

MULTI-OBJECTIVE OPTIMISATION OF AUTOMOTIVE COMPONENTS

Alberto Clarich, Mauro Poian, Carlo Poloni

ESTECO srl,
Area Science Park, Padriciano 99, 34012 Trieste, Italy
clarich@esteco.it (Alberto Clarich)

Abstract

In this paper the multi-objective design environment modeFRONTIER is applied to the design of automotive components.

This approach allows the integration of different computational codes, including parametric CAD systems like SolidWorks, FEM codes like Abaqus and Femfat fatigue tool, into a common design environment.

In this environment, the complete logic flow from CAD parameterisation to performances evaluation (in terms of safety factors, maximum stresses, global mass, etc.) is defined by the user, have to select the optimisation algorithms accordingly to the defined objectives; these algorithms drive automatic series of simulations, allowing also distributed and parallel computations to fully exploit the available computational resources, until the objectives are met.

At the end of the multi-objective optimisation, the designer can select the definitive solution among several alternatives as the best compromise among different and often contrasting criteria. The influence of all the parameters in the process can be analysed in detail by the use of statistical analysis and response surface methods.

In the paper, after a first simpler application to the design of a connecting rod, that is used to describe the CAD parameterisation techniques and the Abaqus and Femfat integration with modeFrontier, a more complex application to modal and harmonic analysis of a crankshaft is presented, to show the efficiency of the optimisation methodology.

Keywords: multi-objective optimization, automotive, fatigue analysis, harmonic analysis, CAD parameterization, distributed and automatic computational environment.

Presenting Author's biography

Alberto Clarich. Date of birth 11/02/75, Trieste. Msc degree in Mechanical Engineering, University of Trieste, February 2000. PhD degree in "Innovative Parameterisation and Optimisation Methodologies in Aeronautic Field", University of Trieste, March 2003, in collaboration with Dassault Aviation. Presently, he is working at ESTECO srl on multi-objective optimization and robust design with modeFRONTIER, collaborating with companies like Toyota, Ferrari, Honda, BMW, Motorola. He has published more than 30 papers in journals and conference proceedings



1 Introduction

Multi-objective optimisation in automatic and distributed environment, that allows direct communication between parametric CAD and simulation software, is becoming continuously a key factor in automotive industries and not only.

This paper illustrates how the optimisation environment modeFRONTIER [1] allows to practically implement this methodology, whatever software is used in the simulation phase.

Two application cases are presented: first, a connecting rod is to be optimised in order to find the minimum mass that guarantees an acceptable safety factor. Several CAD/CAE tools are required for this purpose: a CAD is used to modify the connecting rod shape through some chosen geometric parameters, a FEM structural analysis code is necessary to compute stresses, and finally a fatigue tool is used to quantify the safety factors.

The traditional design approach ('trial and error') would require many attempts from the designers, that each time need to modify CAD by hand, run several solvers and are not supposed to know in which direction of the variables space they need to move in order to find the best solutions.

Conversely, the multi-objective design environment modeFRONTIER integrates the different computational codes into a common design environment, allowing the automatic run of a series of computations, until the specified objectives are satisfied.

The full efficiency of the modeFRONTIER software is then proved by the second application shown in this paper, that is relative to the shape optimisation of an engine crankshaft, for which the safety analysis is performed by the modal and harmonic analysis approach.

2 modeFRONTIER multi-objective design environment

modeFRONTIER is a multi-objective design environment software that allows the integration of any commercial or built-in house computational code (CAD, FEM, CFD, Matlab, etc..) into a common environment, in order to run automatically a series of designs, proposed by the available optimisation algorithms, until the defined objectives are satisfied.

In this modular environment, each component of the optimisation, including input variables, input files, scripts to run the commercial softwares, output files, output variables and objectives, is defined as a node to be connected to the other components.

In this way, the complete logic flow from CAD parameterisation to performances evaluation is defined by the user, that can select among several available optimisation algorithms, accordingly to the defined objectives; they include Genetic Algorithms [2],

Evolutionary Algorithms, Game Strategies [3], Gradient-based Methodologies, Response Surfaces and Robust Design Optimisation, as well as main DOE (Design Of Experiments) algorithms (Sobol, Factorials, Latin Square, Montecarlo, D-Optimal, etc.).

These algorithms drive automatic series of simulations, allowing when available distributed and parallel computations to fully exploit the computational resources, until the objectives are met.

In addition, the influence of all the parameters in the process can be analysed in detail by the use of statistical analysis (correlation matrix, t-Student, etc.) and response surface methods (Kriging, Neural Networks, Radial Basis Functions, SVD, Parametric, Gaussian, etc.), that can be also used to reduce the number of computations required in the optimisation, allowing an extrapolation of the results.

3 Case study 1: optimization of a connecting rod

3.1 Introduction

In this first application, the purpose is to optimise the shape of a connecting rod, whose engine main data are reported in table 1, and derived by the data of Pravardhan Msc thesis [6].

The objective is to reduce as much as possible the mass, guaranteeing at the same time acceptable value of Endurance Safety Factor ($SF > 1$). The latter is in fact defined as ratio between the ideal stress that would give endurance for an infinite number of cycles and the actual stress, produced by the operating conditions of the connecting rod.

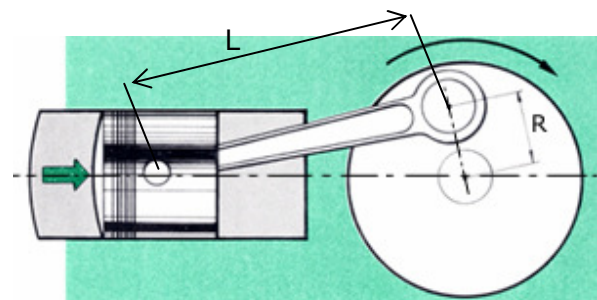


Fig. 1 Connecting rod

Tab. 1 Engine data

Engine Power	50KW@2000rpm
Cylinders nr	4
Crank R	32mm
rod L	150mm
Piston diameter D	85mm

3.2 Connecting rod parameterisation

The model is fully described in the SolidWorks [4] CAD, that allows to define some geometric

dimensions, like point co-ordinates, thickness and fillet radius, as parameters. The parameters can be modified in batch mode by the use of a VisualBasic script (that can be interfaced as the macro of an Excel sheet), with the effect of an automatic update of the CAD model.

Figure 2 shows the CAD model, with the dimensions relative to the 9 selected parameters highlighted.

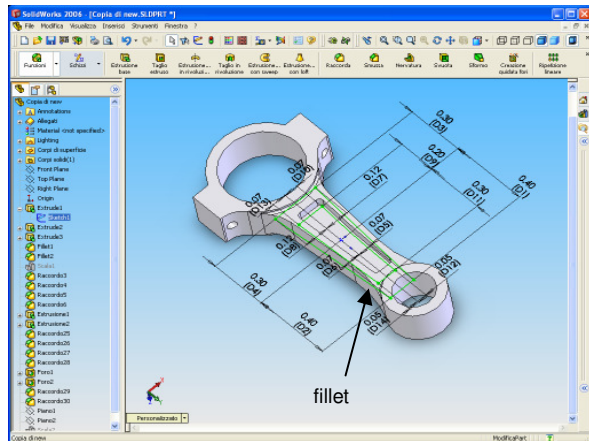


Fig. 2 Solidworks model parameters

In particular, 6 parameters control the abscissa and ordinate of the points in the sketch (that define the width of the rod and of the central hole), another parameter control the fillets radius and two other parameters controls respectively the thickness of the rod and the depth of the central hole.

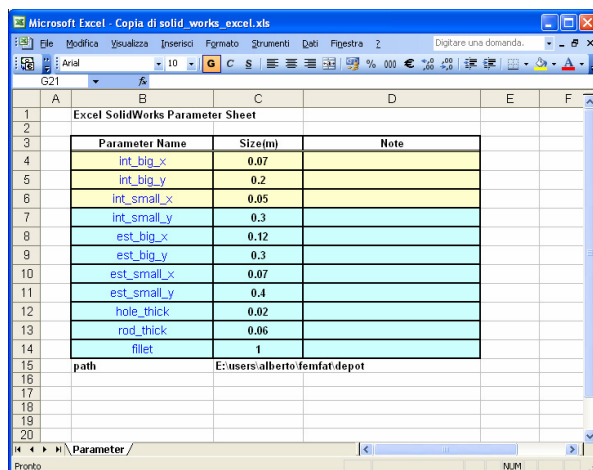


Fig. 3 Solidworks model parameters

Figure 3 above shows how an Excel sheet is used to indicate, in the C column, the values relative to all the 9 parameters, whose name is defined in column B. An internal VisualBasic macro simply reads each value of column C, and updates the corresponding parameters of the SolidWorks model; the last command of the macro forces SolidWorks to save in Parasolid format the updated model.

The modularity of this VisualBasic macro allows to extend this application to any CAD model, it is just

necessary to modify parameters name and values, and model path and name (row 15 in fig.3).

3.3 Structural analysis and fatigue analysis

The updated model, in Parasolid format, is used by the FEM structural software Abaqus [5] to perform a static analysis, giving as forcing load the maximum axial peak, as it is defined by the forces scheme of figure 4 [6].

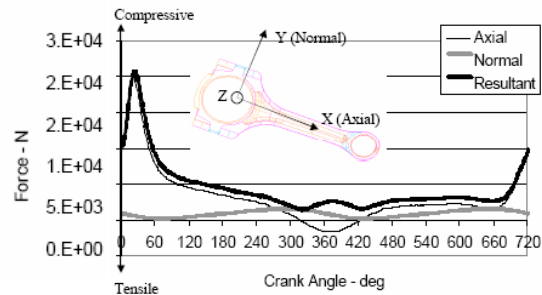


Fig. 4 Forcing loads on the connecting rod

In the figure, both normal and axial forcing components acting on the connecting rod, as well as their resultant, are reported versus crank angle. The data are relative to a crank speed of 2000 rev/min.

The Basic module of the fatigue analysis software Femfat [7], however, requires the specification of only one forcing channel (component), to be defined as a loading cycle, for which maximum and minimum stresses have to be specified.

From fig.4, we can disregard the normal component, and consider only the loading cycle constituted by the application of the axial (compressive) force.

By approximation, the upper stress of the cycle can be defined equal to zero, and the lower stress (negative) is obtained by the application of the maximum compressive peak load (about 20,000N), after an Abaqus static analysis.

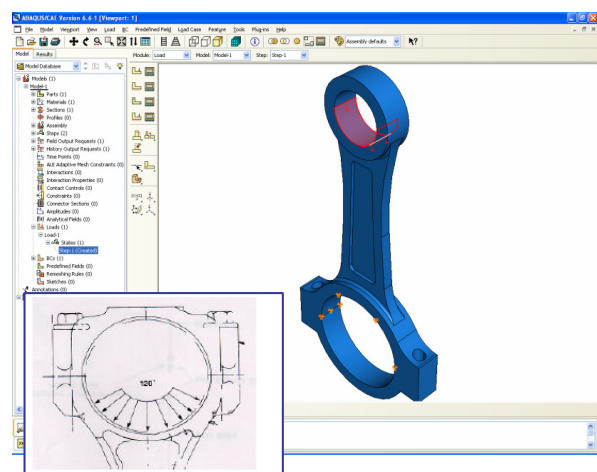


Fig. 5 Abaqus model and boundary conditions

In particular, fig.5 illustrates the Abaqus model with the forcing load and constraints applied, accordingly to the theoretical distribution of forces along an arc of 120°[6].

The *.odb* binary output file obtained by Abaqus analysis, is as discussed above used by the Femfat software to compute the safety factors of the connecting rod, giving as result an ASCII output file named as *.pro*.

In the course of definition of both Abaqus and Femfat models, macro of commands are automatically recorded in two ASCII files, named respectively as *abaqus.py* and *femfat.ffj*.

These two ASCII files play an important role in the automation of the optimization process, and will be used as template for the definition of every configuration proposed by the optimization software modeFRONTIER, as will be described in the next section.

4 Application of modeFRONTIER for conrod multi-objective optimization

4.1 Definition of optimization workflow

Figure 6 shows the modular workflow built up in modeFRONTIER, in order to drive the automatic connection between the CAD/CAE tools used in the optimization, and to specify the parameters of the optimization itself.

At the top there are the icons relative to the input parameters, for each of them the range of possible variation is specified.

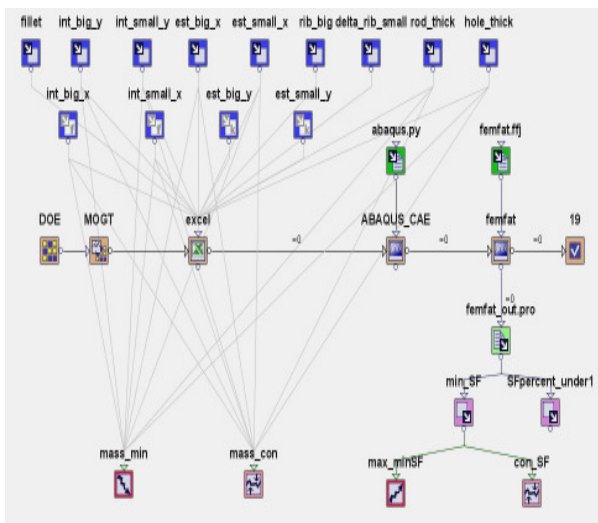


Fig. 6 modeFRONTIER optimization workflow

As discussed in sub-section 3.1, the input parameters enters directly in the Excel node (a different cell of the Excel sheet is assigned to each of them), and, as output, the updated model is passed directly to the following application, Abaqus node.

This node run a batch command to launch Abaqus pre-processor and solver, giving as input the macro of commands *abaqus.py*, that, as stated in sub-section 3.3, was recorded in the definition of the original Abaqus model and is then used here as template, to run each new configuration proposed by the optimization algorithm.

This is possible because, for each new configuration, all the operations from CAD model reading, mesh creation, boundary definition, to analysis run, are exactly the same, and what indeed changes is just the updated CAD model.

The following node runs in a similar way the Femfat solver, giving as input the *.odb* stresses file obtained from Abaqus and the macro of commands *femfat.ffj*, previously recorded in the same way as for Abaqus.

The final output node is the output ASCII file *femfat_out.pro*, from which the output variables to which we are interested, are extracted as shown in figure 7 below.

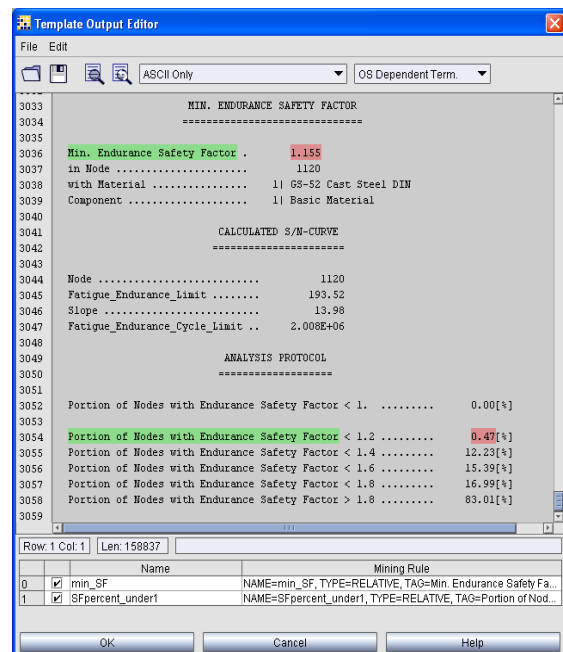


Fig. 7 Parsing of output variables in modeFRONTIER through the output file template

The output node in modeFRONTIER allows to open an existing template of the output file coming from the software, and to indicate, by a relative tag, which are the output variables to extract for each configuration proposed during the optimization.

In this case, we search for the tag 'Min. Endurance Safety Factor', and for the tag 'Portion of Nodes with Endurance Safety Factor < 1.2', obtaining the corresponding values, as is illustrated in figure 7.

The two data will be used in the optimization workflow as constraints: we need the minimum Safety Factor to be higher than 1.0, and the Portion of Nodes with Safety Factor less than 1.2, to be less than 1%, in order to be better guaranteed by the safety conditions.

The objective of the optimization becomes in this case the minimization of the conrod mass (defined as output in the output file and parsed in the same way here defined).

4.2 Selection of optimization algorithm and optimization results

Several Optimisation Algorithms are available in modeFRONTIER, including Genetic Algorithms (MOGAI, NSGAI), Evolutionary Algorithms, MOGT (Multi Objective Game Theory: it is an algorithm based on Nash equilibrium of Competitive Games Strategies [3]), Gradient-based Methodologies, Response Surfaces and Robust Design Optimisation, etc., and the designer can choose among them accordingly to the particular case as necessary.

In addition, in order to better initialise the optimisation, the user can choose among different DOE (Design Of Experiments) algorithms, including SOBOL, Factorial Methodologies, Montecarlo, etc.

In this particular case, since the objective of stress reduction is substituted by a couple of constraints, and only mass reduction is defined as optimisation objective, the mono-objective Downhill SIMPLEX [8] algorithm can be used, for its efficiency and rapidity of convergence. Genetic Algorithms are in fact more indicated when there are more than 1 objective, while gradient based algorithms are not to be used for a global search, because they are accurate but not robust if used starting from random designs as DOE.

The SIMPLEX algorithm just requires 10 random designs as starting DOE (1 plus the number of variables, 9 in this case), and at each iteration a new design is proposed, improving the objective and trying to respect the constraints proposed, until a convergence is found. This is obtained when the objective function improves less than a defined limit with respect to the previous iteration, or when a defined number of iterations is reached.

Figure 8 shows the results of the optimization: each bubble represents a different configuration obtained during the optimization, indicating in abscissa the value of the minimum Safety Factor and in ordinate the value of the mass.

A total of about 70 designs were needed to reach the algorithm convergence (fixed to a factor of $1e-4$ for the objective relative variation with respect to the previous iteration), and since each design has required few minutes for the different software evaluations, less than 1 day of automatic runs was required to complete the optimization and find the best solution (in fig.8, best design number 69 is highlighted and compared to the original design, 0). Table 2 reports the comparison of the two configurations relatively to objectives.

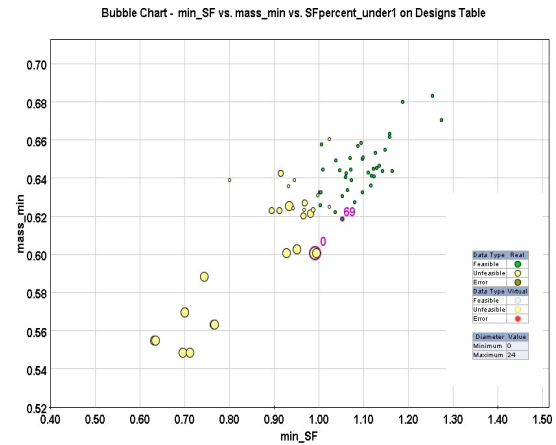


Fig. 8 modeFRONTIER bubble chart: yellow points violate constraints, green ones respect it; bubble size indicate value of second constraint (portion of SF<1.2)

Tab. 2 Comparison between original and best configurations

	Original	Best
Mass (kg)	0.60	0.62
min SF	0.99	1.05
SF%<1.2	24	0.2

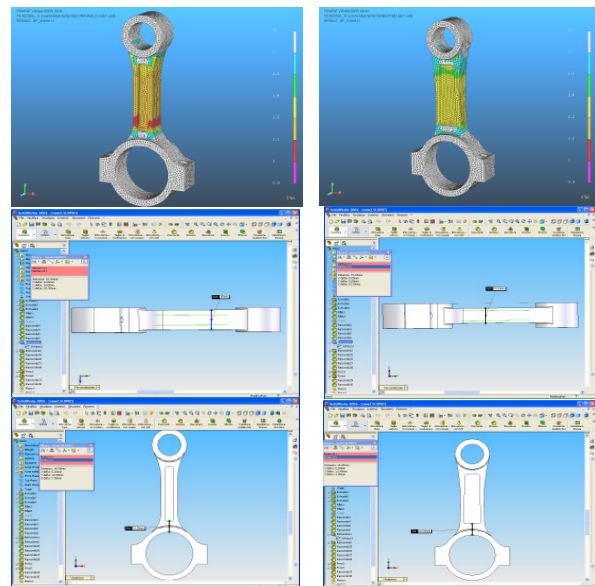


Fig. 9 Original configuration (left column) and optimized one (right column): safety factors and geometry

Figure 9 illustrates the optimized configuration (right) compared to the original one (left): not only the minimum safety factor is higher, but the region where SF>1.2 covers quite all the conrod, meaning that the security is higher (a value of SF too close to 1.0, could be not so satisfactory, because few fluctuations from the numerical model could give not expected ruptures after a finite number of cycles; a safety factor above

1.2 has been supposed to give more guarantees). This has been found by modeFRONTIER without varying much the global mass, resulting in a better distribution of masses than the original one, raising the width but reducing the thickness of the rod.

5 Case study 2: optimization of an engine crankshaft

5.1 Introduction

In this second application, the purpose is to optimize the web shape of a 4-stroke, 4 cylinders engine crankshaft, whose power data are the same of table 1. Also in this case, the optimization objective become the mass minimization, with the addition of two constraints relative to Endurance Safety Factor: the minimum SF have to be higher than 1.0, and the volume portion of the crankshaft where SF is less than 1.2, have to be less than 1%, to avoid large portions where SF is close to 1.0, and thus to increase the safety conditions

5.2 Crankshaft parameterisation

The crankshaft model is defined also in this case in the SolidWorks CAD, and the parameters to modify in this case are relative to the web shape only.

In particular, the sketch of the web is constituted by 5 blocks (fig.10) joined by fillets, and the abscissas and ordinates of the corner points of these blocks are defined in Solidworks as parameters (a total of 9), that can change resulting in a variation of the web shape.

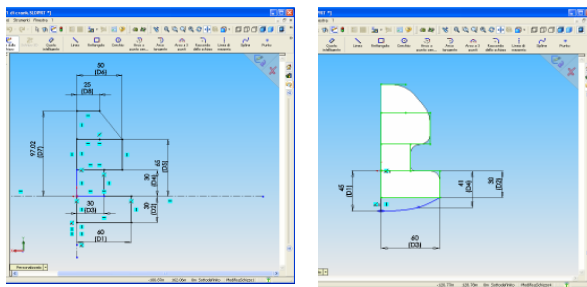


Fig. 10 Parameterisation of crankshaft web in SolidWorks

The parameterization of the CAD model is realized, like for the first application, through the use of a VB macro, that updates the CAD parameters reading the values on an Excel sheet, that is updated by modeFRONTIER for each design proposed during the optimization (see section 3.1).

5.3 Modal, harmonic and fatigue analysis

In this case, we need to apply the MAX (Multi-Axiality) module of Femfat to perform the fatigue analysis, because the forcing cycles acting on 4 crank-pins with different phases, do not allow the

approximation assumptions of the conrod model (1 single forcing cycle); MAX module of Femfat allows just the definition of multiple forcing cycles.

In fact, there are 4 crank-pins and 5 crank-bearings on which are applied forcing loads not in phase, and the best suited approach in this case is the modal-multi channel analysis [7].

First of all, a modal analysis of the unconstrained crankshaft is performed, in order to find the first natural vibration modes [9].

The analysis is made in Abaqus: the model is defined without constraints, and the results of the analysis is the frequency of each natural mode (in fig.11 only first two modes are illustrated), and the corresponding deformation/stress fields.

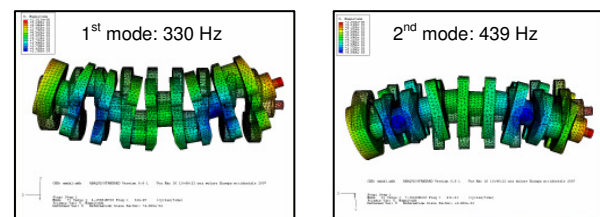


Fig. 11 Modal analysis in Abaqus

Second, a FFT (Fast Fourier Transform) of each forcing load on pins and bearings is performed, in order to define an amplitude vs frequencies plot of the harmonic components of the forcing loads (fig.12).

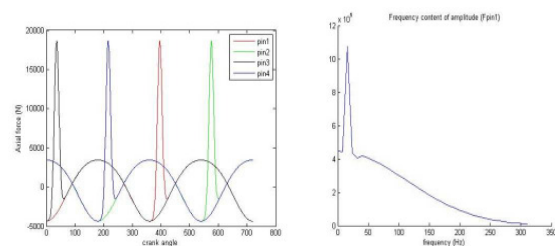


Fig. 12 FFT (right) of pin forcing loads (left)

In fact, each signal in the time domain (pin forcing loads, figure 12 left), can be decomposed in a sum of infinite harmonic components, each one of them characterized by a frequency, an amplitude and a phase (fig.13, right, amplitude vs frequencies).

The FFT analysis is necessary to find the amplitude of each harmonic component of the forcing loads, corresponding to its frequency. Also the phase of the forcing loads can be obtained by the FFT, but in this case, since the forces on the 4 pins are equally phased, the phase of the harmonic components presents difference of half π , following the ignition order of the pistons (fig.13, vector indicates the different phase of the harmonic components on the 4 pins).

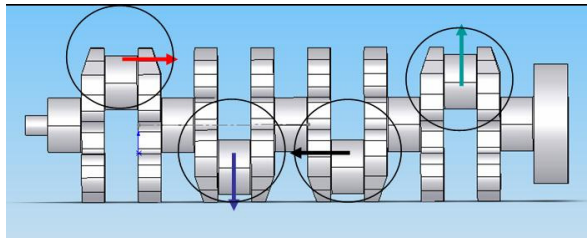


Fig. 13 Phase of the load harmonic components on the 4 pins

We still need to define the harmonic components of the reaction loads on the bearings. As approximation, we consider that each pin load is balanced equally by the two closest bearings, therefore each bearing is loaded by the sum of two half pin loads, except of the first and last that present only one component (fig.14)

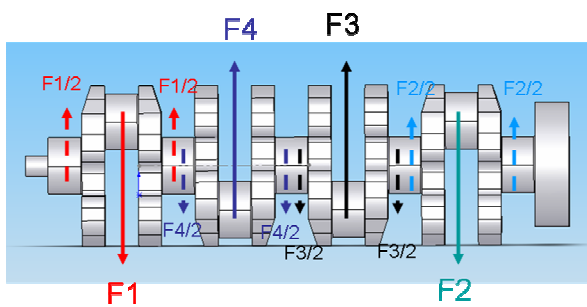


Fig. 14 Reaction load on bearings

In analogy, also the harmonic components of these reaction bearing loads are obtained by the sum of the relative harmonic components of pin loading forces, and therefore all the loads on the crankshaft can be now expressed in terms of harmonic components, including phase and amplitude for each harmonic frequency.

From vibration mechanics we know that only the harmonic components corresponding to the natural frequency of a system can produce a large amplification effect of deformations/stresses (resonance), and therefore, the harmonic analysis will be performed on the crankshaft applying only the harmonic components corresponding to the first natural frequencies.

All the loads, forcing and reactive, are applied on the pins and bearings specifying amplitude and phase corresponding to the selected frequency, and as result the stress field is obtained (figure 15, harmonic analysis in Abaqus). A structural damping coefficient equal to 0.03 is considered.

The harmonic analysis is repeated for the second natural frequency, but the stresses found are less than 1/10th of the first frequency ones: therefore, in the fatigue analysis, only one channel corresponding to the first natural mode is considered.

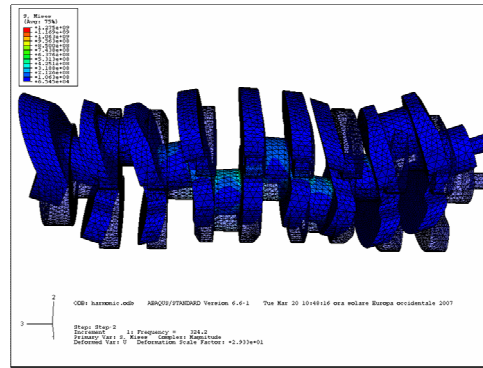


Fig. 15 Harmonic analysis in Abaqus (for frequency equal to 330Hz, 1st natural mode)

Femfat MAX module [7], in fact, allows to specify several channels, in each of one stresses coming from different harmonic components can be specified. In our case, however, to reduce the global computation time, only the stress coming from the first harmonic component are considered.

6 Application of modeFRONTIER for crankshaft multi-objective optimization

6.1 Definition of optimization workflow

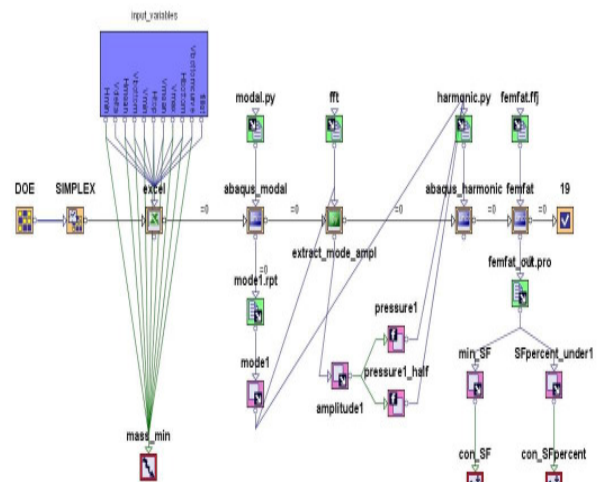


Fig. 16 Optimisation workflow in modeFRONTIER

Also in this case, the link between all the softwares used is defined in the main modular workflow of modeFRONTIER.

All input variables entering in Excel node are collected into a subsystem for simplicity, then there are 4 following application nodes in the main central chain.

Abaqus_Modal script runs, through the recorded macro of commands *modal.py*, the modal analysis, and from the output *model1.rpt* ASCII file, the first natural frequency is parsed and assigned to the output

variable *model*. This variable is used by the following application node, “*Extract_mode_ampl*”, that is a Python list of commands that reads the FFT data corresponding to the forcing load (fig.12), and extract the amplitude corresponding to the frequency defined by the variable *model* (*amplitude1* is the corresponding variable name).

At this point, the variable *amplitude1* is given to the following application, *Abaqus_harmonic*, that runs the harmonic analysis following the instructions of the originally recorded macro *harmonic.ffj*, updated by the new values of frequency and amplitude for the harmonic components.

The last part of the workflow follows the same steps of the conrod application: femfat run and extraction of outputs from ASCII output file.

6.2 Selection of optimization algorithm and optimization results

The same algorithm and optimization strategy is applied in this case, and the convergence of the solution is obtained after just 60 designs, that corresponds to about two days and half, considering that the complete simulation of 1 design takes about 1 hour on a double CPU Windows Machine (2Gb, 2.13GHz).

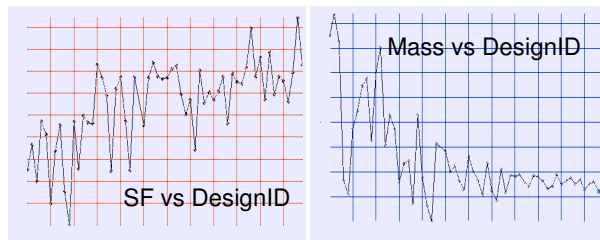


Fig. 17 Convergence of SIMPLEX after 60 designs

Fig.17 indicates the values of the two objectives considered, SafetyFactor on the left and mass on the right, versus each design proposed in the course of the optimization (a total of 60). As it is possible to see, modeFRONTIER improves design after design both the two criteria, the SF is raised and at the same time the mass is reduced.

Tab. 3 Comparison between original and best configurations

	Original	Best
Mass (kg)	21.7	18.5
min SF	0.33	1.025
SF%<1.2	4.9	0.4

Table 3 shows the comparison between the original crankshaft with the optimized one, and it is clear how both mass and safety factor criteria are satisfied. The optimized crankshaft present SF higher than 1.0 in all the parts as well as the portion of nodes with SF lower than 1.2 are almost disappeared (fig.18 below), and at the same time the global mass is sensibly reduced.

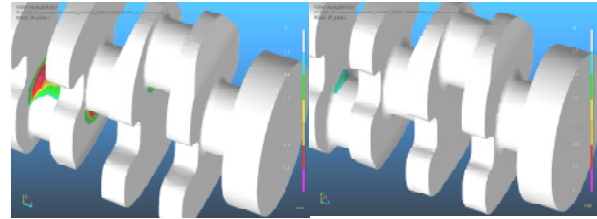


Fig. 18 Comparison of original (left) and optimized (right) crankshafts: safety factors

7 Conclusion

This paper illustrates the efficiency of modeFRONTIER in the multi-objective optimization of automotive components.

The software drives the optimization linking the different CAE software used in the model definition, and allows to find automatically with the lowest computational effort as possible the optimized solutions, in terms of best compromise between contrasting objectives.

The most complex application to the crankshaft harmonic and fatigue optimization has required, once the models from the designer are available, few hours for the workflow definition, and just two days of automatic runs to complete the optimization and find the optimized solution.

8 References

- [1] <http://www.esteco.com>
- [2] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. In *IEEE Trans Evol Computat Journal*, Vol.6, 2002.
- [3] A. Clarich, V. Pediroda, C. Poloni. A competitive Game Approach for Multi-objective Robust Design Optimization. In *AIAA 2004-6511*, Chicago, 20-22 September 2004.
- [4] <http://www.abaqus.com>
- [5] <http://www.solidworks.com>
- [6] S. Pravardhan. Dynamic Load Analysis and Optimization of Connecting Rod. Msc thesis, University of Toledo, May 2004
- [7] <http://www.femfat.com>
- [8] J. A. Nelder, R. Mead. A simplex method for function minimization. In *Computer Journal*, 1965
- [9] H. Muller-Slany, F. Brunzel, J.A. Pereira. Structural Damage Detection Based On Highly Accurate Updated Models. In *Proceedings IMAC-XIX*, Kissimme, Florida, USA, February 5-8 2001