Gero Zechel, Michael Beitelschmidt, Dresden University of Technology;
Helmut Netter, Bombardier Transportation

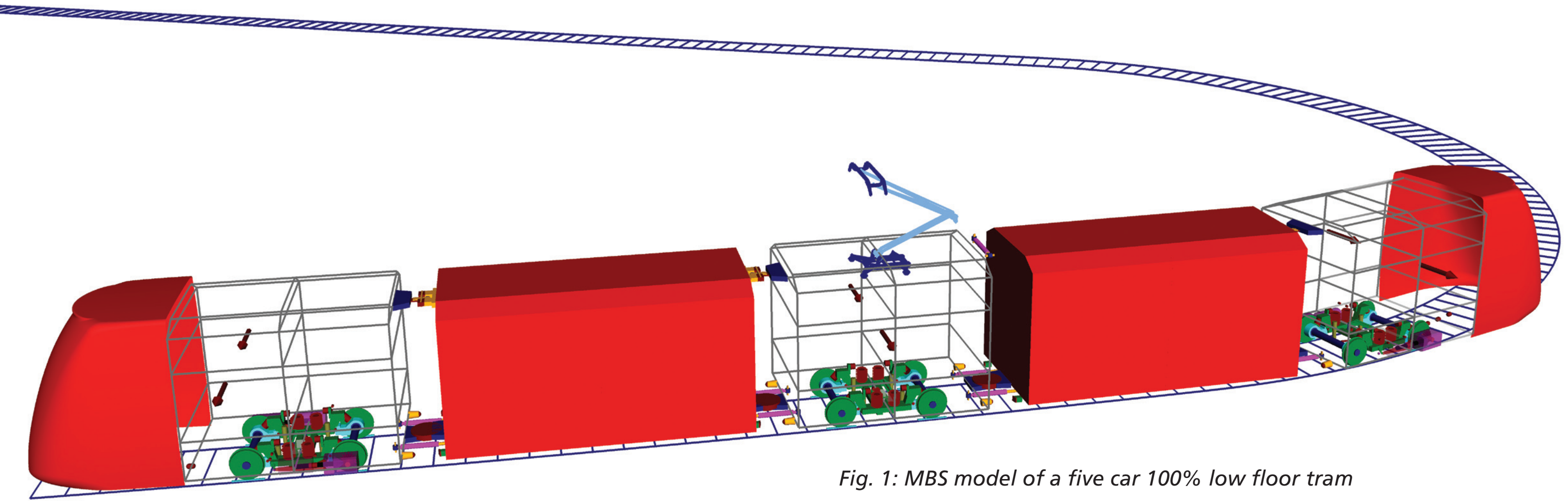

Fig. 1: MBS model of a five car 100% low floor tram

Development of a SIMPACK User Routine for Dynamic Light Rail Vehicle Gauging Simulations

TECHNISCHE UNIVERSITÄT DRESDEN

BOMBARDIER

More and more cities are seeking public transportation systems that offer safe, comfortable and effective mobility integrating seamlessly into the urban landscape. To meet this demand, made-to-measure vehicles have to be developed that allow high capacity cars on narrow spaced infrastructure. In a joint venture between the Chair of Vehicle Modeling and Simulation at the TU Dresden and LRV (Light Rail Vehicle) Vehicle Engineering at the Bautzen site of Bombardier Transportation, a SIMPACK User Routine has been developed that obtains the dynamic vehicle envelope for arbitrary train/track combinations with the push of a button.

LIGHT RAIL VEHICLE GAUGING

The complex kinematic and highly dynamic behavior of modern articulated light rail vehicles require elaborate research of the resulting dynamic vehicle envelope and the structure gauge needed to rule out collisions of vehicles with other vehicles and trackside objects.

The variety of vehicle configurations, the uniqueness of infrastructures, and the generally low vehicle quantities common in light rail demand flexible and efficient gauging methods. While kinematic studies are crucial for the early conceptual design, all dynamic factors have to be included for design verification at the end of the development phase.

Conducting early gauging with simplified kinematic models and integrating gauging simulations into full-scope multi-body simulations, like the one shown in Fig. 1, throughout development of a vehicle is highly beneficial.

INPUT CONCEPT

Even for sophisticated MBS models, the outer shape of the vehicle can be completely ignored. For gauging, however, the precise vehicle contour is vitally important. To make it available in the models without relying on CAD data, SIMPACK Input Functions have been chosen. They can be used to describe the contour as the car width over the car center plane. To define the origin of these contours on the car, body fixed markers can be created separately.

This way, the contours can be created in

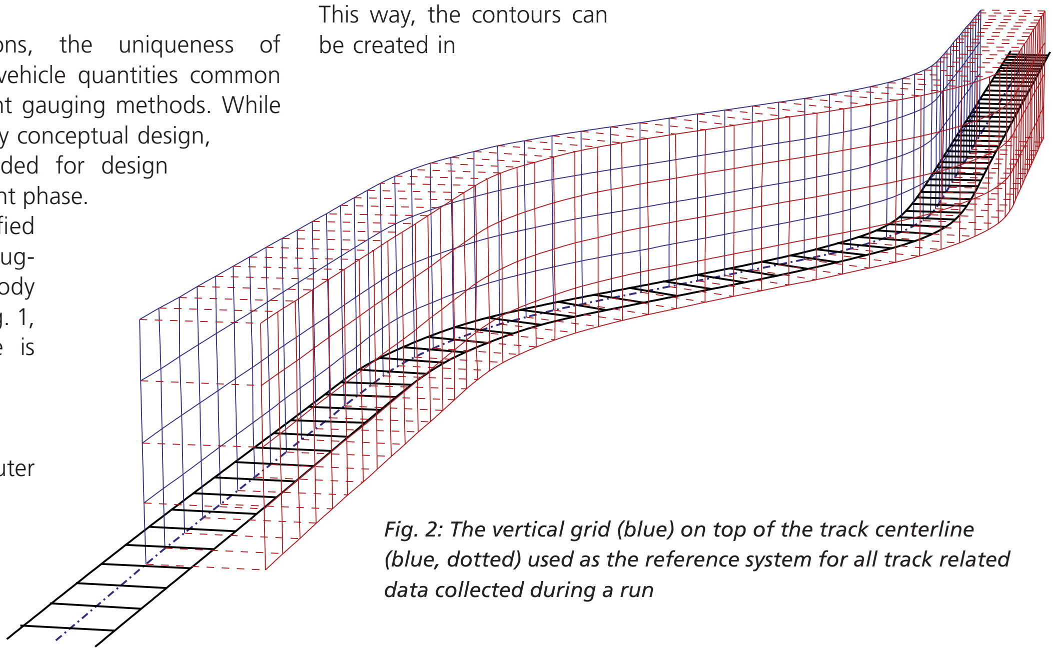

Fig. 2: The vertical grid (blue) on top of the track centerline (blue, dotted) used as the reference system for all track related data collected during a run

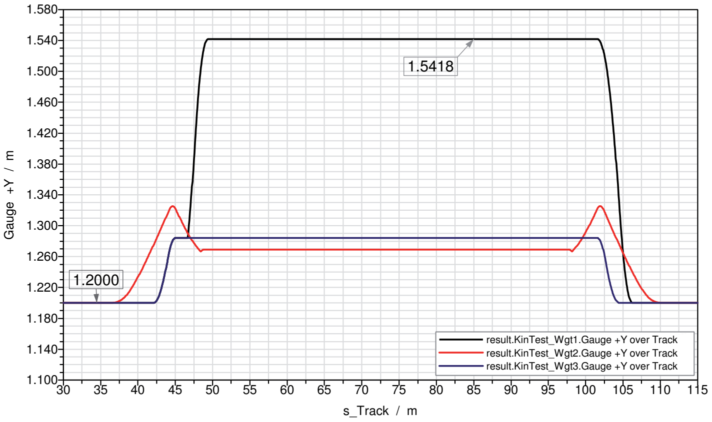

Fig. 3: The maximum lateral distance of the first three cars to the track centerline, plotted over the track length

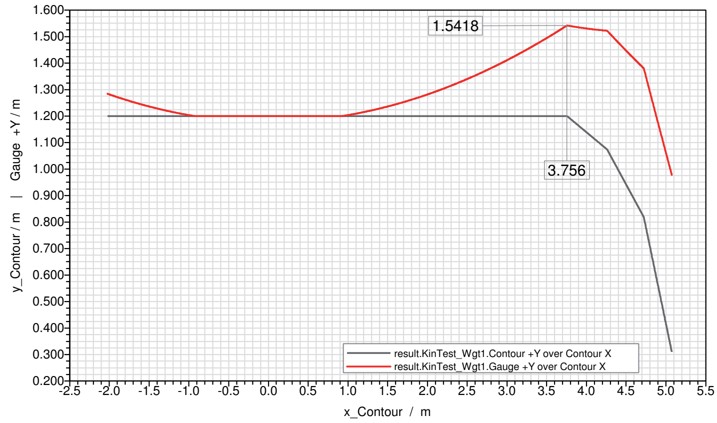

Fig. 5: Grey: The left side contour of the first car (view from above, car pointing to the right). Red: The dynamic envelope of the car during the whole run (relative to the car reference system)

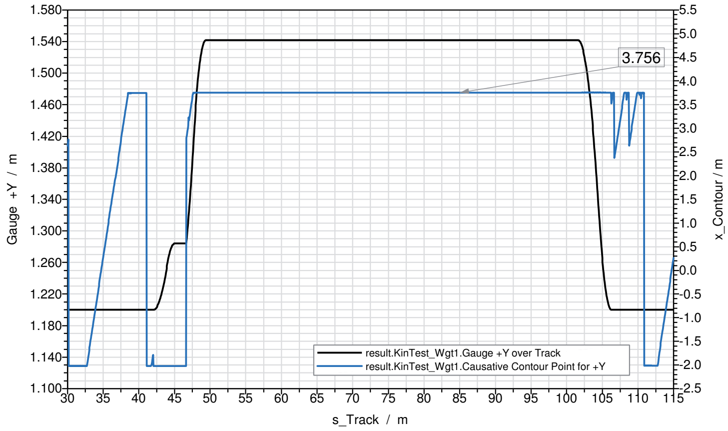

Fig. 4: The first car's maximum distance along the track (black), along with a curve indicating the contour point responsible for the maximum value (blue)

substructures, varied using substitution values, stored in databases, and reused over several cars or models. For gauging simulations using these contours a SIMPACK User Routine has been developed.

IMPLEMENTATION

To achieve a seamless integration into the SIMPACK user interface, the concept was realized by means of a SIMPACK User Result Element written in Fortran using the built-in editing and compiling tools with a third-party compiler.

This User Result Element can be integrated in any vehicle model by using the Model Setup GUI. In the Result Element window, the Input Function and the Marker can be picked from the list of available MBS elements and output options can be set. Although one Result Element represents only a single contour, it can be added as many times as needed to cover the whole vehicle.

"The concept was realized by means of a SIMPACK User Result Element."

In general, Result Elements are evaluated offline when Measurements are performed. They are called by SIMPACK beforehand for initialization, then at every communication point, and afterwards to write the output vectors.

While the measurements are running, four data collections are built up and stored in memory allocated by SIMPACK using the available Access Functions. The first data collection keeps state information that is needed for every time step, like the number of contour points, the track length, and a time step counter. The second data collection is more complex — it stores all values related to the infrastructure. To do so, a grid is spanned over a vertical surface positioned on top of the track centerline, as shown in Fig. 2 in blue. Every point in the area around the track can be assigned to a unique grid point as long as the distance from the track is less than the smallest track radius — this covers the relevant area at all times. The maximum centerline distance any contour point reaches within each of the resulting volume segments (Fig. 2, shown in red) is gathered during the measurement run.

To get this information, an algorithm is run for each time step that circles through all contour points one by one and calculates their distance to the grid using an iterative method. If a new maximum distance is found along the track, this value, as well as the index of the contour point responsible for the new maximum and the absolute position of the contour point, is stored in the memory.

The third data collection is filled in a similar way — but stores values in reference to the contour itself. At the end of the run, it therefore contains the maximum distance from the track for each contour point along with the information at which grid point this maximum occurred. Finally, the fourth data collection gathers time-domain data like the absolute position of each contour point to certain time steps.

While memory usage stays reasonably low, these four data collections are not suitable for plotting as is, so they are evaluated at the end of the run to gain comprehensive one-dimensional output vectors. This data is stored in the resulting standard result-file and can be easily plotted using the SIMPACK PostProcessor.

DATA EVALUATION AND PLOTTING

First and foremost, the track related data collection is evaluated to get the maximum distance of the vehicle to the track centerline over the track length. Two output vectors are created for the left and for the right hand side of the track. Fig. 3 shows the results for a kinematic vehicle model passing through a right turn. Three Result Elements where defined for the first three cars of the vehicle. In the plot, the results for the left-hand side of the track for each car are overlaid by drag-and-drop in the SIMPACK PostProcessor. The plot shows that the maximum distance from the track at curve entry and

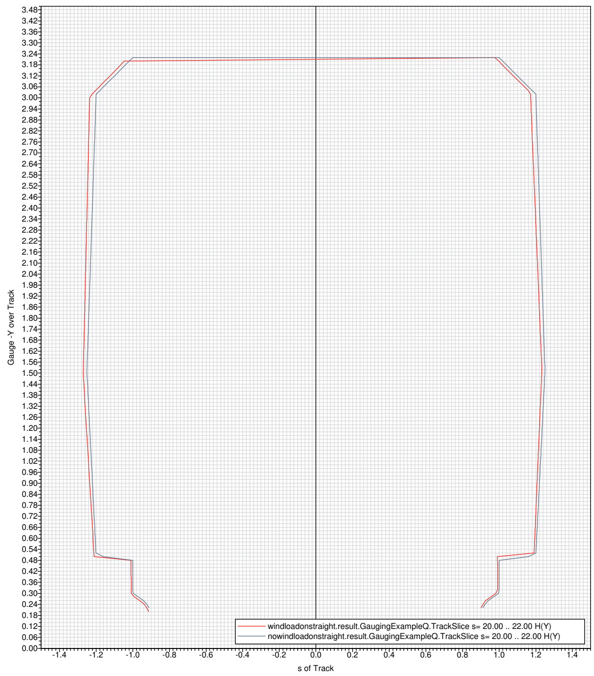

Fig. 6: Structure gauge for a vehicle on a straight track with and without crosswind load

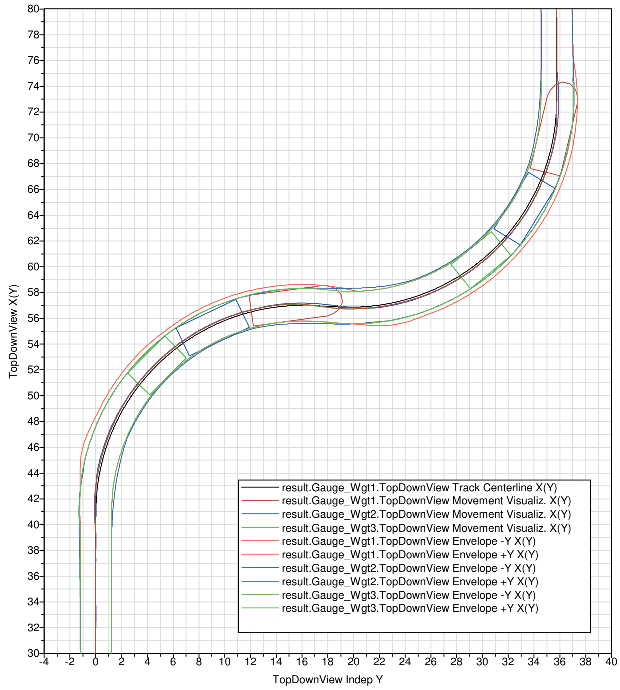

Fig. 7: Top-down view of an S-bend track showing the track centerline, the position of three car contours at two different time steps, and the kinematic envelopes of each car projected to the ground

part of the curve exit is caused by the second car (red), while the first car (black) leads to the maximum distance along the track between curve entry and exit (1.54 m).

To provide information about what part of a contour is responsible for maximum displacement, the track related data collection is also filtered to produce the output vector shown in Fig. 4. The plot indicates that the contour point located 3.76 m in front of the bogie (where the body fixed marker is placed) results in the maximum distance for the given example.

Finally, the evaluation of the contour-related data collection provides the outputs shown on Fig. 5. The left side contour of the first car is depicted in grey, the dynamic envelope of the car during the whole run is shown in red. This plot confirms the values found above, and shows that the head contour needs to be changed to lower the maximum distance to 1.4 m from the track centerline (everything in the range of 3.5 m to 4.3 m in front of the bogie).

By applying more advanced evaluation techniques to the track related data collection, plots are made possible that show the structure gauge needed by the vehicle along the track. This can be helpful when analyzing the rolling behavior in curves, the clearance at railway platforms and the influence of crosswind loads, Fig. 6, for example.

If gauging data is needed in the context of the simulated scenario, absolute positions stored in the track-related and the time-domain data collection can be combined to get a general idea of the vehicle's kinematic and dynamic gauging behavior. Fig. 7 shows an example plot of three cars passing an S-bend. In addition to the track centerline and the car contours at two different time steps, the kinematic envelope of each car projected to the ground can be seen.

CONCLUSION

When combined in the right way, SIMPACK's built-in tools and functionalities can be used for gauging simulations with high accuracy, even for highly dynamic situations like the one shown in Fig. 8. The complexity of the Gauging User Routine can be reduced to consider just one single contour, because it can be utilized many times in a single model at the same time and the results can be overlaid in the PostProcessor. Its use, however, is not restricted to the car body — it can also be applied to pantographs, bogie parts, or even markers fixed to the inertial system to include objects of the infrastructure like railway platforms or masts of the catenary wire.

This extreme versatility is possibly the biggest advantage compared with many other gauging software tools as it allows for the analysis of a huge variety of vehicle designs and infrastructure systems which LRV-engineering has to cope with every day.

"...SIMPACK's built-in tools and functionality can be used for gauging simulations with high accuracy..."

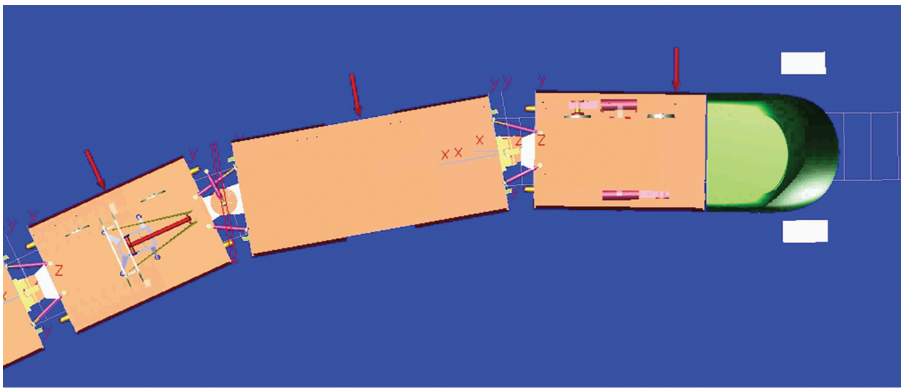

Fig. 8: Dynamic model of a 5-car tram passing a gate after a curve (top view)