

# An automated belt fitting tool for 6 and 10 year-old child crash dummies

Gillian Mara, Christian Dalton

Jaguar Land Rover, Semcon

## 1 Introduction

Modelling child occupant safety through computer aided engineering (CAE) within a vehicle is an area that is constantly developing. In 2003, Euro NCAP introduced a child occupant protection rating to inform consumers the results of vehicle safety assessments [1]. These assessments used 18 month and 3 year old dummies in frontal and side crash. Developments in technology and crash dummies have led to the introduction of new larger child dummies in the Q series, those being the 6 and 10 year old. In 2016-18 it is proposed these will be used in frontal and side EuroNCAP consumer crash ratings, where the dummy will be sat in a child seat. In the competitive market there is a demand on manufacturers to produce cars in a shorter time frame from idea to production, therefore making the overall process more efficient and cost effective. One area that this places emphasis on is predicting how cars perform where it has the potential to reduce the need to produce as many prototype vehicles. To ensure the results are accurate the predictive method of finite element analysis (FEA) is very dependent on the inputs, where those are primarily grouped into materials, geometry and boundary conditions. In terms of the child occupant prediction this means obtaining realistic inputs to gain realistic output kinematics and injury criteria during the frontal or side crash. The belt restraint system is one of the most predominant in affecting the child kinematics and one that needs to be modelled accurately. This paper will describe an automated seat belt fitting method using the LS-DYNA® package that is efficient and effective in creating a belt geometry and path for a 6 and 10 year old dummy in a child seat.

The path the belt has to take from the retractor to buckle head and then to the anchor point on the car, has to undergo a number of obstacles when a child seat is placed in row 2. Such obstacles include, a clip typically attached to the headrest of the child seat, under the arm of the dummy, and around the armrest on the child seat to the buckle tongue. In reality, fitting the belt around a child in a child seat can be cumbersome and awkward. Replicating it in a CAE environment requires mesh and material properties that can handle these obstacles accurately and where possible applying no extra stresses or strains into the model. The belt path also needs to be able to feed freely through and around these obstacles, ideally achieving a mesh with elements that have the same length across the width and length of the belt.

Using the current primer belt fitting tool (defining the path and mesh then fitting the belt) generates unsatisfactory geometry where many elements are irregular and the path of the belt is extremely dissimilar to reality, shown in Fig 1. The irregular elements create uneven tension across the belt as the curved geometry is loaded during the crash. The method also struggles to generate a close fit path around the increased number of obstacles on the child seat. This then has an impact during the crash as the excess is tightened as the dummy moves forward in the crash.

An adaptation to the tool is where the user performs an external tensioning simulation. This achieves slightly better results through the belt being pulled from the retractor and anchor point ends. However, both methods are dependent on selection of reference points and individual judgements in editing x,y,z coordinates of points and modifying angles between points. Often the definition of the path has to be attempted several times to obtain the best fit as well as create extra webbing resistance parts defined as beams to try and overcome the curved geometry during a crash. Overall, the whole procedure requires a large number of manual operations, which is time consuming and increases the opportunity for error and/or variability to be introduced. Also if you had a new environment i.e. a different car, seat position, new dummy, the whole process must be restarted from the beginning as

the belt path definition is deleted during creation. Therefore, there is a need to try and create a repeatable and less user intense simpler belt fitting method that can produce realistic fitted belt paths that reduce user variability.

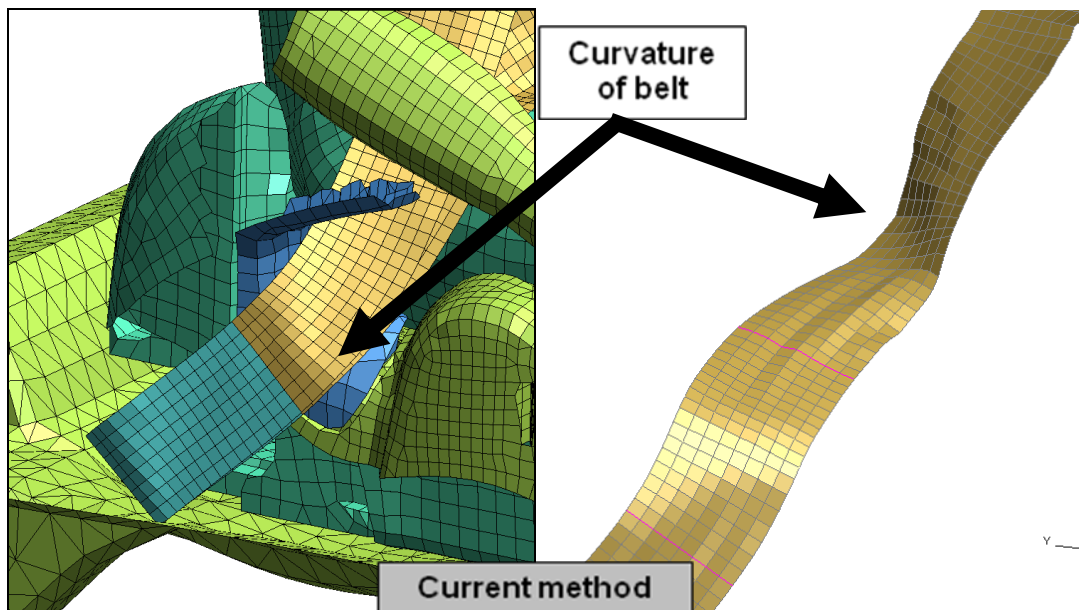


Fig. 1: Final belt geometry showing irregular elements at buckle head and from head restraint clip to retractor

## 2 Method

The criteria for the belt fitting tool would be one that is a simplistic process, creates repeatable belt paths for the same environment, easy to recreate a belt path if a new environment is required and the final geometry shape produces undistorted elements. To generate such a method we must first understand the process in reality. To fit a belt in reality, the belt already exists and is pulled out from the retractor and is navigated around the obstacles and then any slack is taken off through pulling the belt tight. This pulling process typically pulls the belt from the anchor point to the buckle head and then the buckle head to the retractor. This concept can be the basis for an automated belt fitting tool.

The advantages of the current method is that it is already a part of PRIMER, whereby primer takes the user through a number of steps, these are, defining the belt path, define mesh size and material type, defining and assigning contacts and then doing a fit to tension the belt. Combining the processes in reality and what needs to be defined in the CAE environment provide the fundamental requirements for the automated fitting tool. Using the idea that the belt already exists and is navigated around the child and child seat and is then pulled tight, in reality the human gets rid of slack from the anchor point to buckle head and then from the buckle head to the retractor, forms the different stages of the method. These stages can be described as, creating the environment, defining the final positions e.g. retractor, buckle tongue and anchor point, then the fitting simulation. This would reduce the need for the user to define mesh and material types, and define the belt path. This in itself would produce more repeatable and consistent belt fits as it would no longer rely on manually choosing points for the path definition.

In brief to create the environment, the user positions the child seat and dummy in the vehicle. They then extract the items necessary for the belt fitting from the complete vehicle model; child seat, dummy, rearseat base and back, geometry for anchor, retractor, d-loop etc. They then rigidise the environment and write it out as an include file, to be used within in a master include which incorporates the belt fitting include. The already existing belt is moved approximately into the right position. The extra shapes in the belt fitting include need to be positioned using the translate-rotate function within primer, those being the seat shapes, shoulder guide and arm shapes need to be positioned using the translate-rotate function within primer. Contacts need to be defined (belt to dummy and belt to seat back). The second stage of defining final positions of the belt system components is assisted through a script, which collects the coordinate information of specific nodes and then labels them as parameters. The script automates collecting the initial position of the anchor

point, tensioning end point, retractor, buckle and then final positions of each. The user then writes out and runs the belt fitting model and once run a fitted belt is generated. The user then can export the node positions of the fitted belt and buckle and paste them into a final belt model which can then be attached to the rest of the car.

During the simulation, the parameters that were defined and generated in the script, allow the belt to navigate around the child seat and dummy and finish with the belt passing through buckle tongue and retractor positions. The slack is taken away by pulling the belt from the anchor point and retractor ends. An example belt fitting will now be gone through in order to explain each of the stages.

The belt fitting tool file has a number of already defined components in it. For the purpose of this example a Britax KIDFIX seat and Q6 dummy was used. The user needs to define the environment and extract the surfaces of the dummy's torso, arms, and neck and the child seat, ensuring the arm rests, head restraint clip are included and the rear seat and write them as an include. The seat shapes, arm shapes and shoulder guide, shown in Fig 2, are orientated to match their final positions that are set by the car environment. The contact definitions need to have the correct part identification numbers assigned to them to ensure the belt to dummy contact and the belt to seat are correct.

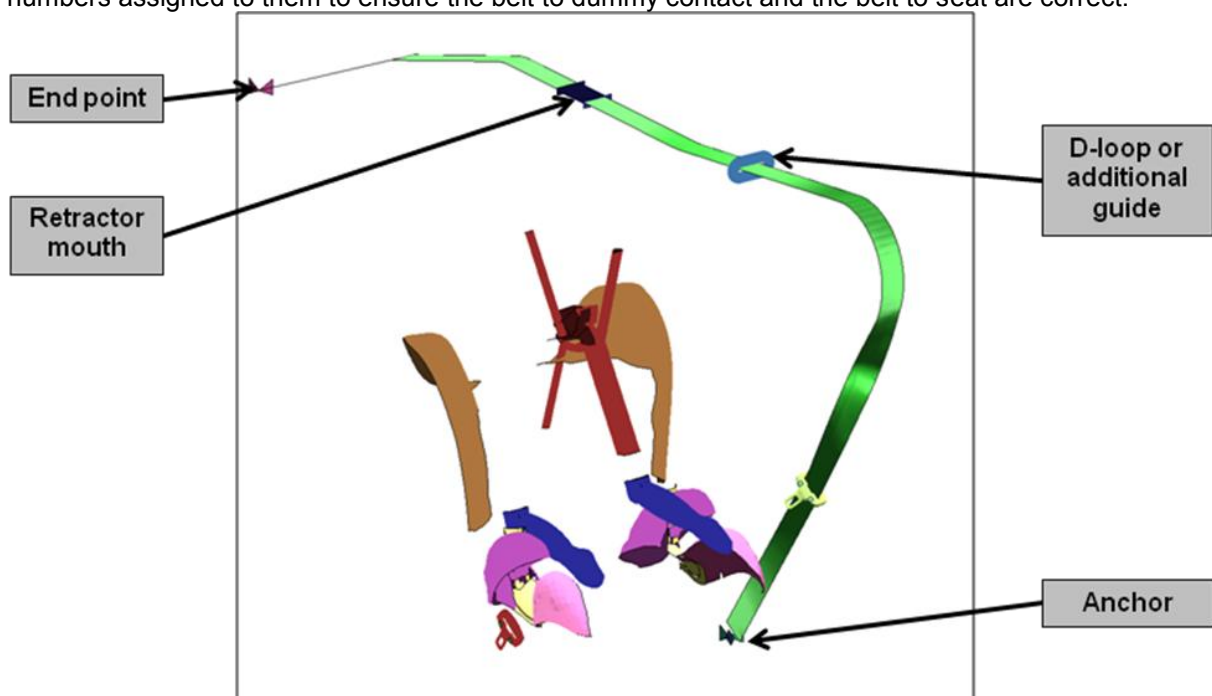


Fig. 2: Components in the belt fitting include file

In the brief explanation of the method, it stated a number of user defined points needed to be generated as parameters. The initial and final positions of the buckle tongue, retractor anchor point and tensioning end point all need to be defined, this is required so that the processes that are carried out in the simulation take the belt from the initial belt position as shown in Fig 2 to the final fitted belt shape. However, all of the parts are rigid apart from the belt to increase efficiency of the run time of the simulation. A rigid body by definition can be described as a singular point and if one wishes to translate a singular point to a specific finish position, it is trivial. If one wishes to rotate, one must define and know the order the rotations need to take place. This can be problematic and require complicated mathematics to try and orientate if the initial position and final translational position is known, but the angles of rotation are unknown. To get around this problem, a simple solution was found and used the mathematical principal of isometry [2]. In order to ensure the buckle tongue went from its initial position (green buckle tongue in Fig 2) to its final position (red buckle tongue) three small deformable patches were put on the buckle tongue, each that had one node that was not coincident with the rigid nodes on the buckle tongue, as shown in Fig 3.

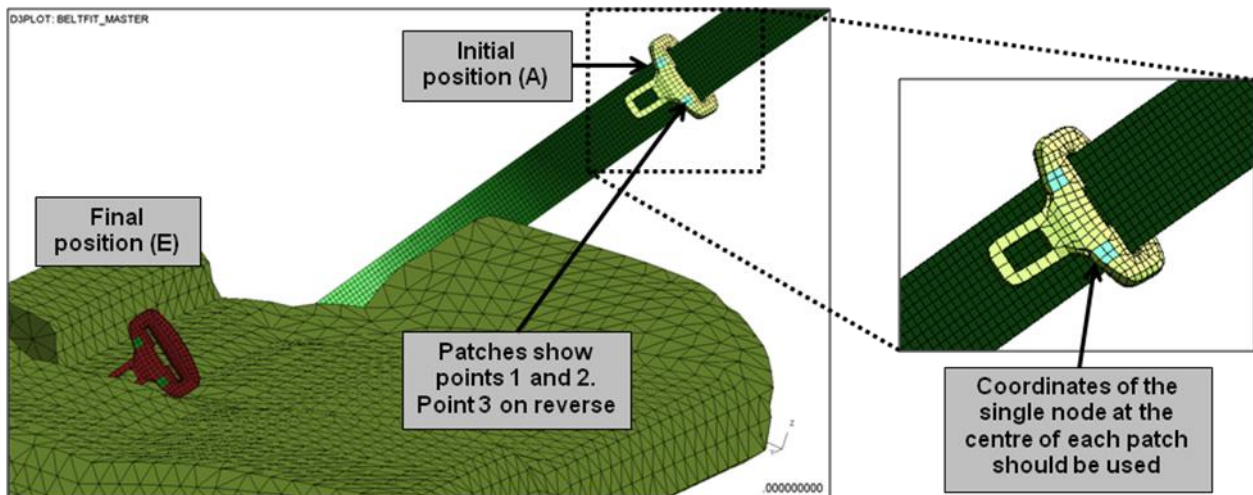


Fig 3: The deformable patches on the buckle tongue used to guide the rigid body from its initial to final position, providing the user with control in all six degrees of freedom

The user through the script clicks on each of the three nodes and the coordinates for each are generated as parameters. Calculating the translations and rotations is now trivial as through a vector displacement the orientation of the rigid body can be achieved. The minimum amount of information that is required to perform this orientation is Point 1  $A(x,y,z)$  Point 2  $B(x,y)$  and Point 3  $C(x)$  as the rest can be calculated through the following equations and the relationship between the lengths of the sides of each of the points.

$$l_{AB} = \sqrt{((A_x - B_x)^2 + (A_y - B_y)^2 + (A_z - B_z)^2)}, \quad (1)$$

$$l_{AC} = \sqrt{((A_x - C_x)^2 + (A_y - C_y)^2 + (A_z - C_z)^2)}, \quad (2)$$

$$l_{BC} = \sqrt{((B_x - C_x)^2 + (B_y - C_y)^2 + (B_z - C_z)^2)}. \quad (3)$$

This means the user has to define the three points on the buckle tongue at its initial and final position. Rather than the user clicking each of the points, which can be time consuming and has a higher chance of human error being induced, a script was created that identifies each of the initial and final positions and writes them out as parameters. The retractor mouth and D-loop uses the same process as the buckle tongue as it too needs to be translated and rotated. In this example the D-loop was not used. The nodes for the anchor point and tensioning end point initial and final points can also be specified through the script. Finally the user needs to specify a point on the dummy, O, which identifies where the dummy is in relation to the belt, shown in Fig 4. This is because the belt during the fit stage does not take a linear path.

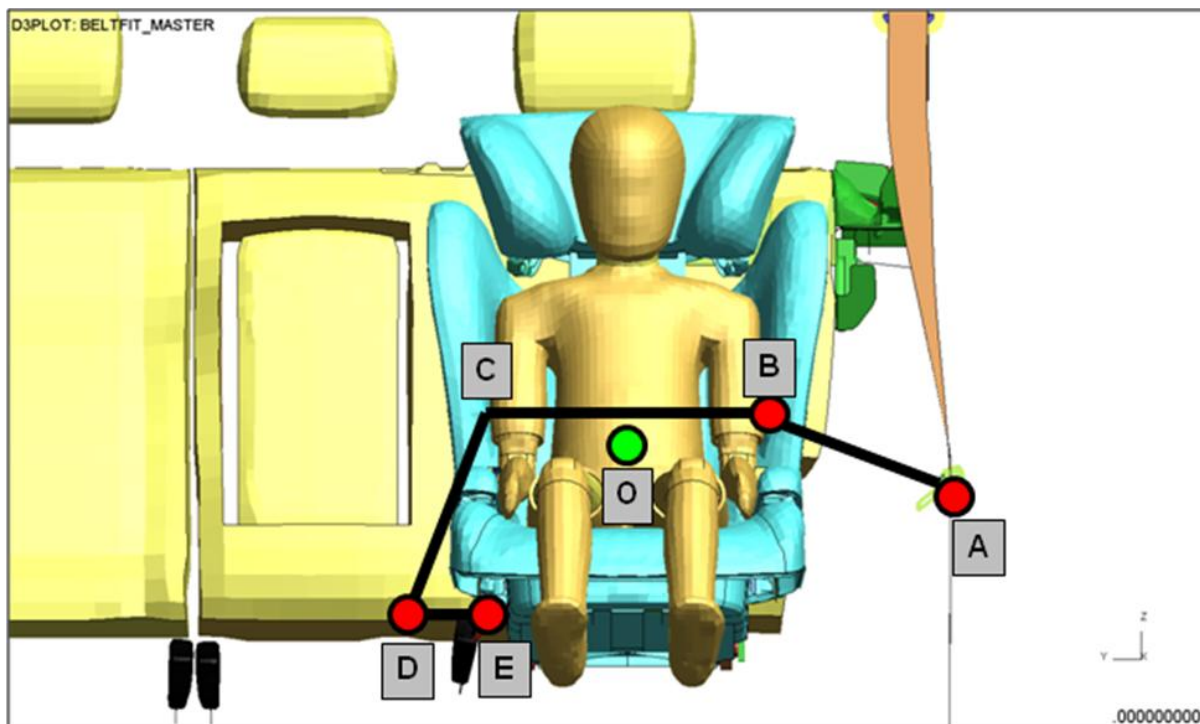


Fig 4: The belt does not go through a linear path during the belt fit simulation, the above shows the path it takes before the slack is taken out. A and E are user defined by the three points on the buckle tongue

The set up is now complete and the model can now be run, whereby to ensure the slack is taken out, the belt is pulled from the end point and anchor point as shown in Fig 2. To ensure the belt does not penetrate through the obstacles during the fitting stage as well as the belt not passing through a linear path from point A to E, a number of other processes occur. As the simulation begins the arm rests raise upwards and one of the shoulder guides moves upwards whilst the other down. The arm rests allows the belt to pass below it and before the slack is taken out, it moves downwards to its final position. The shoulder guides take the belt and guide it, to avoid any contact penetrations through the head restraint clip.

### 3 Results

Once the user has defined all of the parameters and run the simulation a fitted belt is created. In order to check the fitted belt replicates what is seen in reality a number of checks should be taken, i.e. the head restraint clip to retractor, buckle head to arm rest, across the lap as the previous method had difficulty in modelling this. The results can be seen in Fig 5.

Compared with the previous method improvements can be seen in the quality of the mesh, as irregular sized elements are not apparent in these areas. Due to the processes in the simulation, the likelihood of contact penetrations is eliminated but as with any modelling should be checked. Once the checks are complete the belt is now useable in a full car model.

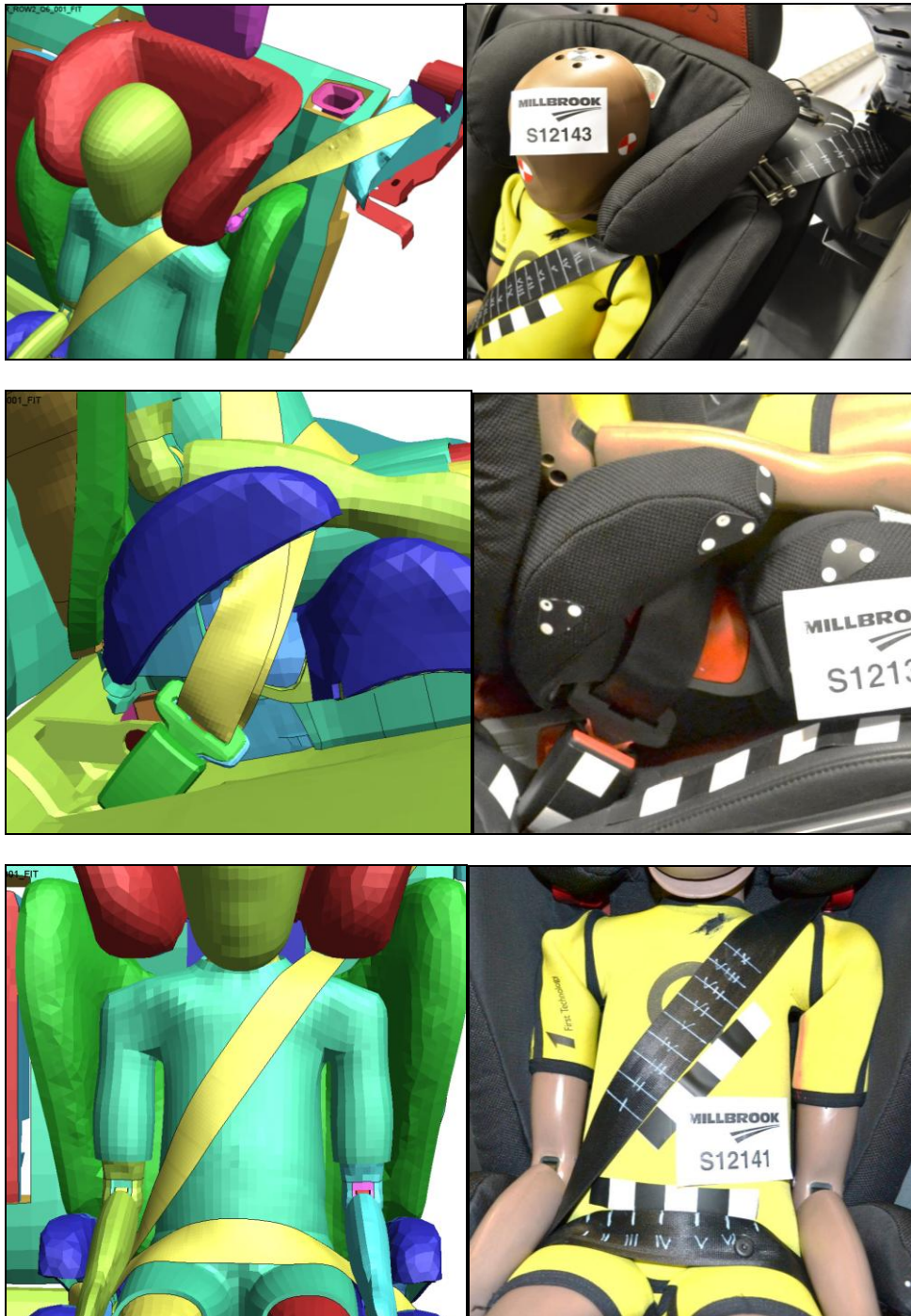


Fig 5: The fitting belt compared with reality for the head restraint clip, buckle tongue and across the lap of the dummy

#### 4 Discussion

The new method has generated a belt which is repeatable and simplified from the current adapted primer method. However, it is not completely automated as the user still has to create the environment and export geometry and then run a script that picks a number of points. In comparison to the primer method, the time needed for the user to generate a fitted belt is substantially reduced. During some sensitivity studies it was found across 3 different users to take between 3 to 6 hours to produce one fitted belt in the original method versus 30 minutes for the new. There is still potential in the new method for user error as the user has to pick some points, although the script was generated to reduce the user from manually writing in coordinates.

It would be almost impossible to completely avoid any user intervention in generating the belt. If you compare both methods, the user has to position the child seat and dummy in the rear seat, however this is where the methods then deviate. In the primer method the belt path needs to be defined through point picking and in the new method the geometry needs to be exported and implanted into a separate include. This is where the main difference occurs and where the variability in users generating a belt in the same environment is eliminated as the new method uses an already generated belt. Reducing variability offers a number of advantages as it should mean the affect on kinematics and injury values should be minimised for that specific environment. Further ahead if a new car program is being created and a prediction for performance against these criteria is required, past results of previous vehicles can be used to help ascertain the accuracy of results. Less variability and having had good correlation to reality are good indicators to give confidence using FEA as a predictive tool for assessment.

The belt material in the new method is different from the previous method as they are not seat belt elements, the belt also does not use any slings at the buckle tongue and retractor, which can be common practice when defining belts in adult dummies. This means the user has to rely on the contact definitions, which can potentially make a small impact into run times as one of the highest expenses in computation time, is through contact definitions [3].

The belt uses MAT\_034 and to ensure during a full vehicle crash the belt does not 'curl up' a number of beams are placed along the belt. This is not necessarily a disadvantage as these beams were required in the original method as well to assist the belt, therefore there is some similarity between both methods.

## **5 Summary**

Overall the automated belt fitting method has provided a tool to generate repeatable belt paths for a given environment. In comparison to previous methods where user intervention was high, the new method is much more simplistic even though the user still has to specify a number of key inputs. The method was found to show much more realistic belt paths when compared to the primer fitting tool method and also comparing it with reality. Finally, the belt fitting method has created a more efficient and effective way of generating a belt across a child seat and child dummy.

## **6 References**

- [1] Euro NCAP: "European New Car Assessment Programme Assessment Protocol – Child Occupant Protection", version 6.1, 2012.
- [2] Mara, G.E.: "Boundary Conditions for the Virtual Testing of Athletic Footwear", PhD thesis, 2007.
- [3] Arup: "Oasys LS-DYNA Environment Training Courses – Introductory Training Course", 2012.