CRASH MANAGEMENT SYSTEM OPTIMIZATION TOOL IN ANSA AND META.

Pedro Ruiz, Alvaro Arconada, Ricardo Liñán.
SOLUTE Ingenieros, Spain

KEYWORDS –
Crashbox, Crash, low speed, RCAR,

ABSTRACT –

The crash management system, namely CMS, is located at both ends of the body car serving as an energy dissipation element that aims to control the local collapse of structure when the car suffers a low speed impact. The target is that both CMS structures (front-end and rear-end) not only meet legislative requirements, but also improve the damageability and reparability of the vehicle. This ensures the cost remains as low as possible, and the insurance companies reflect this decrease in the insurance policy.

There are also additional scenarios that must be considered in their design, such as the compatibility with high speed crashes or pedestrian protection and the integration into the overall concept of the vehicle. Needless to mention that the cost effectiveness is a key factor in the determination of definitive solution.

Within the CMS, the crashbox plays a vital role in the energy absorption. The parameters that determine its behaviour at low speed impacts are several, but geometric features (thickness, beads, height, width, cone-shaped angles) do offer a very open scenario when a new crashbox is designed. As there are so many possibilities, a quick definition of each one is key to sweep all the desired combinations.

With all this, the development of a FEM automation tool was decided in SOLUTE to speed-up the time of development, as well as the internal productivity.

The target of the tool is to set up the different crashbox geometries to be simulated as well as respond to the simulation results to reach a closer solution to the optimum geometry for each car design.

To achieve this, the new tool is defined by scripting in ANSA and META. As a result, engineers can study the sensitivity of each design parameter to match their production requirements.

1. ABOUT SOLUTE

Solute is a multi-disciplinary engineering firm headquartered in Madrid, Spain. We take great pride in our expertise, clients and projects. Our mission is to provide quality services to different companies that demand advanced, innovative and reliable technical solutions. These solutions have been made and offered for operational services and R&D projects. Currently Solute is acknowledged as a premium quality CAE provider for many industries.

Figure 1 – *SOLUTE's fields of engineering services.*

1.1 AUTOMOTIVE

The broad experience amounted to in the automotive industry constitutes one of our main pillars of knowledge and our portfolio encompasses almost all competences available in the market.

The manufacturers and OEMs that have relied on Solute to aid their developments are located around the globe so different CAE programs, working structures, R&D concepts and structural solutions are kept inside Solute's know how.

1.2 DISCIPLINES

Solute has a wide experience in many different fields of simulation. Each of this fields are applied to the automotive industry for different disciplines.

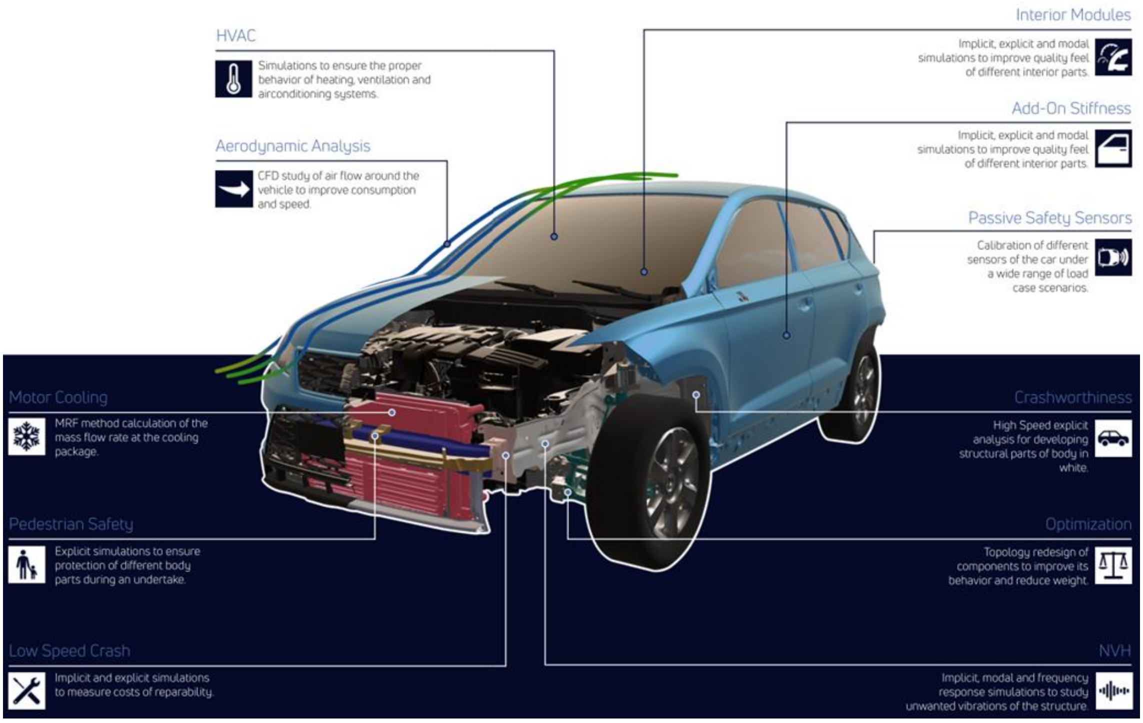


Figure 2 – *Automotive disciplines.*

PEDESTRIAN PROTECTION: The homologation process of each car must be safe not only for the people inside the car but also for pedestrian. In order to qualify a car as safe, the **Pedestrian Protection tests** measure levels of potential risk at injuries to the human body. The body parts that could be irreversibly damaged during an undertake at 40km/h are represented by three different tests:

• Lower Leg impact.

• Upper Leg impact.

• Head impact.

Figure 3 – *Pedestrian protection tests scheme*.

RCAR: Concepts such as damageability or reparability have started to be considered paramount features during the design phase of the vehicle. The aim is to provide solutions that minimize the extent and cost of the losses suffered in an accident and ease repairing. The costs of reparability after a low speed crash are key factors for insurance companies when establishing the rating of each car model with a degree of reparability and its associated insurance cost of the vehicle for the owner. Low speed crash tests try to represent a broader range of occurring impacts situations at different speeds and with different impact testing devices (AZT, RCAR Bumper and pendulums) that requires the complete vehicle simulation in FE.

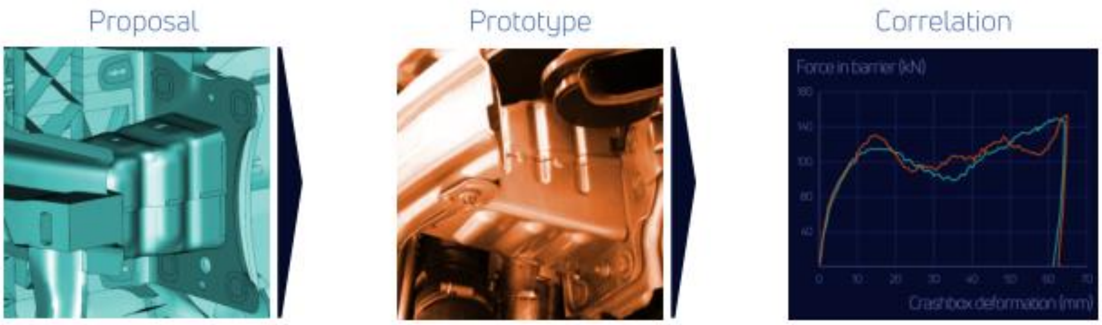


Figure 4 – *Crashbox development scheme*.

For RCAR test, the vehicle must withstand with no evidence of damage in the main structure. This is a complex issue due to the current compact vehicle design, where very little space is available for an effective energy absorption. The integration of the **crash management systems** (from now on CMS) into the overall concept of the vehicle represents a challenge for manufactures. Furthermore, the outer design specifications of the vehicle often determine the shape of the CMS among other elements.

2. INTRODUCTION

The CMS development has a great complexity due to the design constraints. On one hand, engineers have the challenge to develop a CMS that not only works for low speed crashes but also it must be compatible with other disciplines such as high-speed crash or pedestrian protection. On the other hand, the CMS must be friendly with the rest of the components: bumper, cooler, frontend, etc.

Focusing on the crashbox, its design must assure that after a low speed crash up to 16km/h, there is no visible damage in the vehicle body (for example folds or marks) particularly in rails.

Moreover, the displacement in rails must keep below a specific limit to ensure the CMS can be replaced easily after the impact. In certain circumstances, these criteria must be compatible with a limited crashbox compression level. This prevent an excessive intrusion of the cross-beam or the bumper and consequently helps to avoid damage in components such as the cooler or some liquid tanks.

In this scenario, it is a need to have at your disposal a tool that makes faster and simpler the crashbox design and optimization process. The development of a multi-objective process that consider all these aspects and variables represents a complex challenge.

In this paper, the way this tool helps to reduce the complexity of the process and how engineers can be helped to take decisions is presented. The crashbox will be designed for the AZT load case based on the RCAR (Research Council for Automobile Repairs test) conditions.

For this purpose, this tool must consider several input values, particularly incoming forces in rails (total and by each side), and crashbox deformation. Bear in mind, this tool will be used in the early stage of the vehicle development, and for this reason, the number of parameters is reduced and, for example, parameters like the displacements in rails will not be part of the study. Thus, the objectives are the following:

- Maximum energy absorption in the crashbox considering the force and deformation constraints.

- Crashbox must be manufacturable, which means, it must be easy to integrate into the overall concept of the vehicle.

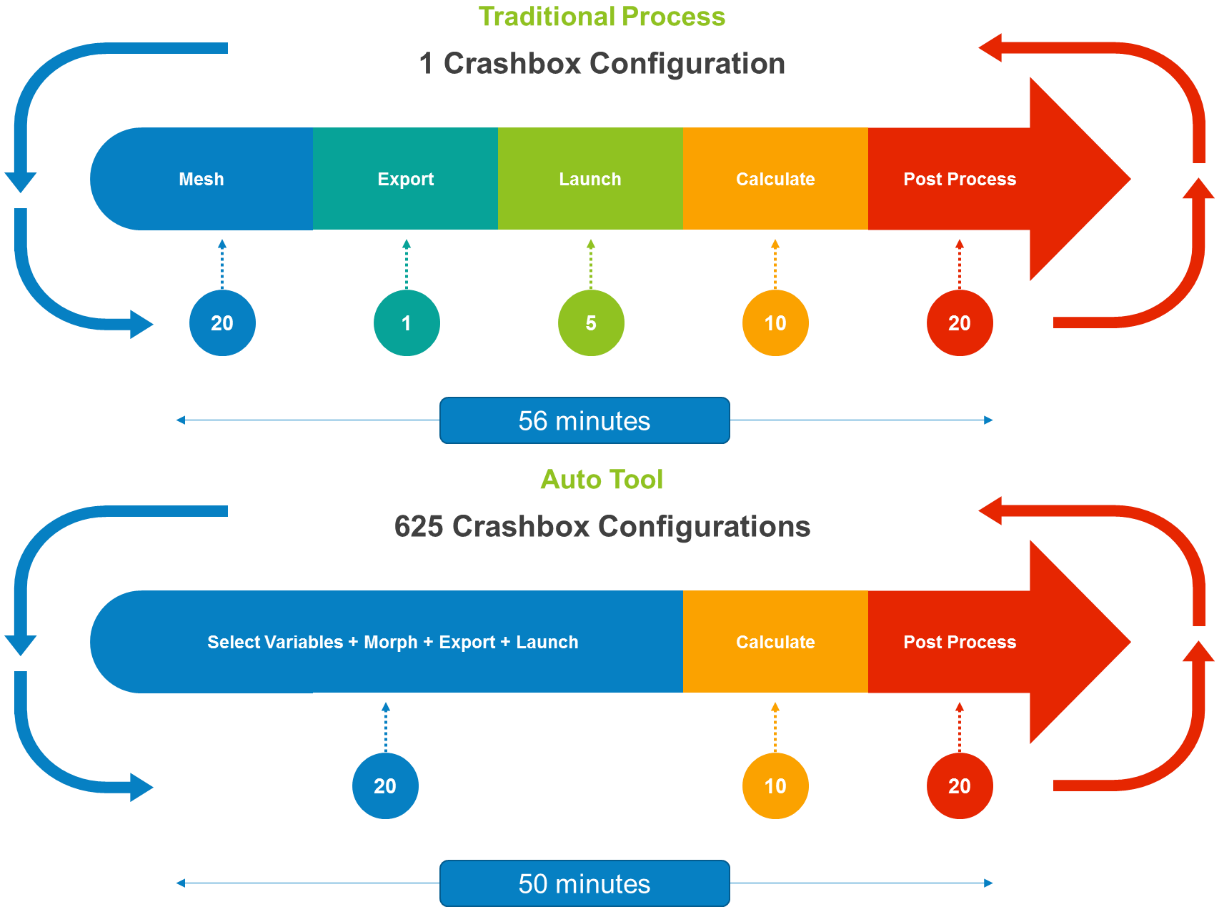
3. CRASHBOX DESIGN AND OPTIMIZATION TOOL.

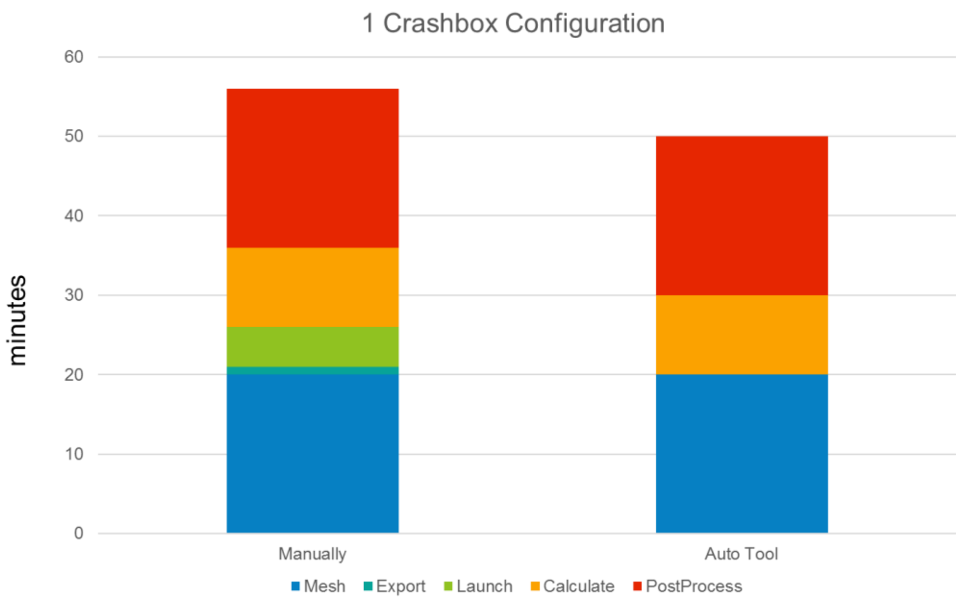
<u>Will the tool help to save up time?</u>

BETACAE offers a scriptable environment (in python and beta language) which offers the opportunity to automate processes. For this reason, the crashbox development process was analyzed in detail, and the points where the process can be automated were determined.

Thanks to the BETACAE scripting tool, it is possible to simplify tedious and exhaustive tasks into accurate and practical processes. One of the most time-consuming tasks is the generation of crashbox models. It is necessary to generate a great number of tests to design a crashbox with good behavior and within the requirements. With this automated tool, it is possible to generate crashboxes automatically and in a controlled way, using the **ANSA morphing tool** by scripting.

In like manner, the postprocess task is also a time-consuming process, as it is necessary to postprocess many models and manage a considerable number of parameters. The automated postprocess tools offer the capacity to get the desired values from each results file reducing the time of postprocessing and optimizing the process to avoid errors and duplicities.


Traditional Process
1 Crashbox Configuration
Mesh
Export
Launch
Calculate
Post Process
20
1
5
10
20
56 minutes
Auto Tool
625 Crashbox Configurations
Select Variables + Morph + Export + Launch
Calculate
Post Process
20
10
20
50 minutes


1 Crashbox Configuration
minutes
60
50
40
30
20
10
0
Manually
Auto Tool
Mesh
Export
Launch
Calculate
PostProcess

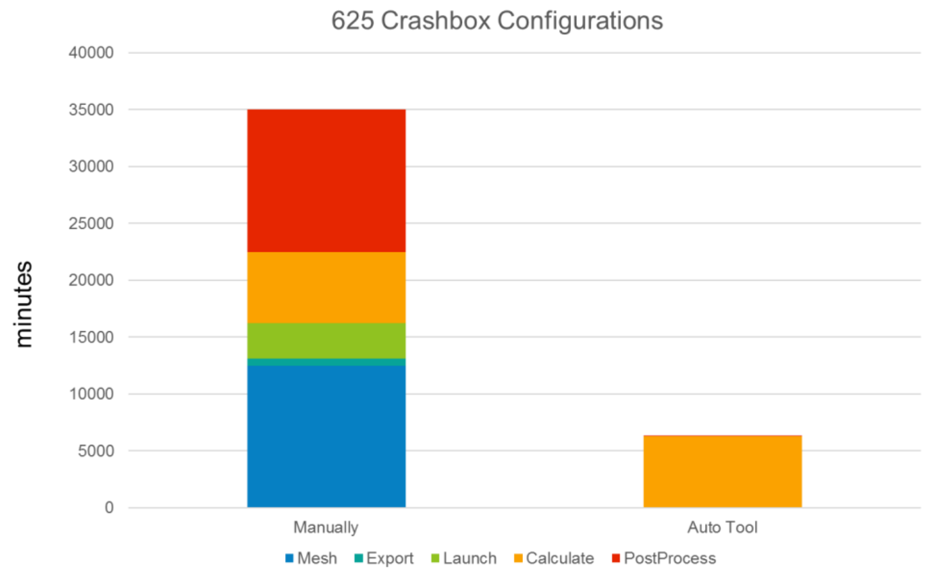


Figure 5 – *Time-saving scheme*.

To summarize, scheme above shows how the automatic tool can save time in this process.

4. BODY IN WHITE STIFFNESS ANALYSIS.

As it has been described before, the objective of the crashbox optimization process is to get the best crashbox behavior and avoid damage in the vehicle body. With this intention, previously to optimize the crashbox, the car body in white is tested to obtain the maximum incoming forces that the car body can withstand without suffering any visible damage. Also, the most suitable incoming force in each rail side can be determined. These values are obtained by means of a body in white stiffness analysis.

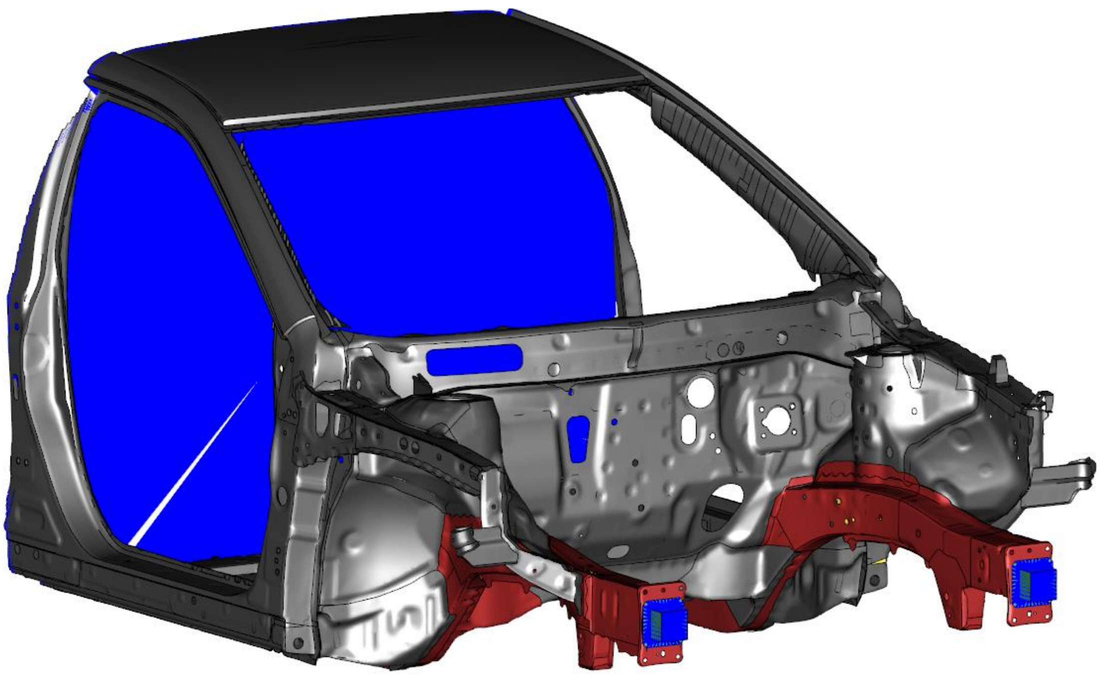


Figure 6 – *Body in white stiffness analysis example.*

The test is made with an explicit analysis, where a force on the rail is applied by means of a stamp centered in the rail plate (See figure below).

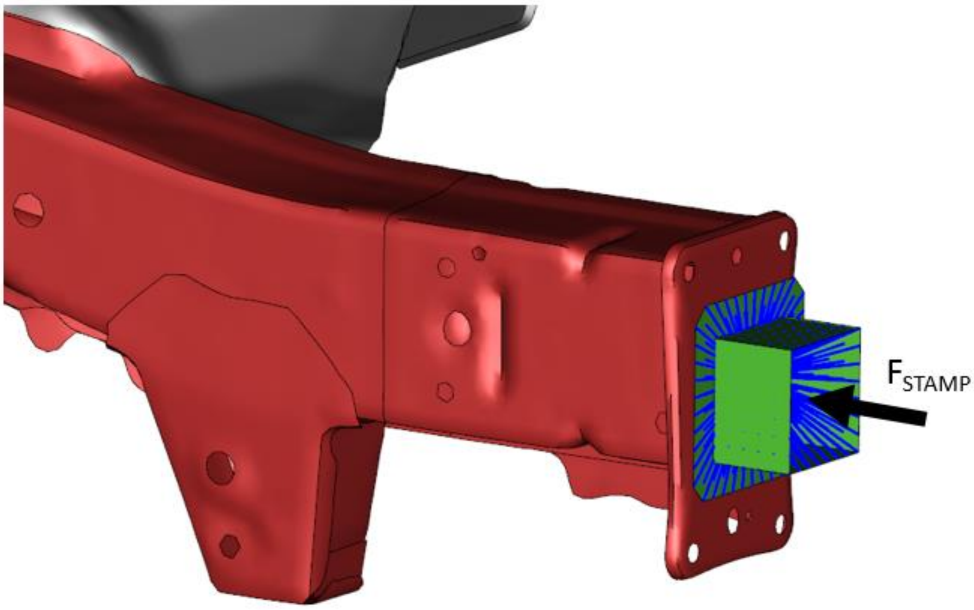


Figure 7 – *Force application on the stamp.*

Increasing the force level step by step, the maximum force can be reached. Also, the incoming forces by each rail side (upper, lower, inner and outer) can be obtained varying the position of the stamp. In this way, the best position or, in other words, the most suitable incoming forces can be obtained to avoid damage in rails and vehicle body.

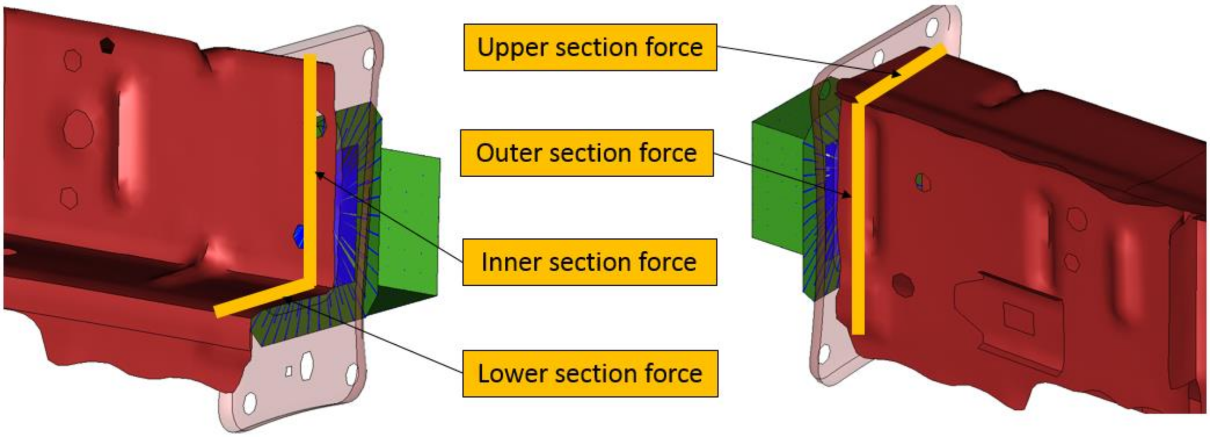


Figure 8 – *Section forces defined in the model scheme.*

These parameters, beside the crashbox deformation, are used as control points by the tool to design and develop the most optimum crashbox.

5. MODEL DESCRIPTION.

To help in the tool development process a a low computational cost model was created to analyze efficiently the crashbox behavior against low speed impacts.

Several models were under study, but finally, a **lumped model** was chosen since it can represent the crashbox and cross-beam with all the details while it simplifies the rest of the vehicle. It is justified given the need to select a model where the main object of study, the crashbox, is modeled as realistic as possible and with a low computational cost to analyze as many configurations as possible.
Moreover, the mass distribution allows the effects of asymmetry load case to be captured, which makes the model suitable for the load case to analyze, the AZT (10° Frontal impact

with a rigid barrier RCAR test). At the same time, the model is suitable for symmetric load cases.

Next figure shows a scheme with the concept of the model proposed.

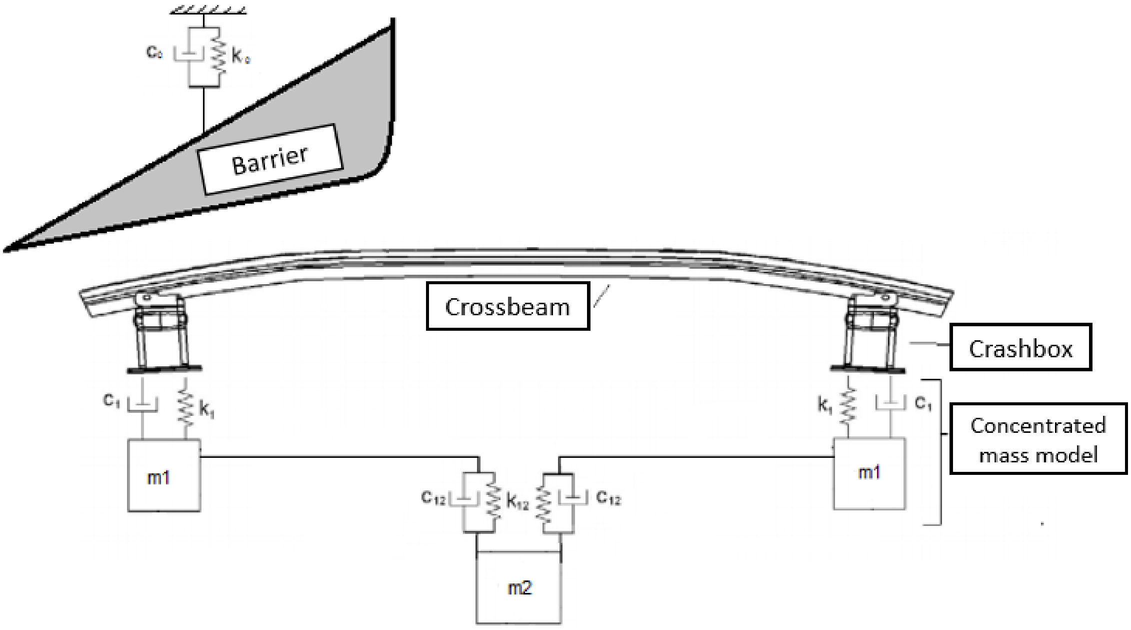


Figure 9 – Lumped model scheme.

In this lumped model, the barrier impacts directly the cross-beam, while in a real vehicle the bumper and other devices are placed between the cross-beam and the barrier. For this reason, with the aim of generate a model as accurate as possible, a non-lineal spring was used to simulate the significant effect of the front bumper.

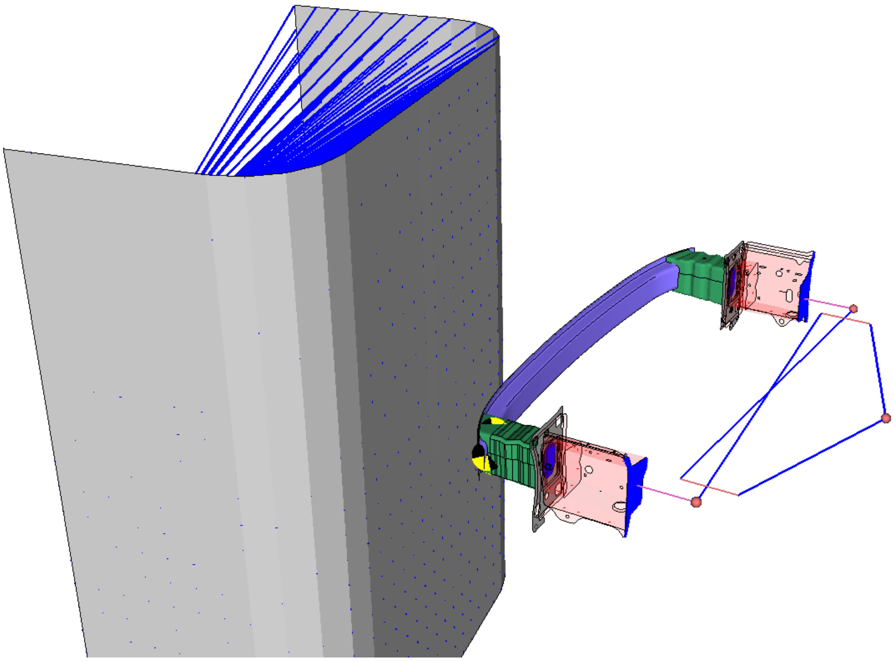


Figure 10 – *Lumped model, general view.*

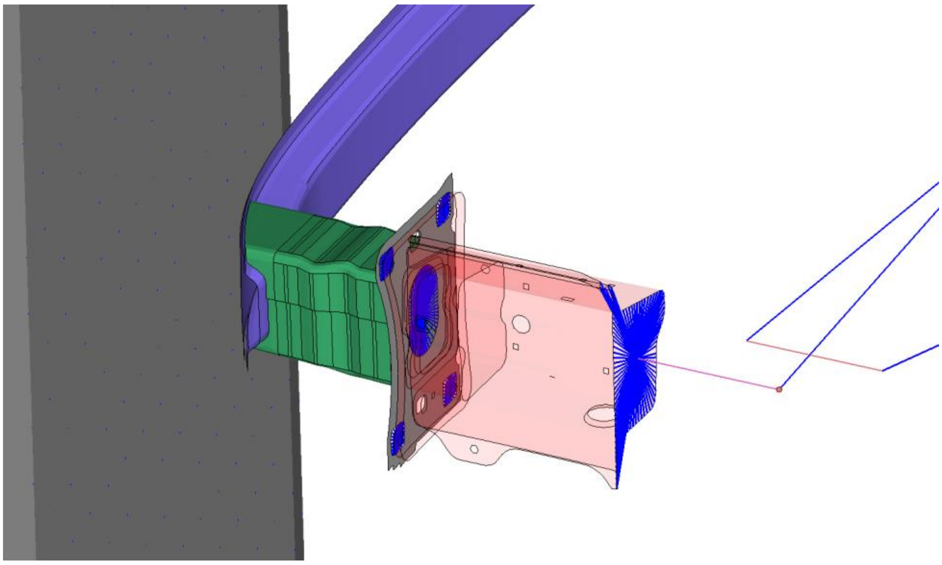


Figure 11 – *Lumped model, crashbox detail view.*

The model was built in ANSA, and PAMCRASH was used as a solver.

4. TOOL FUNCTIONS.

Optimization approach: The proposed design and optimization process is divided in two stages. In the first stage, the number of beads is determined to comply with the rail force and crashbox deformation limits. This step leads the engineers know the number of bead necessary and even the height of each bead.

In the second stage, the distribution of beads among the crashbox will be studied. The crashbox not only have to comply with the two first requirements, but also must achieve a goal force in each side of the rail to assure the best behavior and to avoid damage in the vehicle body (rails). In this way, the tool will help to develop a crashbox with a constant and controlled deformation.

Geometric variables: Several geometrical variables are used to develop the different crashbox configurations. These parameters are defined by the user. Any variation of one of these parameters makes a new crashbox configuration.

The geometrical variables defined in the tool are:

- Thickness
- Beads number
- height bead
- material
- deformable space

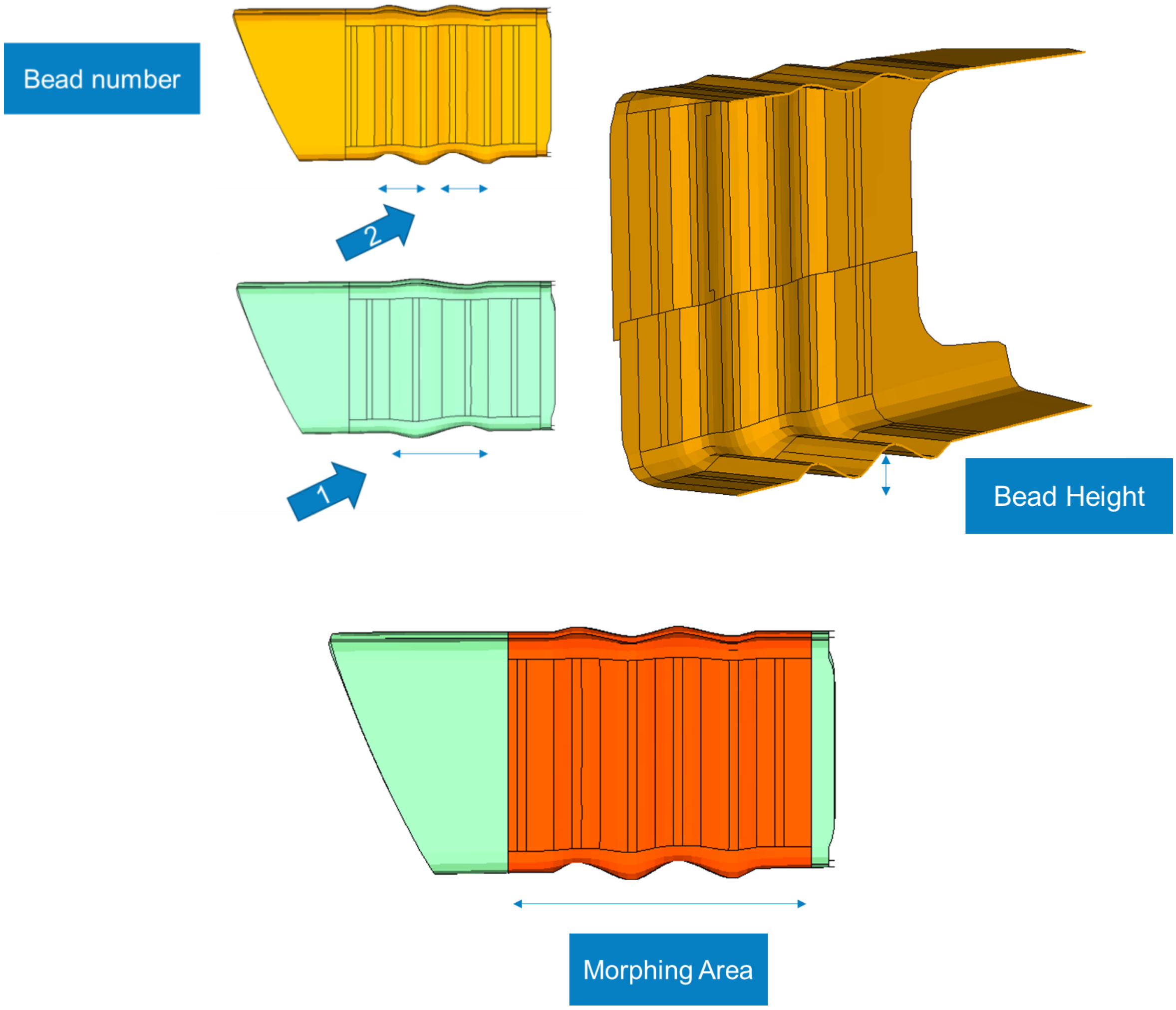


Figure 12 – *Geometric variables.*

Input variables: A maximum force value is defined according to the data provided for the stiffness analysis. Also, recommended force values in each side of the rail are used in the postprocess. For that purpose, the maximum peak forces are analyzed for each model:

A) Incoming force in each side of the rail (Upper, lower, inner, outer)
B) Total incoming force in rail (sum of the four forces)

Moreover, a maximum intrusion is defined. This intrusion is defined considering the minimum distances to components like liquid tanks or cooler among other.

Decision and ranking variables: On one hand, the peak forces and the average forces are influential parameters in the crashbox crushing, since they are directly linked to the crashbox folding.

The **crush force efficiency (CFE)** is a measure of the incoming rail forces variability. If the values are close to the unit, it means that the crashbox is folding with an average force similar to the maximum force, which means, the changes in the crashbox folding are mitigated. This is highly suited in the crashbox design. Hence, the closer to the unit the better crushed force efficiency.

Mathematically, it can be expressed as:

$$\text{CFE}= \frac{F_{avg}}{F_{max}}$$

Apart from this parameter, the crashbox energy absorption is analyzed. Similarly, the greater efficiency value the better crashbox design, since less amount of material is required to absorb the same amount of energy.

It can be expressed as:

$$\text{Efficiency}= \frac{\int_0^d F(x)\cdot dx}{F_{max}\cdot d_{max}}$$

On the other hand, the crashbox can be folded axially on itself (crushed) or bending laterally (lateral bending)

Figure 13 – *Examples of crashbox deformation, crushing (left side) and lateral bending (right side).*

The first deformation mode is advisable in order to take the full advantage of the crashbox and get high CFE and Crashbox efficiency levels. However, the crashbox can behave in a bad manner if the crashbox is very lean, or the crashbox is very inflexible and certain lateral forces are acting.

For this reason, the bending angle is also analyzed. This parameter measures the difference between the deformation in the center of the crashbox with respect to the sides (inner and outer).

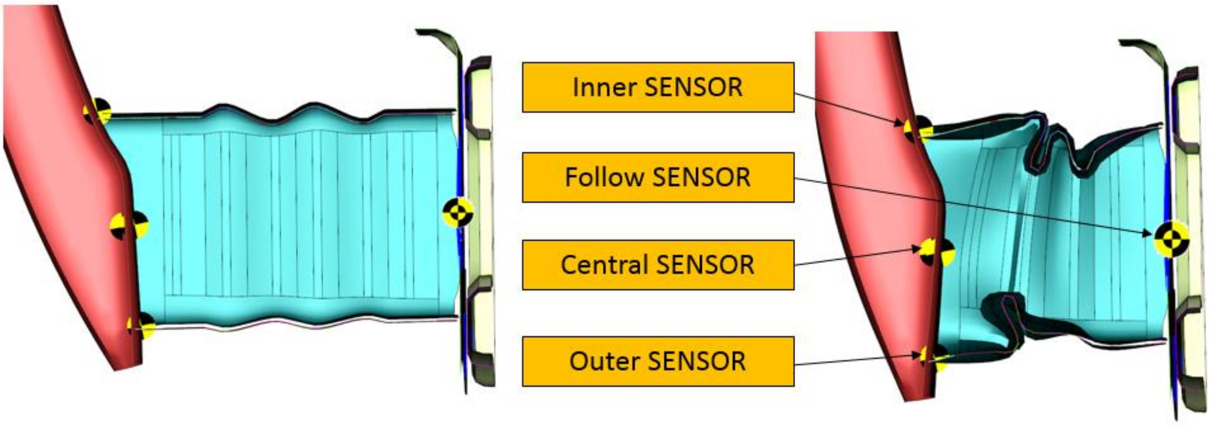


Figure 14 – *Crashbox sensors scheme.*

Crashbox generator: Using the morphing Deck by python scripting the non-static variables changes for each configuration and the geometry is adapted. The geometry of the CMS is exported (.inc), a launcher is modified (.pc) and the analysis starts to calculate in our servers.

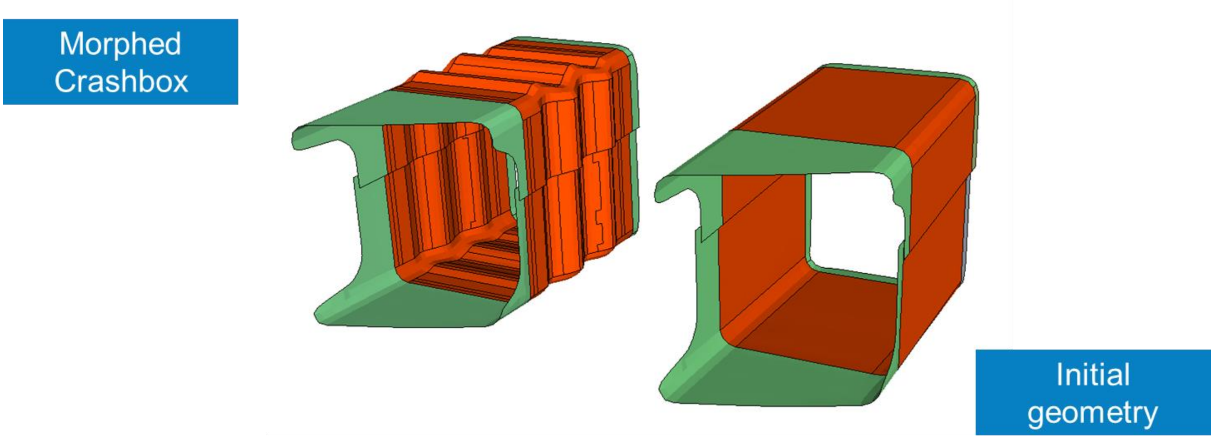


Figure 15 – Morphing process.

Postprocessing: Once the models have been calculated, next step is to postprocess the results files. In this step, the maximum forces in rail, the crashbox deformation and the decision/ranking variables described before are obtained.

With this aim, some scripts in Python (and beta language) have been developed to offer a worthwhile solution to postprocess many results files and to get a comparison among different results.

The **first script** makes possible to postprocess with promptness and agility a large number of data files (.THP in this case) getting all the parameters outlined before.

After running the script, an **excel** with all this parameter is created (Total force, upper force, lower force, inner force, outer force, crashbox deformation, efficiency, CFE and bending angle). These parameters are listed per model as it can be seen in next figure.

Model	Total Force	Deformation	Eficciency.	Upper Force	Upper CFE	Lower Force.	Lower CFE.	Inner Force.	Inner CFE.	Outer Force.	Outer CFE	Bending angle
1537	135.93	46.71	82.65	31.56	0.70	31.20	0.71	34.29	0.62	74.67	0.43	8.41
1540	155.35	37.16	82.45	37.46	0.68	38.04	0.65	44.31	0.62	75.81	0.36	16.43
1536	148.04	44.65	82.12	39.19	0.66	36.78	0.67	48.73	0.56	71.19	0.32	16.60
1539	143.54	47.75	80.22	32.38	0.68	28.01	0.61	44.81	0.64	77.35	0.42	7.33
1535	143.91	44.33	79.73	39.14	0.64	32.30	0.59	54.17	0.61	69.51	0.30	13.83
1529	139.17	47.15	79.24	31.26	0.66	30.80	0.64	49.59	0.52	76.35	0.40	8.40
1533	148.43	48.31	77.94	33.37	0.64	27.45	0.62	52.04	0.62	77.50	0.39	7.36
1534	153.63	46.71	77.85	45.18	0.61	33.03	0.64	49.87	0.61	58.38	0.37	13.02
1538	149.37	43.67	77.08	38.27	0.66	39.87	0.61	34.91	0.72	78.21	0.30	9.11
1531	152.44	44.47	76.03	37.70	0.64	35.59	0.63	52.25	0.59	80.34	0.27	9.67
1530	149.25	48.96	71.32	35.72	0.64	31.75	0.64	51.46	0.53	75.83	0.33	10.68
1532	151.40	49.03	70.80	37.21	0.69	30.41	0.67	49.51	0.56	74.17	0.29	15.08
1528	136.27	52.28	67.59	39.50	0.66	40.99	0.63	36.75	0.60	41.01	0.48	6.19
1527	145.80	56.12	66.09	57.05	0.39	40.76	0.47	63.17	0.55	43.74	0.37	5.98

Figure 16 – *Results written in the excel view.*

Moreover, this script creates some graphics from the above parameters. The curves generated are:

- **Rail force vs crashbox deformation (with the efficiency value on the legend) curve:**

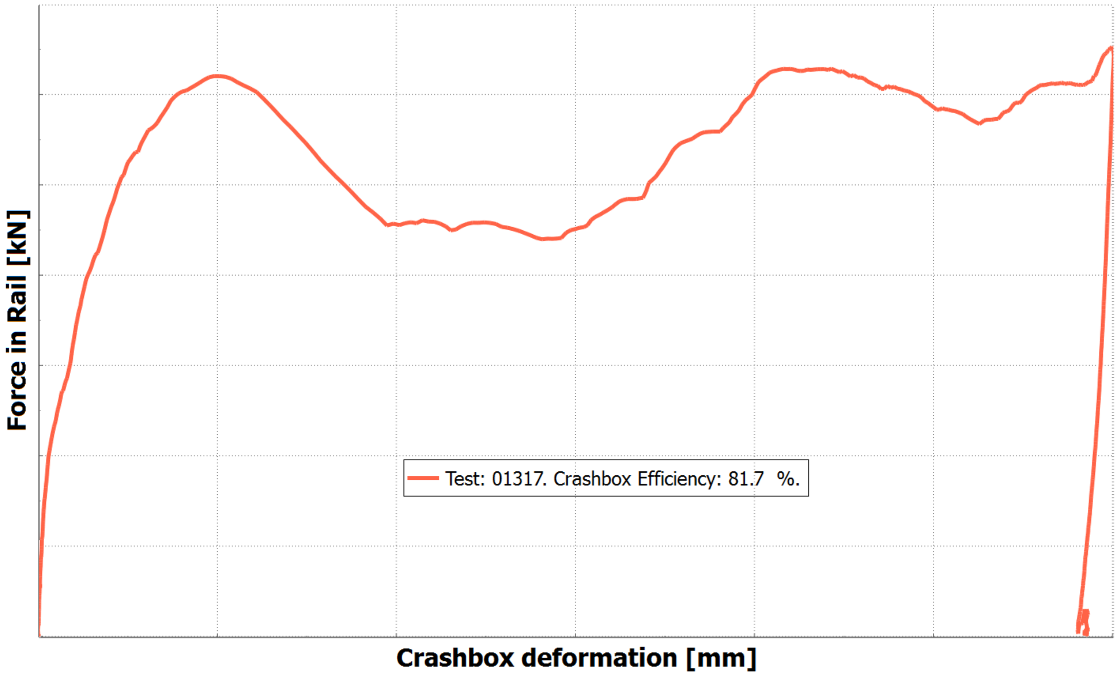


Figure 17 – *Example of force vs crashbox deformation curve.*

- **Force (in each side of the rail) vs time (with the CFE value on the legend) curve.**

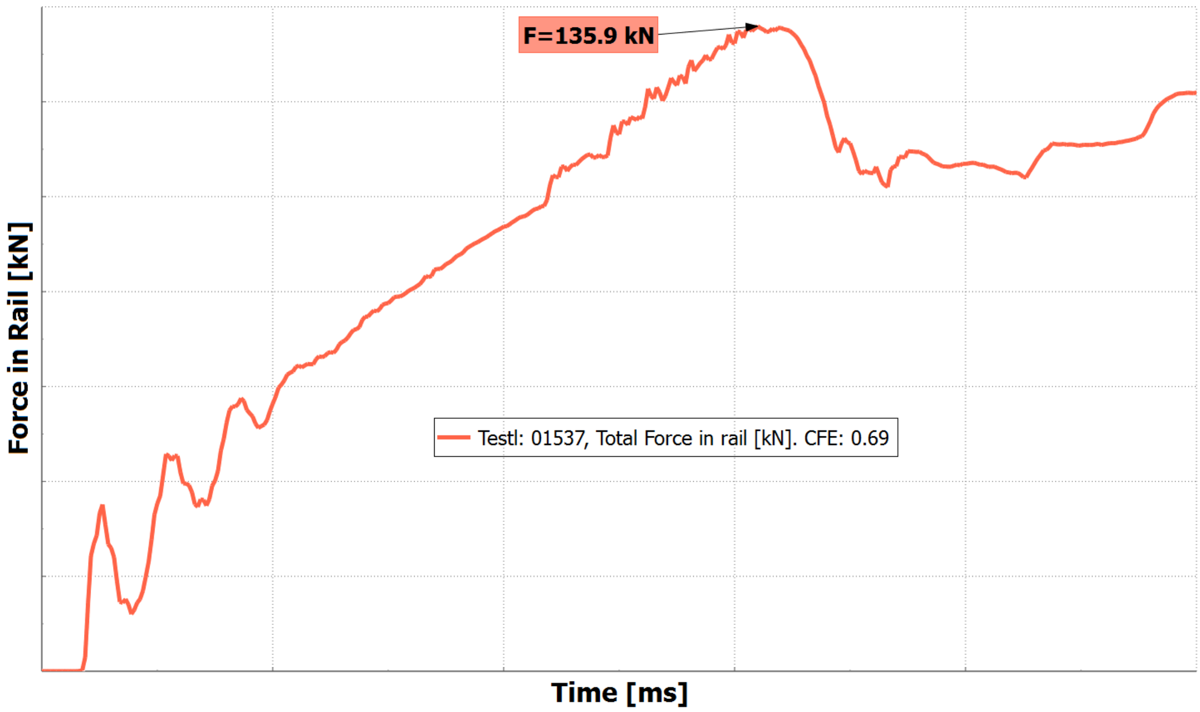


Figure 18 – *Example of force vs time curve.*

- **Crashbox deformation vs time curve.**

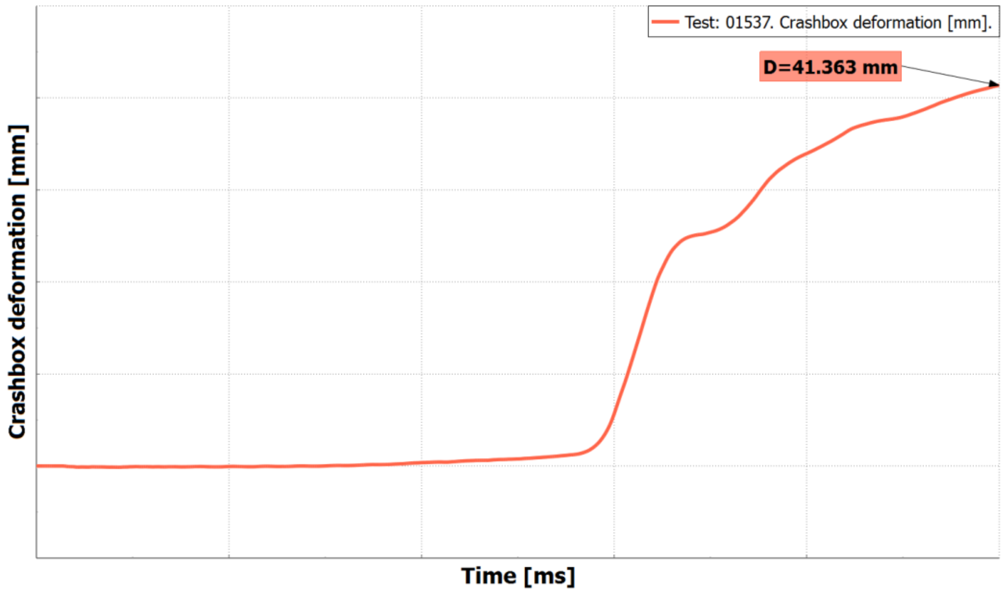


Figure 19 – *Example of crashbox deformation vs time curve.*

All this information will help engineers to select the best choice. For example, the information listed in the excel can be ranked by efficiency or CFE. In this way, the information can be analyzed to distinguish which proposed crashbox works better. In other words, a crashbox that crush progressively, in a controlled manner and with a good efficiency can be chosen.

Apart from this tool, a **second script** has been developed to compare different models. Once all the proposed crashbox have been analyzed with the first script, the ones with better behavior can be compared graphically.

This second script shows the curves listed below to facilitate the decision-making process.

- **Rail force vs Crashbox deformation (with the efficiency value on the legend).**

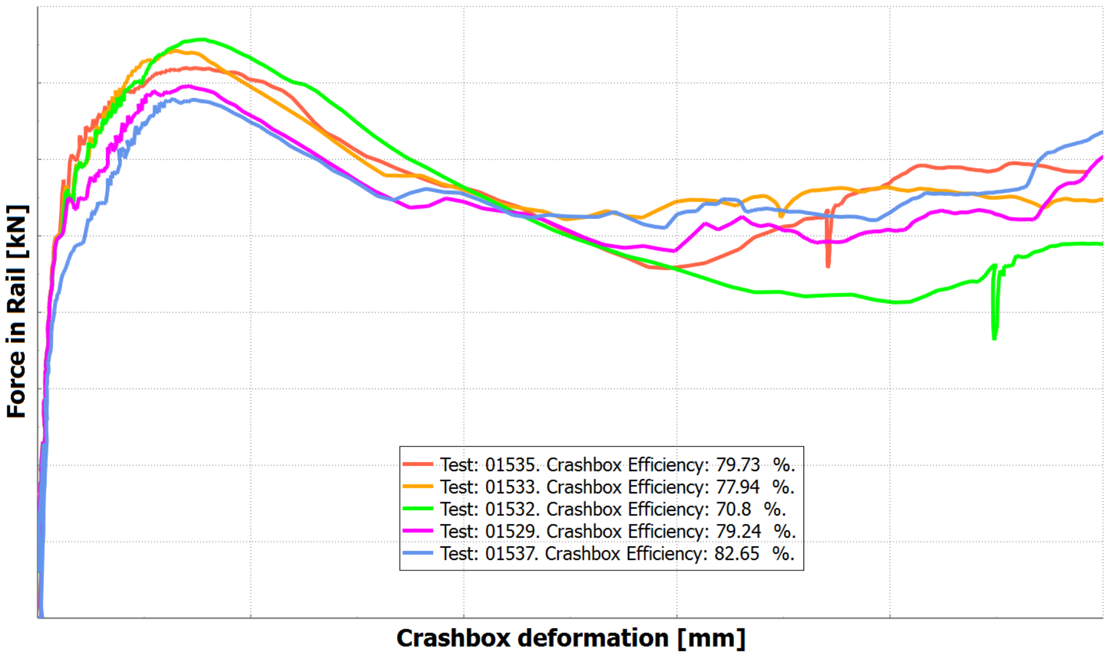


Figure 20 – *Example of force in rail vs crashbox deformation curves comparison.*

This comparison serves not only to choose the crashbox with the top efficiency but also a crashbox with a good behavior.

- **Rail force vs Time (with the CFE value on the legend) curve.**

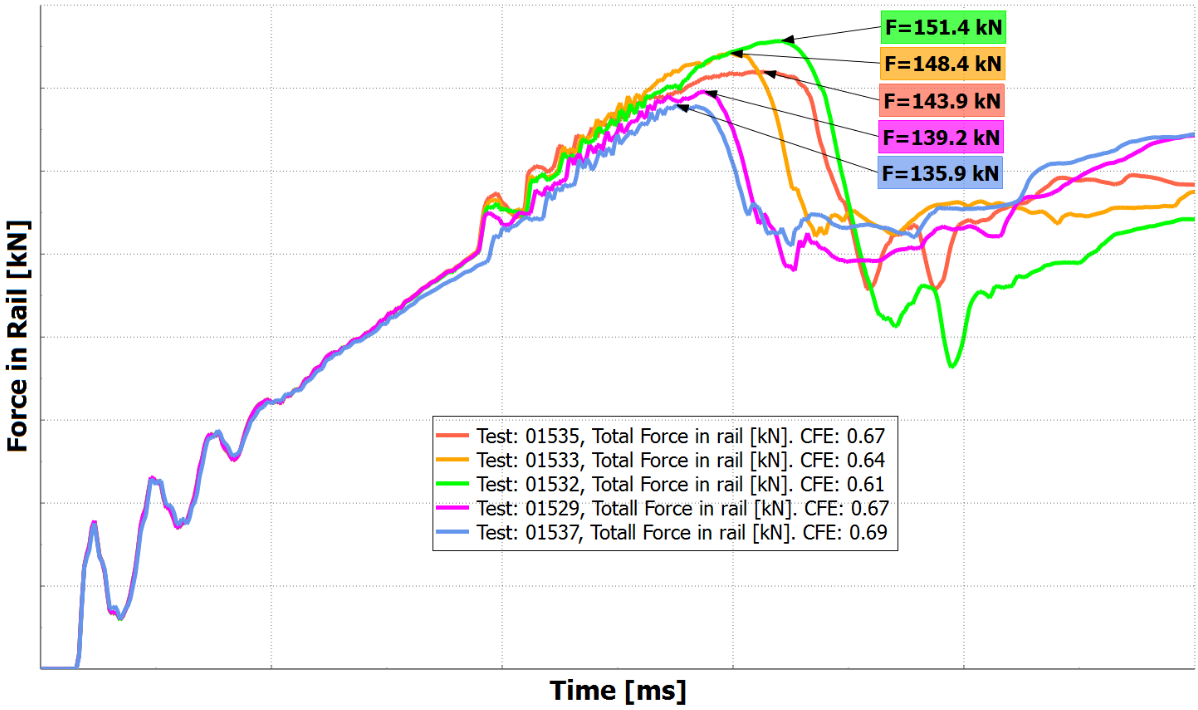


Figure 21 – *Example of total force vs time curves comparison.*

- **Rail forces (per side) vs Time (with the CFE value on the legend) curve.**

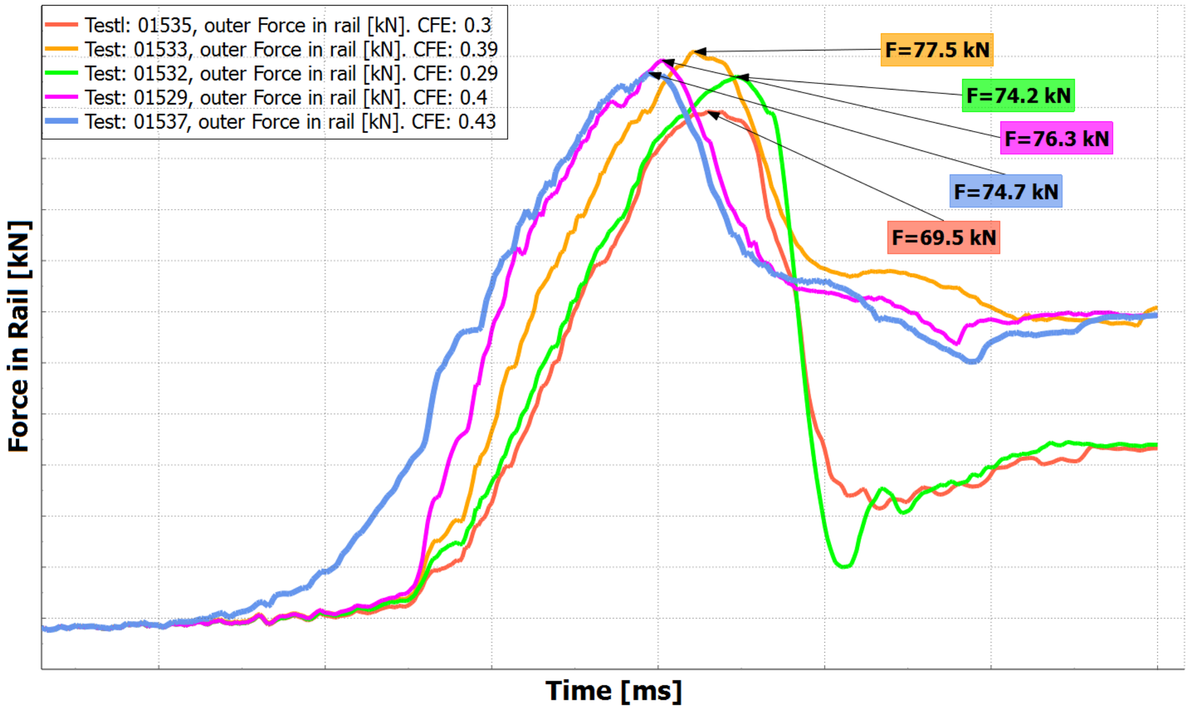


Figure 22 – *Example of outer vs time curves comparison.*

The crashboxes that better fulfill the force goal in each side of the rail can be chosen from these curves.

- **Crashbox deformation vs time curve.** To see which ones better fulfill the deformation goal in each side.

X. SUMMARY AND CONCLUSIONS

Most of the tasks of a CAE Engineer are repetitive and very time-consuming. Thanks to the scripting capabilities of ANSA and META engineers can save a huge amount of time to focus on analysing results and improve their designs.

The crashbox generator by Solute is an example of python scripting language application for ANSA that has allowed to accelerate the early dimensioning of this vehicle part.

Just selecting some static variables, the tool generates all configurations available, launches the analysis and extracts relevant data. After every configuration has been run, a database is created, and the results are ranked by the established criteria (Efficiency, CFE, etc.).

If the results are closed to the objective, the sensitivity process begins to analyse other geometric variables to generate even better configurations.

In future release of the tool, we will add variables of study to offer more advanced solutions to our clients in record time.

REFERENCES

(1) ANSA version 17.0.0 User’s Guide, BETA CAE Systems, July 2015

(2) P. Jonsén, E. Isaksson, K. Sundin, and M. Oldenburg, Identification of lumped parameter automotive crash models for bumper system development, International Journal of Crashworthiness, 14 (2009), pp. 533–541.

(3) N. Tanlak and F. O. Sonmez, Optimal shape design of thin-walled tubes under high-velocity axial impact loads, Thin-Walled Structures, 84 (2014), pp. 302–312.

(4) N. Tanlak, F. O. Sonmez, and M. Senaltun, Shape optimization of bumper beams under high-velocity impact loads, Engineering Structures, 95 (2015), pp. 49–60.