Multi Objective optimization of a car vehicle coupling FRONTIER and MDI-ADAMS

G. Perna¹, M. Poian², C. Poloni³

¹ Engin Soft Trading srl - Trento (Italy) E-mail: gino@enginsoft.it
² Engin Soft Trading - ESTECO - Trieste (Italy) E-mail: Mauro.Poian@com.area.trieste.it
³ Dipartimento Energetica - Trieste University - Trieste (Italy) email:Poloni@univ.trieste.it

Abstract: Thanks to the multi Objective capabilities of FRONTIER, it was possible to optimize simultaneously comfort and handling manoeuvres. Both options of coupling ADAMS ADAMS directly and A/Insight module have been successfully verified. Multiple runs with different manoeuvres are taken into account into the definition of the design problem. FRONTIER system allows to produce several Pareto-Optimal configurations computed by ADAMS and then prompted to the designer whose responsibility is the final best choice.

Keywords: Global Optimization, genetic algorithms, optimization/constraint, multi objective, pareto sets.

Introduction

Optimization is a frequent task in many fields of science and engineering. Automated design is, as well, an important tool in any vehicle industry as it greatly reduce the development period by excluding any human interactions.

It is vital in today's competitive environment because the success of a product depends both on the cost and timeliness as well as on its quality. Toward the automated design of a vehicle suspension components, ADAMS Virtual Prototyping Environment and FRONTIER Optimization Environment have been coupled together to became the state of the art tool in vehicle performance optimization.

A true multi-objective optimization can be performed by the software that, sitting on the simulation codes, drive the virtual prototyping sequence to find the best compromise of optimal designs, taking into account various design variables and constraints (related to
the car itself or to subsystems like the car suspension), as well as different (and sometimes subjective and/or conflicting) technical, marketing and product metrics.

The multi-objectives approach is clearly the only one consistent both from the everyday life and from the engineering/designer viewpoint, whenever there is the need for a search for a ‘best compromise’ solution among different and maybe conflicting objectives, under different constraints, or more precisely a complex process has to be faced, including:

- a number of objectives (requirements, or fitness), which should be satisfied ‘at best’;
- several combinations of available choices;
- different constraints to be satisfied, sometimes strictly (i.e. those laid down by regulations), while sometimes they may be relaxed if necessary;
- different perception, or weight, for each objective, according to the ‘user’, without the possibility to clearly distinguish what is more important from what is not.

With respect to such ‘general’ environment, a design optimization process poses further conditions, in that a design has to be viewed in relation to its ability to undergo quantitative evaluation of its merits. Therefore, the design, generally conceived as a definition of a product or process, should be:

- complete enough to allow the significant measures of its business effectiveness to be evaluated (preferably objectively and consistently via a defined evaluation process);
- so that it is one candidate from a well defined class of designs, which allows ‘cause and effect’ to be established between a design change and the design’s business effectiveness.

The final choice remains always subjective and is left to the designer (in broad terms: manager, product designer, marketing manager etc.). However, it is important to understand that such final selection should nevertheless be based on the ‘family’ of best designs, i.e. those designs which cannot be improved at the expense of a degradation of one or more objectives.

How these concepts can be implemented into a vehicle multi-body scenario?

Normally the key-problems are related to suspension geometry and parameters in different manoeuvres; different computer-programs may also be involved (i.e. computing tire parameters, flexibility, and different simulation codes).

Such problem can be formalized as a set of numerical parameters:

- The geometry of the suspension system;
- The physical properties of the suspension system;
- The Tire model;
- Different Handling manoeuvres;
- Different Comfort manoeuvres.

The final aim is really attractive: using the numerical simulation performed by ADAMS, as well as the DOE tools in A/Insight, as ‘input suppliers’ for FRONTIER
Variables and objectives

In this article a reference vehicle has been parametrized in conjunction with MDI-Italy considering only very basic parameters but two different and contrasting manoeuvres.

The two systems were tested both using FRONTIER capabilities to manage directly ADAMS input files, or using A/Insight DOE experiments to find out better compromise objectives systematic analysis.

VARIABLES

In detail, the variables involved are the following:

1. front_dampers_scale: front suspension damper curve scale;
2. front_springs_scale: front suspension spring curve scale;
3. front_hp_lca_front_y: Front suspension lower control arm front hard point position Y;
4. front_hp_lca_front_z: Front suspension lower control arm front hard point position Z;
5. front_hp_lca_rear_y: Front suspension lower control arm rear hard point position Y;
6. front_hp_lca_rear_z: Front suspension lower control arm rear hard point position Z;
7. rear_dampers_scale: Rear suspension damper curve scale;
8. rear_springs_scale: rear suspension spring curve scale.

OBJECTIVES

Handling:

1. Minimize the absolute maximum chassis pitch angle [degree];
2. Minimize the absolute maximum chassis roll angle [degree];
3. Maximize the minimum front left tire normal force [N];
4. Maximize the minimum rear left tire normal force [N];

Comfort:

5. Minimize the absolute maximum Steering Bar acceleration Z [g];

Other objectives were:

6. Minimize the absolute maximum front left tire lateral slip angle;
7. Minimize the absolute maximum rear left tire lateral slip angle;
8. Minimize the absolute maximum front right tire normal force;
9. Minimize the absolute maximum rear right tire normal force;
10. Minimize the absolute maximum front right tire lateral slip angle;
11. Minimize the absolute maximum rear right tire lateral slip angle.
these supplemental objectives were computed but not taken into account in the optimization process, to produce a better readable result and because some of them are linear dependent from others already taken into account.

**Case study**

The case study has been carried out on the DEMO MDI vehicle shown:

**FIGURE 1. MDI Demo Vehicle**

The main objectives are:

1. Maximize the performance in handling manoeuvres
2. Maximize the comfort for passengers

The two simulations are typical examples of handling and comfort maneuvers for a full vehicle model. Although this is an example vehicle, it is well representing the problem of optimizing and compromising both needs, i.e. having a stable vehicle (stiff) while turning and a comfortable vehicle (soft) while running on uneven roads. Simulations have been run on a Pentium III 866MHz.

**HANDLING SIMULATION**

mail parameters:

- Constant radius turn, 100m;
- Starting speed of 27m/s;
- 0.5g deceleration to 15m/s;
- 0.5g acceleration to 27m/s;
- 7s real time;
Case study

- About 70s computing time.

**COMFORT SIMULATION**

Main parameters:
- Drive Straight at 20m/s on a sinusoidal profile (\(\lambda=5m\), height=25mm);
- 5s real time;
- About 30s computing time.

The model was parametrized and a 1000 DOE experiments have been run; the results were processed by A/Insight and a javascript mail produced. The FRONTIER environment was set up on the results of A/insight results to speed up the process and find the best choice among the possible best designs found.

**FIGURE 2. Typical output from comfort manoeuvre.**
Setting up the optimization process and running the test-case.

A complete description of optimization setting requests follows to show all procedure steps.

PROCESS FLOW

The first job to define the problem is to construct the Process Flow. This can be activated by clicking on the correspondent icon or selecting the ‘Process Flow’ item under the Project menu.
Setting up the optimization process and running the test-case.

The window is divided in two parts:

- the top part (“Graphic Flow”): is the desktop where the logic is created adding and connecting the node.
- The components can be placed on the Graphic Flow Desktop by clicking with the left-mouse-button on the left tool bar to select the tool and then on the table to place the component. A node can be selected by one mouse click, edited by double mouse click, deleted or modified by clicking the right-mouse-button;
- the bottom part (“Logic Log”): displays the errors and comments created by the just-in-time logic compiler.

The logic is created and the problem is correctly defined if and only if no errors are listed in the logic log (as in our case).

**FIGURE 5. Process Flow window with the problem logic flow.**

**INPUT VARIABLES**

The input variable is a free parameter that can be modified by the optimizer.

The problem has 8 input variables:

<table>
<thead>
<tr>
<th>Factor</th>
<th>Min.</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>front_dampers_scale</td>
<td>0.8</td>
<td>1.2</td>
</tr>
<tr>
<td>front_springs_scale</td>
<td>0.85</td>
<td>1.15</td>
</tr>
<tr>
<td>front_hp_lca_front_y</td>
<td>360</td>
<td>440</td>
</tr>
<tr>
<td>front_hp_lca_front_z</td>
<td>160</td>
<td>200</td>
</tr>
<tr>
<td>front_hp_lca_rear_y</td>
<td>410</td>
<td>490</td>
</tr>
<tr>
<td>front_hp_lca_rear_z</td>
<td>165</td>
<td>205</td>
</tr>
<tr>
<td>rear_dampers_scale</td>
<td>0.8</td>
<td>1.2</td>
</tr>
<tr>
<td>rear_springs_scale</td>
<td>0.85</td>
<td>1.15</td>
</tr>
</tbody>
</table>
Setting up the optimization process and running the test-case.

In the summary panel it is possible to see and change the range and base of all the variables. Note that the variables need not to be continuous.

**FIGURE 6. Summary panel with input variables characteristics.**

<table>
<thead>
<tr>
<th>Input Vars</th>
<th>Output Vars</th>
<th>Trans Vars</th>
</tr>
</thead>
<tbody>
<tr>
<td>owskiName</td>
<td>Lower Bound</td>
<td>Upper Bound</td>
</tr>
<tr>
<td>r_damp, s</td>
<td>0.0</td>
<td>1.2</td>
</tr>
<tr>
<td>r_damp, s</td>
<td>1.2</td>
<td>10</td>
</tr>
<tr>
<td>r_damp, s</td>
<td>16.0</td>
<td>20.0</td>
</tr>
<tr>
<td>r_damp, s</td>
<td>41.0</td>
<td>43.0</td>
</tr>
<tr>
<td>r_damp, s</td>
<td>16.0</td>
<td>20.0</td>
</tr>
<tr>
<td>r_damp, s</td>
<td>20.0</td>
<td>24.0</td>
</tr>
<tr>
<td>r_damp, s</td>
<td>0.0</td>
<td>1.2</td>
</tr>
<tr>
<td>r_damp, s</td>
<td>1.2</td>
<td>10</td>
</tr>
</tbody>
</table>

**OUTPUT VARIABLES**

The output variable is a quantity produced by the design logic. Our problem has 12 output variables that are values that we would like to monitor; some of those are taken from the comfort manoeuvre some from the handling manoeuvre.

- `chassis_pitch_ABS_MAX`
- `chassis_roll_ABS_MAX`
- `chassis_yaw_rate_ABS_MAX`
- `chassis_yaw_rate_END`
- `left_tire_f_normal_front_min`
- `left_tire_f_normal_rear_min`
- `left_tire_lateral_slip_front_ABS_MAX`
- `left_tire_lateral_slip_rear_ABS_MAX`
- `right_tire_forces_normal_front_MAX`
- `right_tire_forces_normal_rear_MAX`
- `right_tire_lateral_slip_front_ABS_MAX`
- `right_tire_lateral_slip_rear_ABS_MAX`
- `MAX_Steering_Bar_Acc_Z`

**INPUT FILES**

An input file is a file of data from which an application can retrieve all arguments needed to run. It can feed one or more application with the data needed. Our input variables go in two different template input files: the handling simulation and comfort simulation.
Setting up the optimization process and running the test-case.

FIGURE 7. Template Input Editor window with problem variables inserted.

As an example the XML tag `<VAR name="fr_dampers_sc" format="%f" />` means that the variable named fr_dampers_sc will be inserted in that file point as a float (the format must be given in C style).

All the tags are handled by the mouse and with minimal interaction to the user. This procedure is very easy; selecting the row in the table of variables and right clicking in the file editor are the only actions to do.

OUTPUT FILES

An output file is a file of data, its purpose is to save the data an application has produced. Our output file “resultc.dat” has a fix length and does not depend on the results of the analysis. In this case is possible to extract variables using absolute position rules. In case of absolute position the output value is extracted from the file in a similar way as an input variable is inserted into an input file.

FIGURE 8. Summary panel with output variables characteristics.
Setting up the optimization process and running the test-case.

The rule TYPE=ABSOLUTE, ROW=13, COL1 means that the value of M_S_Bar_Acc_Z is the first column of the 13th row of our output file and is always at the same position.

APPLICATION

The component named “run_handling_comfort.sh” is the object responsible for the execution of our logic and correct handling of ADAMS or the simulation code (in our example the javascript produced by A/Insight). This object is the core of the optimization process. It retrieves data from input files, it makes all the computing and then outputs data to output files. To obtain this is necessary to add a “New Application” object from the Graphic Flow tool bar and place it on the Graphic Flow Desktop. After the application node has been configured in this way:

1. connected with the Start Logic item to indicate that the “run_handling_comfort” node is going to be the first to be executed;
2. connected with the three input files items to provide the application with the proper input data;
3. connected with the output file item “resultc.txt” and “resulth.txt” to specify the application output data corresponding to an application exit condition;
4. connected with the Completed Logic item “End23” in case of correct application exit;
5. connect the Failed Logic item “End12” in case of bad application exit (In our flow we assume that the successful application exits with 0 otherwise it exits with a non 0 value);
6. associated with a valid shell script.

The computing process of every application can be edited like a normal batch file. The script responsible for the execution of our logic flow is shown in this figure.

FIGURE 9. Window for scripting with our problem script.

GOALS

In our problem we used five goal objects to set our targets for the optimization process. It is possible to link a goal to input or output variables and specify a mathematical expression that describes that goal as a function of connected variables.
Setting up the optimization process and running the test-case.

The optimizer will try to reach that goal. We connect our goals only to output variables with the following roles:

**FIGURE 10. Summary panel with objectives characteristics.**

The aim of our work is to minimize all the angular quantity and acceleration quantity and to maximize the minimum normal forces.

Now the logic is completely defined and there are no errors in the logic log in the bottom part of window.

**DESIGN SCHEDULER**

Our problem has 3 goals to minimize and 2 goals to maximize so a Multi Objective Optimizer was selected with 12 iterations. The choice of 2 or more concurrent design evaluations lets multiple analysis to be performed in parallel.
FIGURE 11. all MOGA settings

DESIGN SPACE

The Design space window contains a table with all the generated design quantities, input, output, objectives and constraints plus a marker and the design ID. This table is filled run-time till the end of execution.

At the bottom of the window global numbers are given: number of designs, number of errors, number of designs that are violating constraints, number of marked design.

At the end of our run we obtained 853 different designs. First of all we filtered all the designs in design space to obtain only designs that belong to Pareto frontier. This filter marked 325 different designs.

PARETO DOMINANCE PRINCIPLE

The concept of Pareto optimality is the basis on which is grounded most of cooperative multiple optimizations. It is base upon the principle of dominance which can be defined by:

Let \( F = (c_1, ..., c_G) \) vector of a minimization problem with G objectives. Let \( F1 \) and \( F2 \) be two candidates. \( F1 \) dominates \( F2 \) if

\[
(\forall i \in \{1, ..., G\}, c_1^i \leq c_2^i) \text{and} \exists! (c_1^i \leq c_2^i)
\]
Setting up the optimization process and running the test-case.

A solution $F1$ is said to be nondominated if there is no feasible solution in the search space which dominates it. The Pareto front is the set of all the nondominated solutions.

Therefore, a solution belonging to the Pareto front is optimal in the sense that there exist no other design which is better with respect to all design goals. In other words, such a solution can be improved with respect to one goal only by degrading its performance with respect to another.

**PARETO FRONTIER EXTRACTION**

Using the integrated Chart Desktop we created different type of graphs to view our results. The marked points belong to Pareto frontier.

**FIGURE 12.** The design space window

A parallel co-ordinates chart shows the 5 goals value of all designs.
Setting up the optimization process and running the test-case.

FIGURE 13. All designs in a parallel co-ordinates chart (black lines are a set of Pareto designs)

We made a filter on designs lowering the red line in the graph. So we could select a reduced set of the pareto frontier choosing the best compromise between all objectives.

FIGURE 14. Typical Scatter plot of the design space.
Setting up the optimization process and running the test-case.

FIGURE 15. Parallel co-ordinates chart with a filter on objective values.

FIGURE 16. Marked Selected Objectives scatter plot

MCDM FOR RANKING DESIGNS

The main aim of MCDM is to take a set alternative designs, each evaluated on several criteria, rank them in preference order based on a set of pairwise comparisons and provide a set of utility function to describe the preferences.

The set of utility curves is automatically produced by calculating weights and parameter alpha of a set of curve like:
Setting up the optimization process and running the test-case.

The aim of rational decision making, therefore, is to maximize the positive consequences and minimize the negative ones. As these consequences are directly related to the decision made or option chosen, it is not unreasonable to treat the consequences as aspects of performance. The decision problem then becomes a matter of considering these aspects of performance of all the options available simultaneously so that the decision maker can exercise his choice. In other words, rational decision making involves choice within the context of multiple measures of performance or multiple criteria.

At the end of the process, it is possible to mark the best solution and the worst design.

Some design relationships between Pareto design as shown in the following picture have been set.

Running MCDM we obtain the following design’s rank that suggest us that design 611 is the best for our purpose.
Setting up the optimization process and running the test-case.

**FIGURE 19. MCDM result window**

**IMPROVEMENT OF THE MULTI OBJECTIVE DESIGN**

Based on the utility non-linear function found we could change the flow of the process to take into account only one goal (“the best of the best”). The function is automatically stored in the MCDM database and made available to the frontier user in very simple way. A common gradient-based optimizer o GA optimizer could be started to better improve the design.

In our case, we started a new GA Moga optimizer on the new goal function and found a better design as viewed from the following pictures:
Conclusions

The optimization sequence that has been run using an MDI demo vehicle with FRON- TIER optimization module shows how efficient and easy is the definition and control of a multi-objective optimization procedure in connection with ADMAS, taking into account a variety of parameters and objectives.

An overall better performance of about 20% was achieved on all objectives.
Conclusions

Further advantages can easily be achieved if simultaneous ADAMS runs are carried out on a low cost cluster of PCs:

- time reductions are tremendous to the extent that the computer simulation together with FRONTIER optimization tool become a ‘true’ design tool, with growing advantages with respect to the industrial demand.

Clearly any software tool is nothing more than just a tool in the hands of the designer. But, assuming the designer is conscious of the process he is working at and hence he is able to detail and qualify the objectives as well as the design variables ranges and limits, the information that he can get from a software system such as

ADAMS coupled with FRONTIER is by far more useful than any study which is simply structured on a trial and error procedure. In a word, using FRONTIER and ADAMS the designer can take the maximum possible advantage from his experience and from the simulation work and budget.