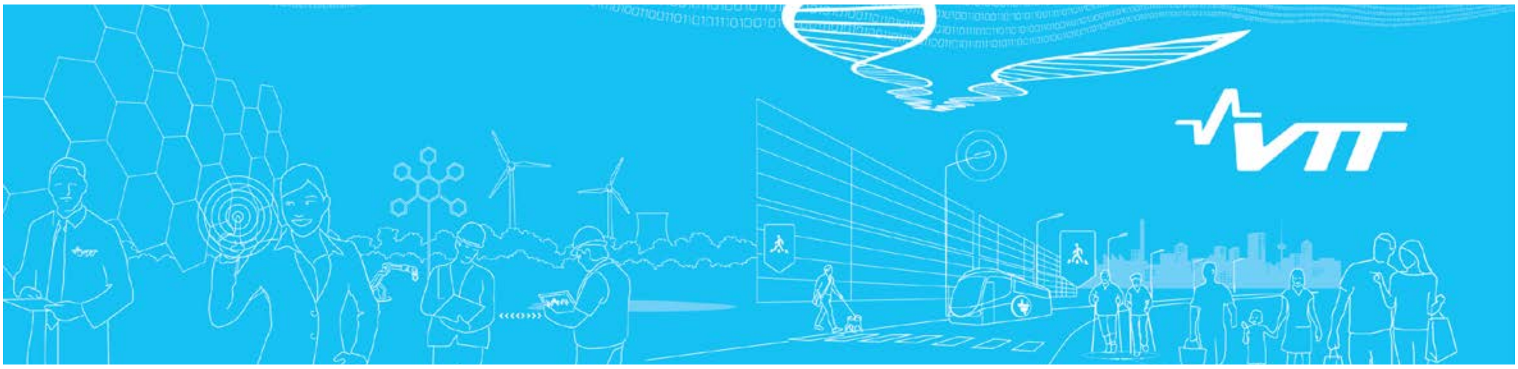
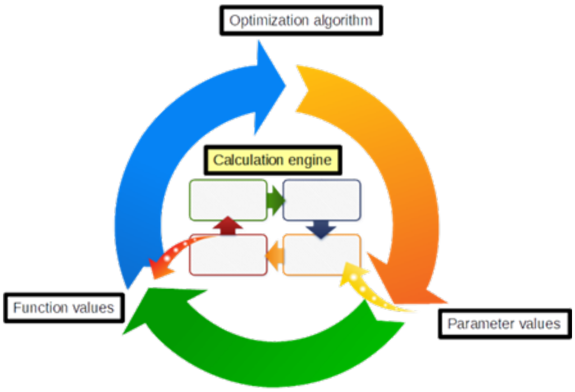


RESEARCH REPORT

VTT-R-04967-15



SIMPRO

SIMPRO Task 2.1. Optimisation of diesel-generating set vibration levels with Dakota

Authors:	Kai Katajamäki
Confidentiality:	Public

Report's title

SIMPRO Task 2.1. Optimisation of diesel-generating set vibration levels with Dakota

Customer, contact person, address	**Order reference**
Tekes, Matti Säynätjoki P.O. Box 69, FI-00101 Helsinki, Finland	1059/31/2012
Project name	**Project number/Short name**
Computational methods in mechanical engineering product development	78634/SIMPRO
Author(s)	**Pages**
Kai Katajamäki	16
Keywords	**Report identification code**
scripting, python, optimisation, Dakota	VTT-R-04967-15

Summary

In product design and development there is a continuous need to increase effectiveness. Timetables are getting tighter and cost awareness is of utmost importance. Product design process includes complicated analysis phases and the effective use of computer resources is demanded. Computational power is increasing all the time and at the same time the hardware costs are decreasing. Automatization of the analysis processes is the key factor if one wants to meet the demanding requirements in product design and in the same time to take all the computational possibilities into effective use. The increase in the computational power makes it also possible to take advantage of the numerous optimisation methods that are available. There exists several commercial optimisation software in the market as well as free software based on open software licenses. Moreover, easy to use and learn scripting languages are coming increasingly part of many analysis software. All these factors give great possibilities to develop automatic simulation systems incorporated into optimisation algorithms and thus providing an effective tool in designing better products and in the development of entirely new products. Automating demanding computational work is a key factor in the utilisation of optimisation methods. Automatic structural analysis systems give many benefits in the design of existing products and also in the design of new products. Modern and flexible analysis software combined with effective computer hardware and connected together with sophisticated and easy to program scripting languages makes it possible to develop wholly automatic and reliable calculation processes. These processes can then be integrated within optimisation algorithms thus enabling product optimisation and control of the inevitable uncertainty in model parameters. In this work such an automatic calculation engine that can be used in structural vibration analysis is presented. It has been connected to general optimisation software. Three different product optimisation cases are described and the results are compared.

Confidentiality	Public

Espoo 26.10.2015

Written by	**Reviewed by**	**Accepted by**
Kai Katajamäki Principal Scientist	Juha Virtanen Research Team Leader	Johannes Hyrynen Head of Research Area

VTT's contact address

VTT Technical Research Centre of Finland Ltd, P.O. Box 1000, FI-02044 VTT, Finland

Distribution (customer and VTT)

Matti Säynätjoki, Tekes, 1 copy
VTT, 1 copy

Contents

1. Introduction

In product design and development there is a continuous need to increase effectiveness. Timetables are getting tighter and cost awareness is of utmost importance. Product design process includes complicated analysis phases and the effective use of computer resources is required. Computational power is increasing all the time and at the same time the hardware costs are decreasing. Automatization of the analysis processes is the key factor if one wants to meet the demanding requirements in product design and in the same time to take computational possibilities into effective use. The increase in the computational power makes it also possible to take advantage of the numerous optimisation methods that are available. There exists several commercial optimisation software in the market as well as free software based on open software licenses. Moreover, easy to use and learn scripting languages are coming increasingly part of many analysis software. All these factors give great possibilities to develop automatic simulation systems incorporated into optimisation algorithms and thus providing an effective tool in designing better products or in the development of entirely new product families. Automating demanding computational work is a key factor in the utilisation of optimisation methods.

The advantages of structural analysis automatization and its incorporation in optimisation and parameter variation algorithms are many. For example one can find following benefits:

- Analysis time reduction and cost savings,
- Reliability increases and avoidance of human errors,
- Possibility to link different analysis types in the same calculation process. Temperature, stress vibration, acoustics, …
- Enables repeatability and modularity in analysis process,
- Automatic model and results documentation and reporting
- Increases understanding related to effect on input variables, sensitivity studies and statistical analyses.

Although the modern optimisation methods are becoming more and more common in industry there is still a clear need to make the use of these methods more effective. The first step could be to begin to use more often simple parameter studies. These can in certain cases already give the required answer for a design problem and time consuming optimisation may not be needed at all. This is currently the common industry practice although the number of cases is usually so limited that it resembles more a traditional trial-and-error type of problem solving. The key factor to make these methods useful to industry is to enable modification of calculation models straightforward without any user interaction or create automatic analysis systems and calculation engines. These combined with an effective hardware environment can help the industry to take these methods in effective use in their product design and possible problem solving cases. This kind of automatic calculation engine cannot be avoided in optimisation systems.

2. Automatic calculation system within optimisation

2.1 Calculation engine

In order to be able to effectively use optimisation in the support of product design certain requirements can be laid on analysis system. It should be computationally effective and relatively easy to modify to meet different analysis needs. A separate calculation engine which can be easily called within the optimisation software is a good solution for this. Schematic diagram how such a calculation engine is incorporated in optimisation is shown in Figure 1. The figure shows how the whole optimisation can be seen as two nested loops. The outer loop is controlled by the optimisation software. It can do a full optimisation task or a parameter study. Despite what kind of studies is conducted the principal behaviour is the same. The software expects to get required function values for given model or analysis parameter values. In the figure these are described as arrows into and from the calculation engine. Based on the given parameter values the engine should calculate the required function values as effectively as possible because in many practical cases the required number of calculation loops can easily be several thousands. The calculation engine might also do some model pre-processing based on the given parameters. In practice it must always do result post-processing in order to obtain the required results for the optimisation software. This result post-processing can as well be computationally demanding task. In practical cases it must also be possible to analyse parallel different cases which are required by optimisation algorithm. In parameter studies as many cases as the hardware infrastructure allows can be analysed at the same time. In gradient algorithms the calculation of gradients can be done simultaneously.

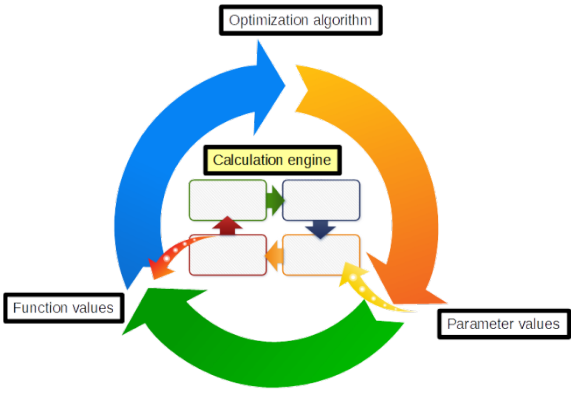


Figure 1. Automatic calculation engine within optimisation.

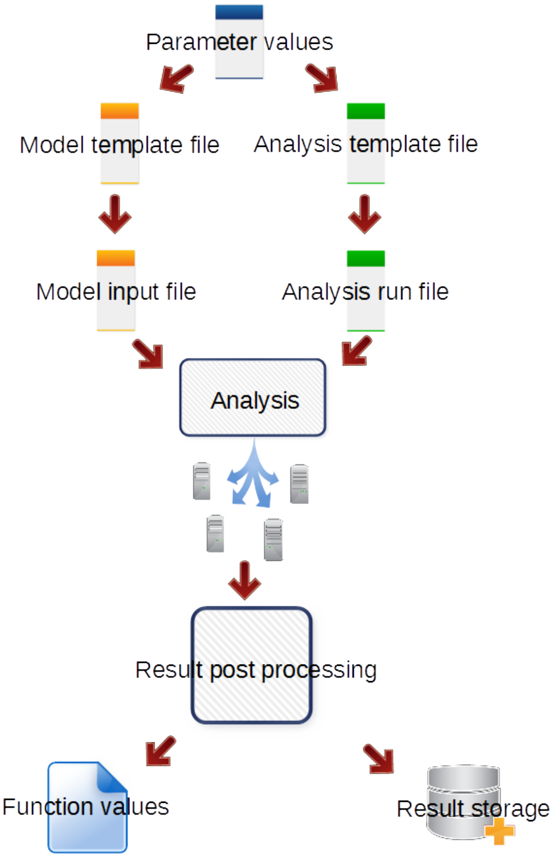


Figure 2. Calculation process.

The calculation engine can be created with different programming languages. In the very simplest it can be only a shell script that submits analysis software into run and fetches the required results after the run has been completed. For more complex cases in which model pre-processing as well as results post-processing is required a more general programming language may be more effective. The need may also be to chain various analysis types or software which in turn may require special processing between the runs. In this work calculation engine created with Python scripting language was used [1]. Basic information on Python language syntax with some examples are presented in [2]. The report also describes some important libraries which are available in Python.

The calculation engine is based on the idea that both the calculation model and the analysis process are based on parametrised template files. A diagram for the calculation process is shown in Figure 2. These files are simple text files which can be easily edited with any text processing program. The optimisation software writes the parameters into a simple text file. The calculation engine reads the parameters and based on those parameters writes the actual analysis input data files as well as the analysis run files. In this way not only numerical data can be parametrised but also more complex model or analysis features. For example whole element groups describing certain sub-structures in the design can be treated as parameters. Consequently, the optimisation of structural layouts becomes possible.

In the example case structural vibrations were considered. The calculation engine includes both the free and forced vibration analysis. The latter is made using modal superposition using the modes obtained from the free vibration analysis. The results post processing includes both the analysis of the mode shapes and the analysis of the forced response results.

In many cases one wants to follow how the frequencies of different natural mode shapes vary with different parameters. To identify natural mode shapes a special correlation function of two vectors is needed. In vibration analysis a so called modal assurance criteria can be used for this purpose. It is defined as

$$MAC(\{\Psi\}_r, \{\Psi\}_m) = \frac{|\{\Psi\}_r^T \{\Psi\}_m|^2}{(\{\Psi\}_r^T \{\Psi\}_r)(\{\Psi\}_m^T \{\Psi\}_m)}$$

In practise if the MAC value is greater than 0.7 … 0.8 the mode shapes can be considered to be corresponding. Typically a mode shape can be a beam type global bending or torsion mode or it can be purely local mode shape. If the calculation model has several millions of degrees of freedom and many mode shapes are analysed, the matrix calculation can become computationally extensive. One way to overcome this is to use some limited number of degrees of freedom. Programmatically very effective program can be created using the NumPy module (Python library) which has very effective functions for vector and matrix operations [3].

The forced responses are calculated separately for each of the engine excitation orders. The post-processing of the results is composed of a separate Matlab-routine which sums all the harmonic components into one function. The routine finds then the maximum value of this function. The required values can then be returned to the optimisation program. The calculation engine returns also the total mass which is used as the objective in the optimisation studies.

A flow diagram of the whole calculation engine is shown in Figure 3.

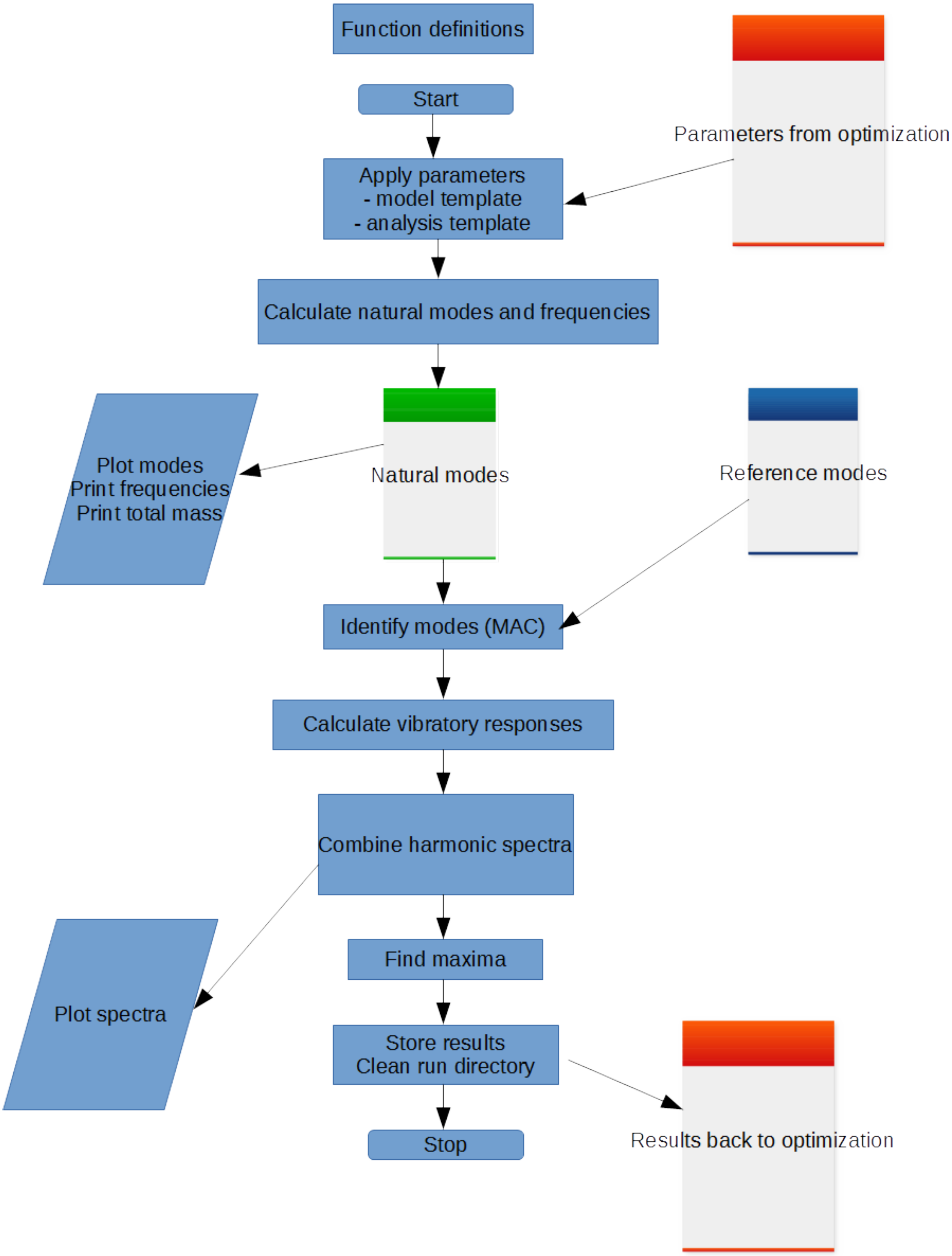


Figure 3. Calculation engine.

2.2 Analysis and optimisation software

Abaqus version 6.14-3 [3] was used as the structural analysis software. With it the natural mode shapes and frequencies were determined. After that the velocity responses due to the dynamic loading were calculated using modal superposition. Vibration spectra for engine excitation orders ½, 1, 1½, … 10 were calculated separately. The nominal running speed of the studied generating set is 750 rpm. In the optimisation a frequency window ranging from 700 rpm up to 800 rpm was considered. The discrete variables were taken into account with the help of Abaqus keyword “MODEL CHANGE”. With this keyword one can remove user given element sets from the calculation model. The calculation engine creates automatically proper commands using this keyword to remove those plates in the model which are not active in current optimisation loop.

In the parameter studies as well as in the optimisation runs the general optimisation software Dakota [4] was used.

3. Case description

VTT’s test diesel generating set was chosen as an example case for current studies. The generating set is of type Wärtsilä Vasa 4R32 and it is located at VTT’s engine laboratory in Otaniemi, Espoo, Finland. The generating set consists of the diesel engine driving a generator which is of type by ABB ALPC560 D8. Both are mounted at a common base frame. The base frame is a welded steel plate structure and it is mounted on a concrete block via flexible air springs. The engine is an inline four cylinder diesel engine operating at a constant speed of 750 rpm and producing 1640 kW of power. The generating set with its calculation model is shown in Figure 4. The calculation model is composed of 44528 elements and 47164 nodes. The number of degrees of freedom is 240627.

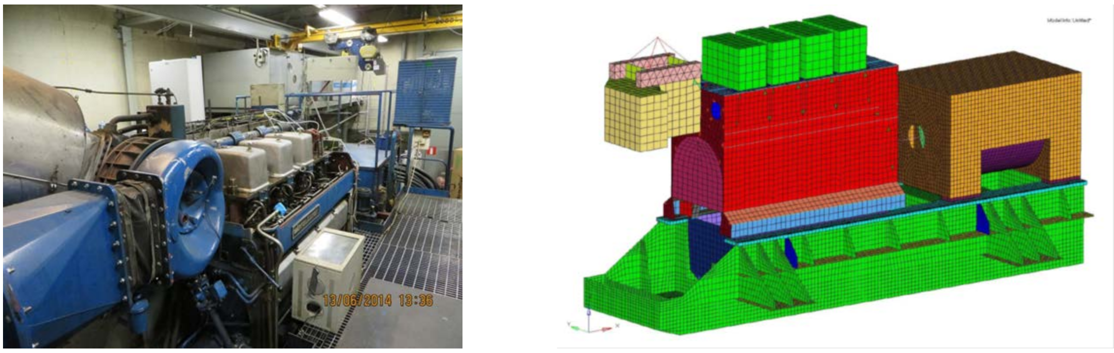


Figure 4. Diesel generating set and its calculation model.

The calculation model has been prepared during an earlier VTT self-funded research project. In this work the common base frame model was slightly modified. Some new artificial plates were modelled and the model was parametrised. Figure 5 shows the naming practice of the parameters. Inner plate named CHUTE1 exists in the real base frame but the inner plates CHUTE2 … CHUTE8 were added to the model afterwards. These chute plates are treated as discrete parameters so in the studies each of them either exists or doesn’t exist. The plate thickness of these chute plates was also considered as a parameter but continuous. Other continuous parameters were the thicknesses of plates SIDE (the two side plates), BOTTOM (the bottom plate), COVER (the cover plate) and the BULKHEAD (the transversal plates). The height of the base frame was also one parameter. It was realised by making the height co-ordinate, z-coordinate in this case, of the bottom plate nodes as one parameter. Thus a total of 14 parameters were used in the common base model, 8 discrete parameters and 6 continuous parameters.

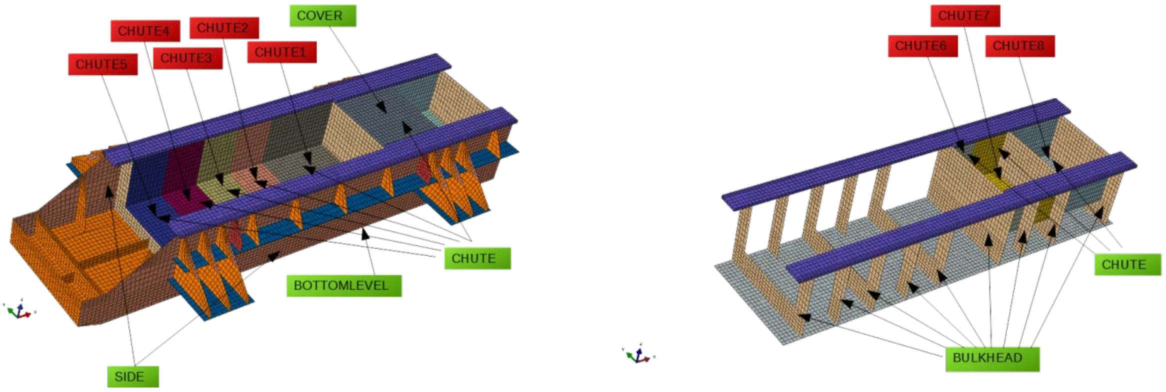


Figure 5. Naming of the parameters used in the common base frame model. The red ones are discrete parameters and green ones are continuous parameters.

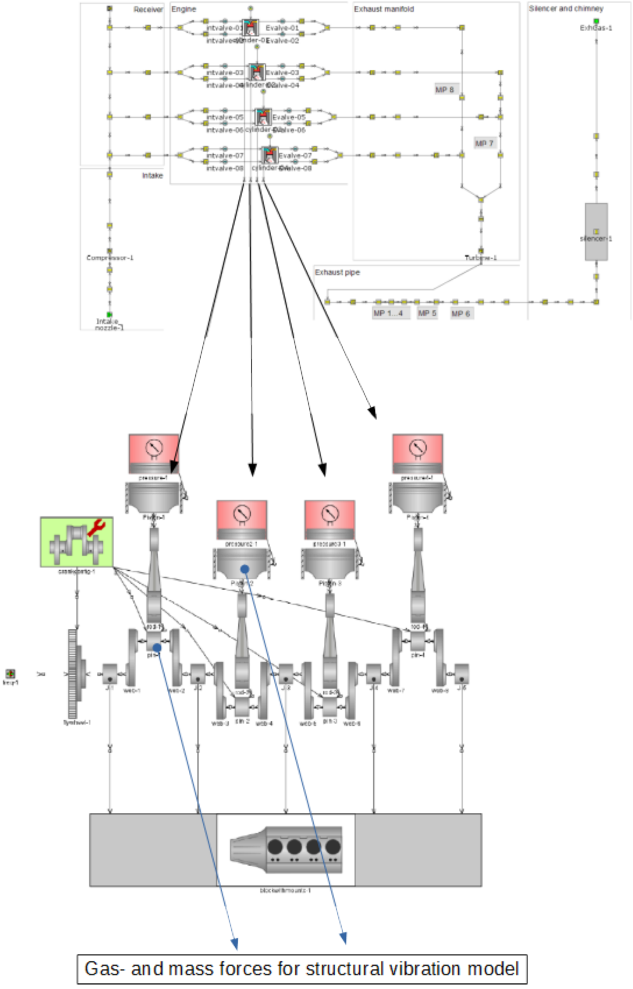


Figure 6. Simulation of engine internal processes with GT-Power and GT-Suite to obtain engine excitation forces for vibration analysis.

The engine operation was simulated using in previously calculated engine excitation forces. The excitation forces were obtained by simulating engine internal processes using GT-Power and GT-Suite software. The simulations were conducted by Antti Hynninen from VTT. A general view of the excitation calculation process is show in Figure 6.

The harmonic components ½, 1, 1½ … 9, 9½ and 10 of the engine excitations were analysed separately. Velocity responses at five generating set locations were calculated. The response points are shown Figure 7.

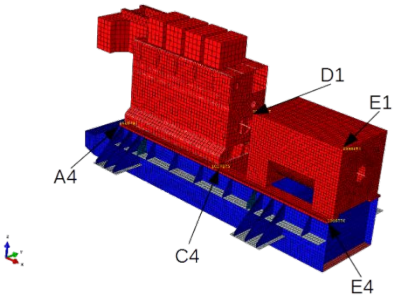


Figure 7. Response points.

The main objective function was the mass of the whole generating set. The dynamic vibration levels at the response points were used as constraints. In the beginning such parameter values were searched using a parameter study that gave strong resonance close to the nominal engine running speed. The resonance can be clearly seen in the calculated overall velocity spectra which are shown in Figure 8. This design was taken as the reference case. The corresponding first global natural mode shapes and frequencies are shown in Figure 9.

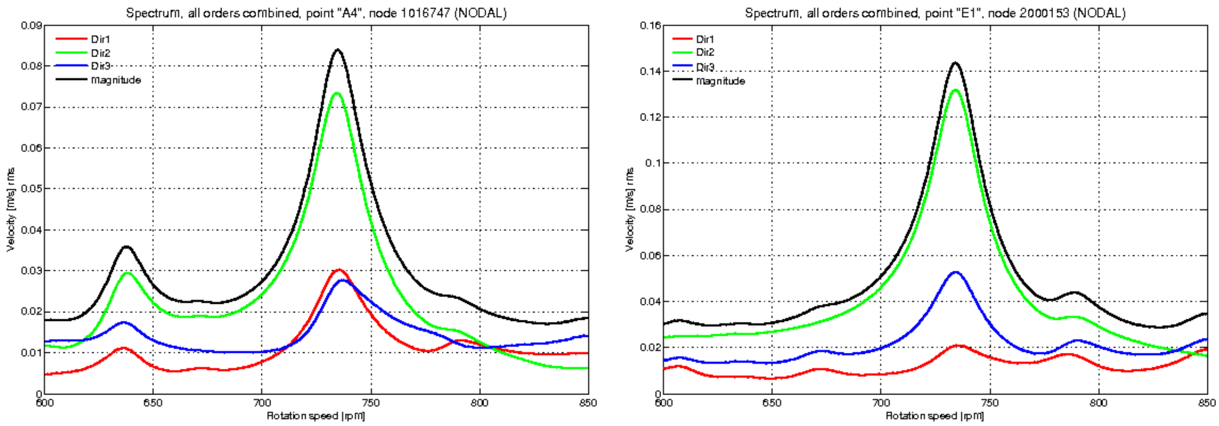


Figure 8. Overall velocity reponses at points A4 and E1 for reference model describing an almost resonance condition of the generating set.

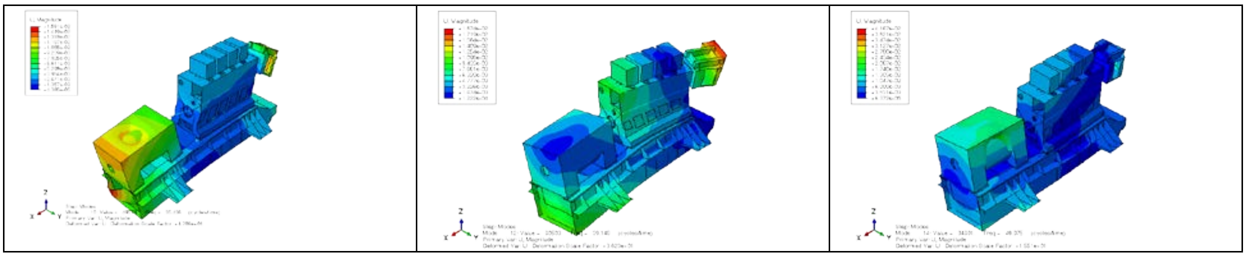


Figure 9. First global natural modes of the reference generating set structure. From left to right vertical bending at 35.4 Hz, horizontal bending at 39.1 Hz and torsion at 49.0 Hz.

4. Parameter variation studies

In the first phase a parameter variation study was conducted. The purpose was to obtain information on the general behaviour of the problem. Moreover, detailed information on the correlation between the results and parameters was obtained. On the other hand this information was used in setting-up the consequent optimisation studies.

In order to study the effect on plate thicknesses and chute plate layout on the responses Dakota's sampling method was used. The study cases were generated using Latin Hypercube sampling approach. A total of 1849 samples were generated. The results are summarised in the simple and partial correlation matrices in Table 1 and Table 2. Table 3 summarises the parameter study results. It lists the three best and three worst designs when the maximum vibration level within 700 rpm … 800 rpm is considered

Table 1. Simple Correlation Matrix among all inputs and outputs.

	SID	*COV*	*BLH*	*CHU*	*BOT*	*BOTH*	*CH1*	*CH2*	*CH3*	*CH4*	*CH5*	*CH6*	*CH7*	*CH8*	*Mass*	*A4*
SID	1.0															
COV	0.0	1.0														
BLH	0.0	0.0	1.0													
CH	0.0	0.0	0.0	1.0												
BOT	0.0	0.0	0.0	0.0	1.0											
BOTH	0.0	0.0	0.0	0.0	0.0	1.0										
CH1	0.0	0.0	0.0	0.0	0.0	0.0	1.0									
CH2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0								
CH3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0							
CH4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0						
CH5	0.0	0.0	0.0	0.0	0.0	0.0	-0.1	0.0	0.0	0.0	1.0					
CH6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0				
CH7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0			
CH8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0		
Mass	0.7	0.0	0.4	0.3	0.4	-0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.2	1.0	
A4	0.0	0.1	-0.3	-0.2	-0.1	0.0	0.0	0.0	0.0	-0.1	-0.1	0.0	0.0	0.0	-0.3	1.0
C4	0.0	0.1	-0.2	-0.2	-0.1	0.0	-0.2	0.0	0.0	-0.1	0.0	0.0	-0.1	-0.1	-0.3	0.9
E4	0.0	0.0	-0.2	-0.2	-0.1	0.1	-0.1	0.0	0.0	0.0	0.0	0.0	-0.1	-0.1	-0.3	0.9
D1	0.1	0.1	-0.2	-0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.1	0.0	-0.1	1.0
E1	0.2	0.1	-0.1	-0.1	0.0	0.0	0.1	0.0	0.0	-0.1	-0.1	0.0	0.0	0.0	0.0	0.9

Table 2. Partial correlation matrix between input parameters and total mass and velocity levels at studied locations.

	Mass	*A4*	*C4*	*E4*	*D1*	*E1*
SIDE	*1.0*	*0.0*	*0.0*	*0.0*	*0.1*	*0.2*
COVER	*0.3*	*0.1*	*0.1*	*0.0*	*0.1*	*0.1*
BULKHEAD	*1.0*	*-0.3*	*-0.2*	*-0.2*	*-0.2*	*-0.2*
CHUTE	*0.9*	*-0.2*	*-0.2*	*-0.2*	*-0.1*	*-0.1*
BOTTOM	*1.0*	*-0.1*	*-0.1*	*-0.2*	*0.0*	*0.0*
BOTTOMLEVEL	*-0.8*	*0.0*	*0.0*	*0.1*	*0.0*	*0.0*
CHUTE1	*0.8*	*0.0*	*-0.2*	*-0.1*	*0.0*	*0.1*
CHUTE2	*0.5*	*0.0*	*0.0*	*0.0*	*0.0*	*0.0*
CHUTE3	*0.5*	*0.0*	*-0.1*	*0.0*	*0.0*	*0.0*
CHUTE4	*0.5*	*-0.1*	*-0.1*	*0.0*	*0.0*	*-0.1*
CHUTE5	*0.5*	*-0.1*	*0.0*	*0.0*	*0.0*	*-0.1*
CHUTE6	*0.4*	*0.0*	*-0.1*	*0.0*	*0.0*	*0.0*
CHUTE7	*0.5*	*0.0*	*-0.1*	*-0.1*	*-0.1*	*0.0*
CHUTE8	*0.8*	*0.0*	*-0.1*	*-0.1*	*0.0*	*0.0*

Table 3. Summary of the parameter study. The five best and five worst designs with respect to the maximum vibration level.

ID	SID	COV	BLH	CHU	BOT	BOTH	CH1	CH2	CH3	CH4	CH5	CH6	CH7	CH8	Mass	Vel
1012	9	11	36	33	19	-1632	OFF	OFF	OFF	ON	ON	ON	ON	ON	1 %	-82 %
1097	9	13	50	48	20	-1709	OFF	OFF	OFF	ON	ON	OFF	OFF	OFF	2 %	-81 %
505	8	28	40	44	21	-1628	OFF	ON	OFF	ON	ON	OFF	OFF	ON	3 %	-81 %
1606	10	13	46	38	25	-1650	OFF	ON	ON	ON	ON	ON	OFF	OFF	4 %	-81 %
357	8	21	46	33	38	-1716	OFF	OFF	OFF	OFF	ON	OFF	ON	ON	5 %	-81 %
93	17	36	24	12	21	-1759	OFF	ON	OFF	OFF	OFF	OFF	OFF	OFF	-1 %	8 %
1647	16	29	15	14	35	-1704	ON	OFF	OFF	OFF	OFF	ON	OFF	OFF	-1 %	10 %
1558	12	34	10	40	23	-1755	ON	ON	OFF	OFF	OFF	OFF	OFF	ON	0 %	10 %
163	10	35	10	35	23	-1784	ON	OFF	OFF	OFF	OFF	ON	ON	OFF	-2 %	13 %
409	16	39	10	20	47	-1796	ON	OFF	OFF	OFF	OFF	ON	OFF	OFF	1 %	13 %

The results for the best case having number 1012 are reported below. Figure 10 compares the velocity spectra at two points with the reference case. It can be clearly seen that the strong resonance is moved to a lower frequency thus lowering the velocity levels at the running speed considerably. In Figure 11 the first global natural modes are shown and compared with the reference case.

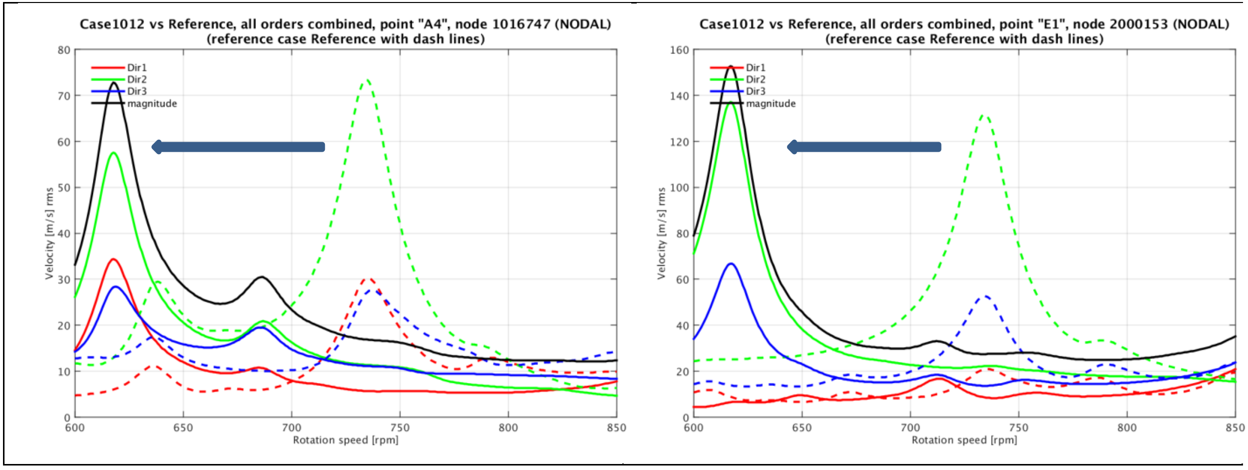


Figure 10. Overall velocity responses at points A4 and E1 for the best case 1012 and the reference case. The reference case is drawn with dotted line. The best case is drawn with solid line.

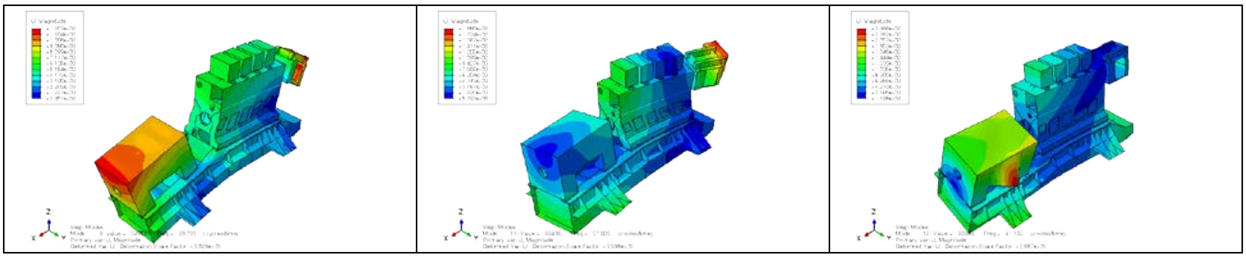


Figure 11. Natural modes for the best case 1012. From left to right vertical bending mode at 29.7 Hz (reference 35.4 Hz, MAC 0.84), horizontal bending mode at 37.8 Hz (reference 39.1 Hz, MAC 0.89) and torsion at 41.2 Hz (reference 49.0 Hz, MAC 0.89).

5. Optimisation studies

The optimisation problem was set up so that the objective function to be minimised was the mass of the base frame. The non-linear inequality constraints were given to the response point velocities. The constraint was defined on the overall velocity level in mm/s in engine running speed range 700 rpm … 800 rpm:

$$\begin{cases} 0 < A_4 < 25 \\ 0 < C_4 < 30 \\ 0 < E_4 < 30 \\ 0 < D_1 < 40 \\ 0 < E_1 < 40 \end{cases}$$

Two derivative free global optimisation methods were tried. The first was so called dividing rectangles (DIRECT) optimisation algorithm (coliny_direct). It balances local search in promising regions in the design space with global search in unexplored regions. The partitioning idea is shown in Figure 12. The purpose is to be able to effectively identify candidate solutions that can further be refined with fast local optimisers. Discrete design variables were ignored by coliny_direct so the optimisation variables were only the plate thicknesses and the common base frame height. The chute plate setup was taken to be the same as in the best parameter variation case 1012.

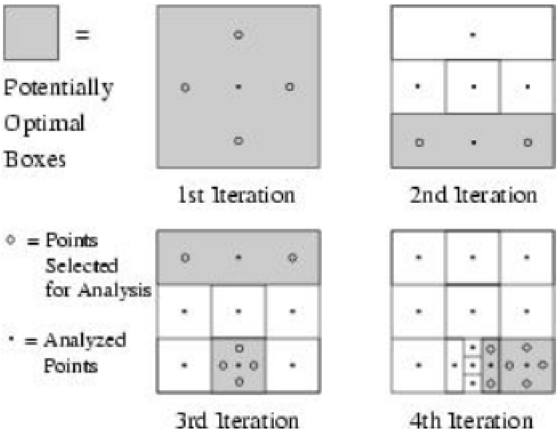


Figure 12. Design space partitioning with DIRECT algorithm [4].

The optimisation run required a total of 617 case evaluations. The optimal was archived at case number 613.

Figure 13 compares the velocity spectra at two points with the reference case. As can be clearly seen the strong resonance is moved to a lower frequency thus lowering the levels at the running speed enormously. In Figure 14 the first global natural modes are shown and compared with the reference case.

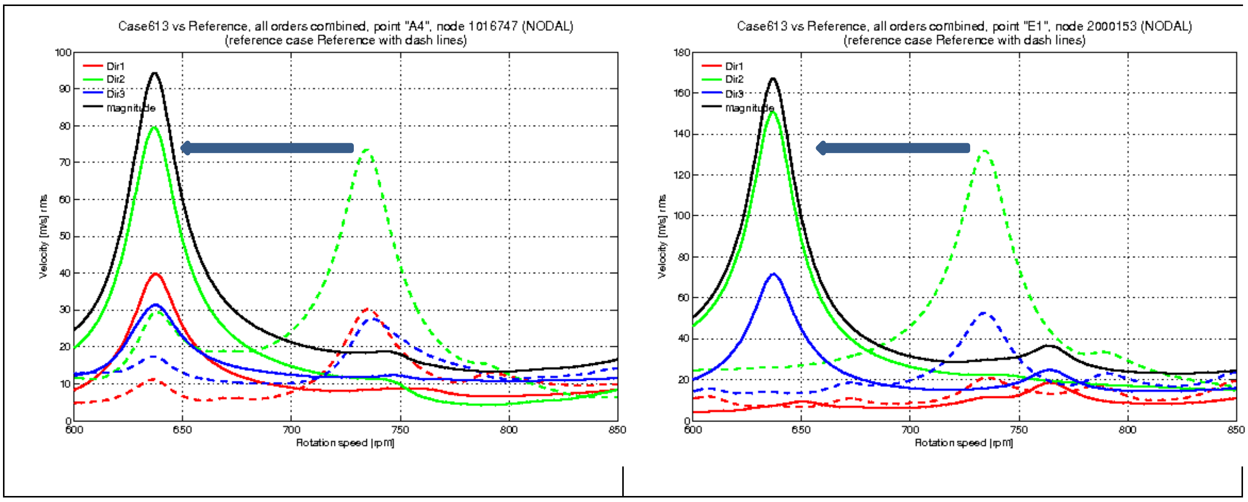


Figure 13. Overall velocity responses at points A4 and E1 for optimal case 613. The reference case is drawn with dotted line. The best case 613 is drawn with solid line. Optimisation coliny direct algorithm.

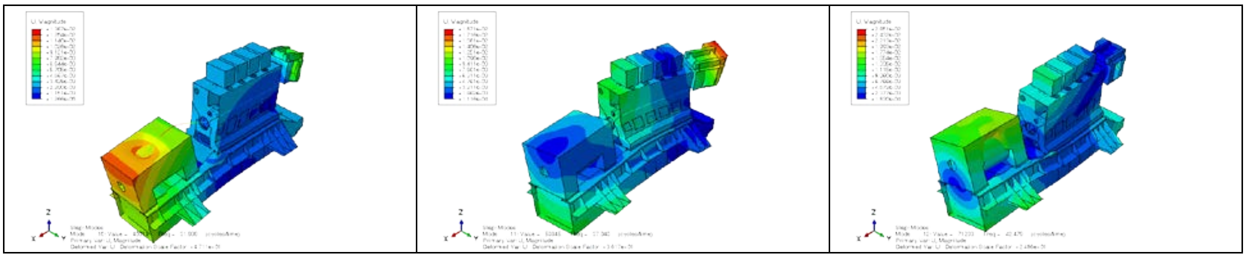


Figure 14. Natural modes for the best case 419. From left to right vertical bending mode at 31.8 Hz (reference 35.4 Hz, MAC 0.92), horizontal bending mode at 37.9 Hz (reference 39.1 Hz, MAC 0.92) and torsion at 42.5 Hz (reference 49.0 Hz, MAC 0.92). Optimisation coliny direct algorithm.

Another optimisation method which was tried was Dakota’s evolutionary algorithm (coliny_ea). The basic steps in this algorithm are shown in Figure 15.

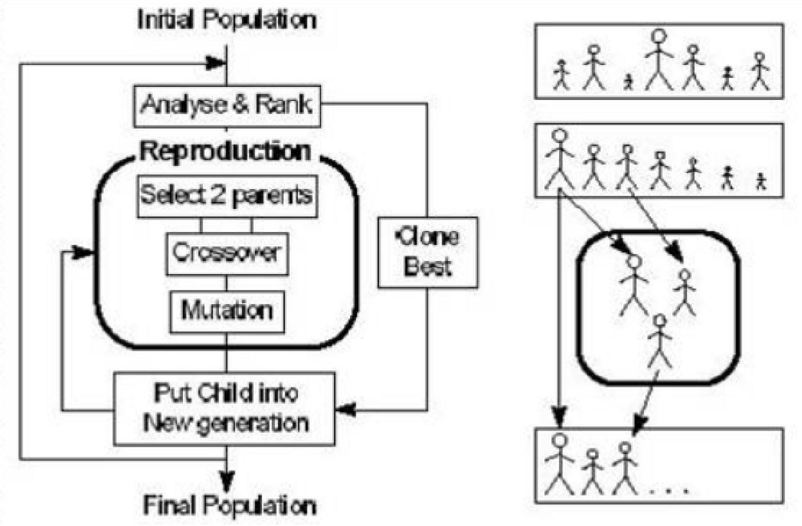


Figure 15. Basic steps of the evolutionary algorithm coliny_ea [4].

The basic steps are [4]:

1. Select an initial population randomly and perform function evaluations on these individuals
2. Perform selection for parents based on relative fitness
3. Apply crossover and mutation to generate new individuals from the selected parents
    a. Apply crossover with a fixed probability from two selected parents
    b. If crossover is applied, apply mutation to the newly generated individual with a fixed probability
    c. If crossover is not applied, apply mutation with a fixed probability to a single selected parent
4. Perform function evaluations on the new individuals
5. Perform replacement to determine the new population
6. Return to step 2 and continue the algorithm until convergence criteria are satisfied or iteration limits are exceeded.

The optimisation run required a total of 4032 case evaluations. The optimal was achieved at case number 3980. Figure 13 compares the velocity spectra at the two response points with the reference case. As with the previous methods also with this method the resonance is moved to a lower frequency thus lowering the levels at the running speed significantly. In Figure 17 the first global natural mode shapes are shown and the frequencies compared with the reference case.

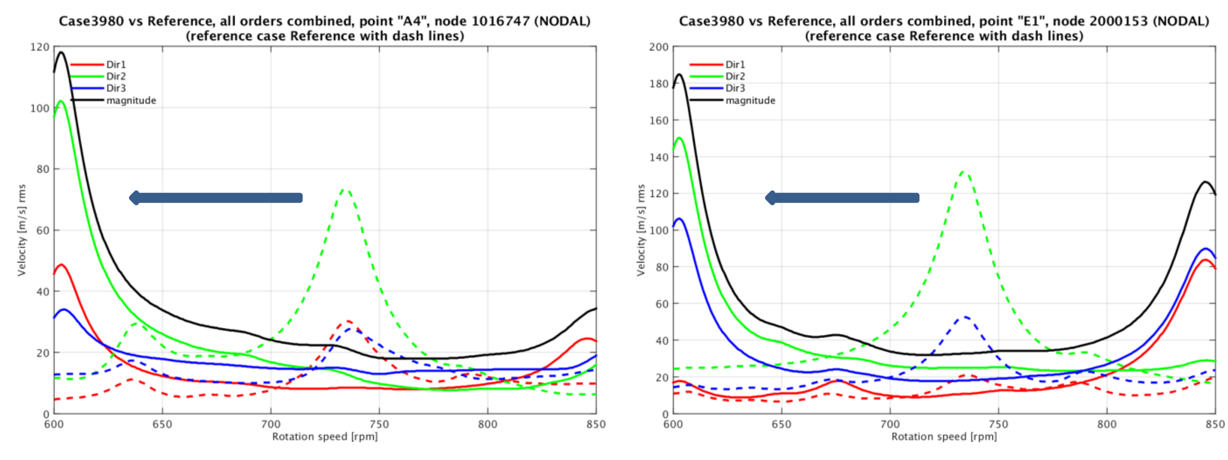


Figure 16. Overall velocity responses at points A4 and E1 for the case 3980. The reference case is drawn with dotted line. The best case 3980 is drawn with solid line. Optimisation with evolutionary coliny_ea algorithm.

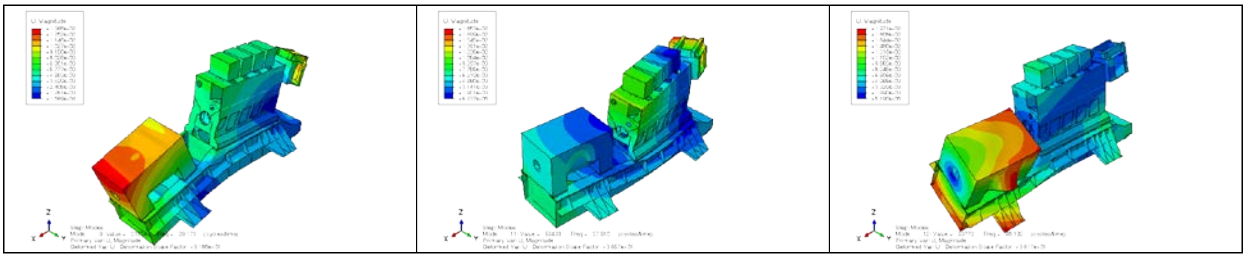


Figure 17. Natural modes for the best case 3980. From left to right vertical bending mode at 28.2 Hz (reference 35.4 Hz, MAC 0.87), horizontal bending mode at 37.8 Hz (reference 39.1 Hz, MAC 0.63) and torsion at 40.2 Hz (reference 49.0 Hz, MAC 0.78). Optimisation coliny direct algorithm.

The results of all the different methods are compared in Table 4. It can be seen that all the studied algorithms could lower the velocity levels remarkably. This was because the natural frequencies of critical natural modes were moved to lower frequencies. Big differences can be seen in the number of required iterations. With the parameter variation method 1849 samples were analysed. This number was quite arbitrarily chosen. With this sample number the same velocity level decrease was achieved than with the two optimisation methods. But because there was no mass target the mass in fact remained the same as with the reference case. Evolutionary algorithm could lower the mass almost 10 % but the dividing rectangles (DIRECT) method could not decrease the mass at all. The latter required 617 iterations compared with 4032 iterations required by the evolutionary algorithm.

Table 4. Comparison between the best designs obtained with the parameter study (Par) and the two optimisation studies (Evo and Dir).

ID	*SID*	*COV*	*BLH*	*CHU*	*BOT*	*BOTH*	*CH1*	*CH2*	*CH3*	*CH4*	*CH5*	*CH6*	*CH7*	*CH8*	*Mass*	*Vel*
Evo	10	8	8	8	9	-1610	OFF	OFF	OFF	ON	ON	ON	ON	ON	-9 %	-78 %
Dir	10	15	15	15	43	-1705	OFF	OFF	OFF	ON	ON	ON	ON	ON	-0.1 %	-78 %
Par	9	11	36	33	19	-1632	OFF	OFF	OFF	ON	ON	ON	ON	ON	1 %	-82 %

6. Conclusions

Automatic structural analysis systems give many benefits in the design of existing products and also in the design of new products. Modern and flexible analysis software combined with effective computer hardware and connected together with sophisticated and easy to program scripting languages makes it possible to develop wholly automatic and reliable calculation processes. These processes can then be integrated within optimisation algorithms thus enabling product optimisation and control of the inevitable uncertainty in model parameters.

Three different algorithms were tested in a case were a strong resonance is close to the nominal operating point. With all the studied methods the resonance could be moved away from running frequencies. With simple parameter study as well as with the dividing rectangles (DIRECT) algorithm the mass remained unchanged. Evolutionary algorithm could deliver the best result. The resonance was moved but also with the solution the generating set's mass was 10 % decreased.

The calculations were conducted in VTT's HPC environment. The optimisation software Dakota as well as the calculation engine was running at the front-end node. The calculation engine submitted the finite element analyses to the calculation nodes. At the most 15 cases were sent to execution simultaneously. Because the finite element model had relatively few degrees of freedom the total analysis time for one case was quite short. It took only about 2 minutes to analyse one case from the very start to the end. Nowadays typical calculation models may have millions or even tens of millions degrees of freedom. As the optimisation requires easily thousands of cases to be analysed becomes the analysis time for one case critical. It is then necessary to decrease the model size. One way is to use approximate much smaller surrogate models. These can be created for example with various sampling or surface fitting methods. The creation of these models may however be computationally time consuming. Another way is to use some method to reduce the model size by reducing the size of sub-models which remain unchanged in optimisation. It is possible to use so called super elements. Using this kind of special elements the finite element models can be compressed into much smaller size and still being able to describe the structure's dynamics properly.

References

1. https://www.python.org/
2. Katajamäki, K. Scripting in high performance computing environment. VTT Research Report VTT-R-03685-15. Espoo. 2015.
3. ABAQUS 6.14 Documentation.
4. Adams, B.M., Bauman, L.E., Bohnhoff, W.J., Dalbey, K.R., Ebeida, M.S., Eddy, J.P., Eldred, M.S., Hough, P.D., Hu, K.T., Jakeman, J.D., Swiler, L.P., and Vigil, D.M., "DAKOTA, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.2 User's Manual," Sandia Technical Report SAND2014-4253, May 2015.