed all the time, reaching its maturity. Current version [14] contains preprocessor to define geometry in an easy way using minimum necessary parameters, vortex wake generator, solver and post-processor to view and analyze the computation results. Although pure potential methods are not able to compute all components of drag, PANUKL can estimate induced drag using Trefftz method. The components of PANUKL package can be run in batch mode, what is useful to automate computations in optimization loop.

### 2.2.3 SDSA - Simulation and Dynamic Stability Analyzer

SDSA module was developed as the CEASIOM[5] module however it can be run as a standalone application as well. It was developed for S&C analysis and is able to compute stability characteristics using linear and nonlinear model (simulation model) as well [15].

SDSA uses the same Six DoF mathematical nonlinear model of the aircraft motion for all functions. For the eigenvalue analysis, the model is linearized numerically around the equilibrium (trim) point. Eigenvalues and eigenvectors analysis allow automatic recognition of the typical modes of motion and their parameters. The flight simulation module can be used to perform test flights and record flight parameters in real-time. The recorded data can be used for identification of the typical modes of motions and their parameters (period, damping coefficient, phase shift). The stability analysis results can be assessed basing on CS/FAR, ICAO, and MIL requirements.

As a module of MADO, it can receive all the necessary data (aerodynamics, mass, inertia),
when available, without special prompting. The necessary data can be delivered to SDSA as an XML file or as a set of plain text files. The second option is useful e.g. for experimental data. The data set contains aerodynamic coefficients or and stability derivatives tables, mass and inertia data, propulsion data, control derivatives and reference dimensions. The control and propulsion data can be completed and edited using special options of SDSA. SDSA accepts aerodynamic data as tables of stability derivatives as function of angle of attack and Mach number. SDSA also accepts a multidimensional array of force and moment coefficients versus six state parameters (angle of attack, Mach number, sideslip angle and rotational velocity components). A similar array is defined for control derivatives and stability derivatives versus selected accelerations (i.e. alpha dot derivatives). All aerodynamic data (derivatives) can be reviewed and are checked by comparison with typical values.

SDSA, running in batch mode, can deliver necessary output data for optimization procedure without any prompting and the optimization process managed by OptoM can run completely in an automatic way.

2.2.4 OptoM

OptoM (Fig.5) is meant to be easy to use tool for all purpose optimization. Core of the application has incorporated many different optimization algorithms, with possibility of adjusting their settings in graphical user interface. Graphical interface also provides some settings for output format, input and output files paths, flags to use constraints, optimization parameters check, fast update of configuration files, error messages and so on. One of the very useful features is "Tester" option for single objective function analyze, which helps to test and debug optimization task defined by the user. Currently available family of optimization algorithms are: Gradient [7, 8], Monte Carlo [9], Genetic [10] and Swarming [11] methods. All the optimization algorithms can act in significantly different manner depending on the settings. User of the application has to know only the basics of the optimization algorithms. The main work for him is to properly define the optimization task in a dynamically linked library,

![Diagram of SDSA and its functionality in optimization loop](image-url)
which is described in more detail in the next chapter. User is also equipped with tools to help him in the definition of the optimization task, like: definition of quadratic penalty function, matrix operations, functions calling external analysis software and many more.

2.2.5 Dynamic linked libraries

Optimization software is part which connects all the MADO tools in one comprehensive aircraft analyze. OptoM has flexible structure which is shown on Fig.6. User has majority of optimization algorithms he can choose from, which are incorporated in the main part of the OptoM program. All settings of the optimization algorithms are done in OptoM GUI. Change from one optimization method to another is done by single switch. Providing that the user has previously adjusted settings of the particular optimization algorithm, or agrees for the default settings, optimization process can start immediately with the new algorithm. Optimization task is defined in dynamic library which is linked with the OptoM. Such solution allows designer to focus on definition of his problem to solve without need of deep understanding of how the optimization algorithms work. Only basic knowledge about optimization is needed to adjust the settings if necessary. Ability to define directly optimization problem in the dynamic library provides flexibility and unbounded possibilities for the user. Optimization task can be entirely defined in the dynamic library, but nothing stands against using external analysis software. Any analysis software, which can take input parameters, scripts and commands can be incorporated Fig.6. Only minor difficulty is to provide appropriate input for the analysis software and read in results from the analysis during optimization.

3 Numerical examples

This chapter shows two examples of numerical optimization, which utilizes MADO concept. First example concerns of optimization of classical airfoil. The second example concerns MAV optimization.
In this example two dimensional airfoil is optimized. Objective function was maximization of lift coefficient for prescribed starting geometry and defined flow conditions. The optimized airfoil was NACA 23012, which had fixed angle of attack equal to 3 deg. The airfoil aerodynamic analyze was done by Xfoil code. It was assumed that the airfoil is optimized for small UAV and Reynolds number was set to Re=200000 and Mach number Ma=0. Design variables were: maximum thickness of the airfoil, maximum camber of the airfoil, position of the maximum thickness and position of the maximum camber. Changes of the geometry shape were easily made by using Xfoil tools for airfoil geometry modification. Optimization software always minimizes objective function, to maximize airfoil lift coefficient the objective function was mathematically defined as in equation (1).

\[ F_{\text{OBJ}} = 1/C_L \]  

(1)

For the optimization Monte Carlo optimization algorithm was used. The algorithm was set for ten iterations with fifty objective function analysis during every iteration. No additional constrains were defined. Convergence of the objective function (maximized lift coefficient) shows Fig.7. History of changing design parameters is shown on Fig.8. Initial and optimized geometry is shown on Fig.9.

After ten iterations airfoil lift coefficient for fixed angle of attack improved by \( dC_L \sim 0.1 \). In the case of the considered airfoil most influence on the objective function had parameters of maximum thickness and position of maximum camber and moderate influence of position of maximum thickness. Definition of the objective function and commands flow to make analysis with Xfoil software took less than one hundred lines of C++ code written in the OptoM shared library. Time of optimization was about fifteen minutes.
3.2 Bee - mini UAV

In the second example minimization of the aerodynamic drag was optimized for constant lift coefficient, with additional constrains put on equilibrium of forces in the aerodynamic $z$ axis and predefined static stability margin. Constrains were realized by quadratic penalty function. Achieved values of penalties from the crossed constrains were very big so the objective function was scaled by a factor of one hundred, which is defined by equation (2).

\[
F_{\text{objective}} = 100C_D + P_1 + P_2 \tag{2}
\]

where:

\[
P_1 = 0.5C_1^2/\mu
\]

\[
C_1 = mg - 0.5\rho V^2 SC_L
\]

\[
P_2 = 0.5C_2^2/\mu
\]

\[
C_2 = -0.1 - dC_m/dC_L
\]

Analysis were done utilizing non-viscous panel code PANUKL[14]. Design variables were: angle of attack, length of the tip chord, three parameters, which controlled nonlinear wing twist distribution defined as a fourth order polynomial center of gravity and four variables defining wing tip cut and filet in the middle part of the wing Fig.10.

![Fig. 10 Bee wing planform](image)

The first optimization results showed that the optimization task was over constrained. Solution satisfied stability constrains by varying angle of attack, but the geometry parameters didn’t change significantly maintaining the old geometry. After this experience variable of position of center of gravity was added. This way geometry could vary still satisfying the constrains.

For optimization genetic and gradient algorithms were used, both giving corresponding results. Genetic algorithm was very robust and always gave solutions. Contrary to that experience gradient algorithm needed quite some time to be set to start the solution to converge, but after it was done it converged much faster than genetic algorithm. The best type of gradient solver was second order Newton method. Although this algorithm needed second order derivatives, its estimation of search direction was so fine, that its efficiency overtook Steepest Descent, Conjugate Gradient and Quasi Newton gradient methods. Interestingly genetic algorithm, which can theoretically lead to random solutions, showed that two competitive solutions are possible, with the current optimization task definition, with completely different geometry (Fig.11).

![Fig. 11 Example of competitive optimization solutions](image)

The obtained solution, from the first optimization tests, revealed a question: are the configurations dynamically stable in all modes?

The basic tests of dynamic stability were performed and satisfying results were obtained for the first configuration, with the aft wing sweep (Fig.12). In the next step SDSA is going to run in
the optimization loop to satisfy the stability criteria "on-line". In the final version authors forecast, that SDSA will be able to modify the decision variables or be a part of constraints system.

For the final optimization, which included more design parameters and geometry details, Swarming optimization algorithm was used. This algorithm was set for ten iterations with fifty objective function analysis during every iteration. History of converging objective function is shown on Fig. 13. It shows how values of the objective function for the best and the worst individual in the swarm are getting closer. It can be seen, that already in the third iteration objective function, indicating best individual in swarm, merely changes. Performance of the optimization algorithm is very good, what can be observed on the Fig. 13 with logarithmic objective function axis.

Final geometry with very smooth pressure distribution after optimization shows Fig. 15. In the case of the MAV shared library was quite big, but most of the code defined highly nonlinear geometry of the flying object. Pure definition of the objective function and commands flow to execute external aerodynamic analysis took less than one hundred fifty lines of code. This example shows that defining optimization task takes less time, but ability to define the task in the shared dynamic library gives endless flexibility and allows for completion of very demanding tasks.

4 Concluded remarks and further steps

The experience collected so far in the first examples is very promising. Most of the current MADO tools have user friendly graphical interface with novel functionality, which makes it possible to define complicated problems for analyze with few variables. Great concern is put on easy data exchange between MADO tools, which allows to concentrate on the very design. Although software maintenance becomes very easy, experienced users will still be able to define complicated analyze and optimization tasks without limitations. MADO is meant to be so flexible, that its working capabilities will grow with the knowledge of the user, no matter on what stage zero, which means that both constrains are satisfied with great accuracy.
of the "know how" he is. In the optimization software any desired optimization task can be realized. The optimization task can be defined entirely in the shared library, or incorporate any external analysis software available with batch mode capabilities. Length of the shared library code, with analysis, optimization flow control, objective function, constrains and so on, in most cases will not even exceed 200 lines. Compiling only the shared library, containing only the optimization task and user equipped with additional features to help him define the task enables for very effective work. Future of the MADO package seems to be bright. The software is constantly developed and improved to make it even more computationally powerful, intuitive and efficient to work with.

References


Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS2012 proceedings or as individual off-prints from the proceedings.