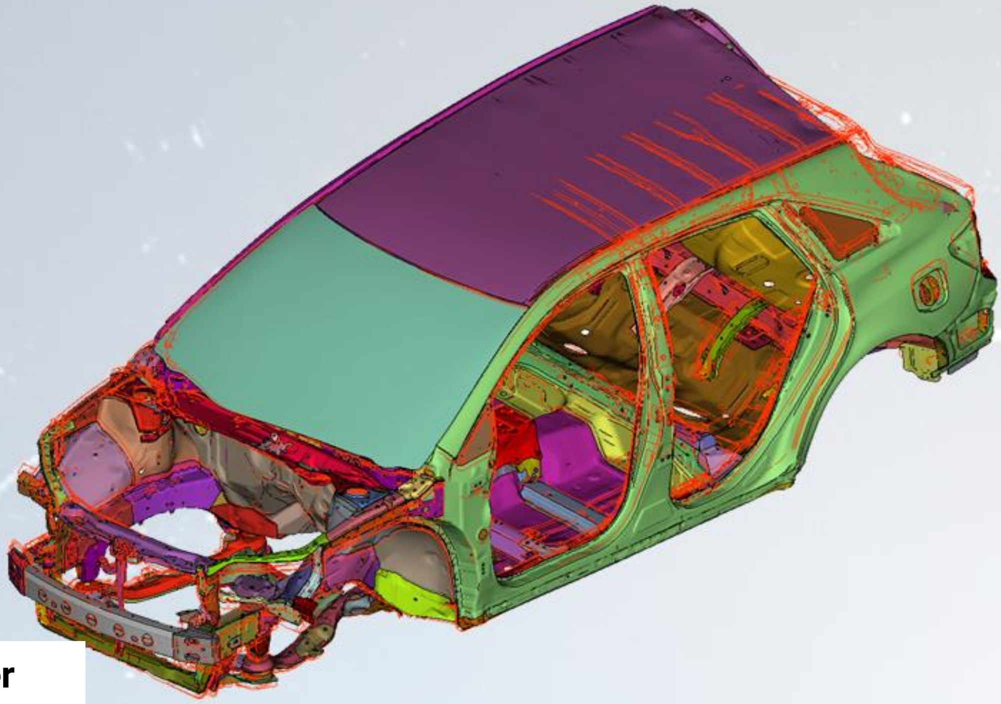
BETA SIMULATION SOLUTIONS

physics on screen



White paper

Simulation enabling technologies

BiW first torsional mode prediction using Feature-based machine learning

The first torsional mode frequency value is a key value for concept cars. Having a trained algorithm, which provides answers based on a Finite Element Body-in-White model, without running a solver analysis, accelerates the digital model validation process.

Introduction

Identifying the first torsional mode of a vehicle's Body-in-White (BiW) is a crucial stage in the product development. During product development, the BiW may undergo several types of modifications in its design and engineering specifications. These modifications may include changes in geometry (shape of parts), changes in parts thickness, and changes in materials or connection types.

Such modifications require the redesign of the BiW and the creation of a new simulation model, and then a new run of an FEM analysis for the collection of the desired responses. This redesign process may need to occur several times in the development cycle of a BiW. To speed up the process, predictive models can be created for the prediction of the first torsional mode value based on a modified Finite Element model of the BiW. This way, the identification of the first torsional value is much faster compared to running an FEM analysis. Multiple "what if" studies can be performed, without the expense in analysis time, and successful or failed modifications can be recognized faster.

Predictive models or "predictors" can be created using Machine Learning functionality implemented in KOMVOS, in combination with model feature extraction algorithms and ANSA Morphing and Optimization capabilities. The use the Machine Learning functionality requires that the optional "ML toolkit" has been installed, and the respective license feature is enabled, which is available upon request.

Both Design Variable and Feature-based Machine Learning predictive models can be trained and used to identify the first torsional mode frequency value. However, there are important differences between the two.

In this study the Feature-based Machine Learning functionality was used to create predictive models that will accurately predict the first torsional mode frequency value of a car's body-in-white (BiW).

Feature based and Design Variable based Machine Learning

Feature-based Machine Learning

The Feature-based Machine Learning method relies on a process called "feature extraction". With this process, critical features of an FE model are collected and their effect towards the requested response is evaluated and documented. The evaluated features are related to the model shape as well as the materials, the connections, etc.

To train the Feature-based model, many different FE models must go through the "feature extraction" process. In this case, the response that is measured and used in training is the first torsional mode frequency value. When the predictor is trained, it accumulates knowledge of many different models' features and their impact to the first torsional mode frequency value.

To make a prediction, the Feature-based predictor reads a new, "unseen" FE model, performs feature extraction, and based on its past training, provides a prediction of the first torsional value for the under study FE model, which was previously "unknown" to the predictor.

Design Variable based Machine Learning

The Design Variable (DV) based Machine Learning is the most common process for collecting input variables (design variables) and output variables (responses) in creating a predictive model.

To train the Design Variable based model, multiple FE models can be created via ANSA's Optimization tool during a Design of Experiments (DOE) process, using Design Variables that modify the models shape and parameters. This way input variables are collected. As responses, any desired value collected from ANSA or META (during post-processing) can be retrieved at each iteration, forming the output variables. With input and output variables available, the training can be performed to create a new Design Variable based predictor.

To make a prediction, the DV based predictor requires values for the Design Variables. It does not require any FE model. Once the DV values have been added, the predictor can instantly provide value predictions for the responses used for its training.

Why Feature based over Design Variable based Machine Learning

The two machine learning models are significantly different in the way they are trained and how they predict. The main difference between the Feature-based and DV-based approach is that the first relies on the actual model geometry, while the DV-based is related to a DOE and the values

of the design variables, and the predictor learns from the impact of these design variables changes.
Design Variable based predictors are faster to be trained and predict but they are constrained by the design variables and responses they were trained with. On the other hand, Feature-based predictors may take longer to be trained and predict due to the feature extraction, but the input is the simulation model itself, without the use of any design variables.
In this implementation case, the ML Toolkit was used for the prediction of the first torsional mode frequency value of a vehicle's BiW, but its implementation can be further extended to the prediction of bending and local mode frequency values.*
The feature-based predictors, even though they are trained on the FE model geometries they are not limited by the preselection of any design variables. As we see in this study, the burden of feature-based predictors training can be lifted by using "Incremental Learning". This way, the process is accelerated, since an existing feature-based predictor is updated with new data, while maintaining the knowledge of previous training cycles.

*Features subject to change. Contact our services for the latest software updates.

Feature-based training Scenarios

Basic

This scenario is the default Machine Learning option in KOMVOS. It uses a variety of extracted features and machine learning algorithms. The initial Dataset is split to Train and Test partitions according to a user defined ratio. The predictor is created through a process of learning and validation and finally the Test data are used to provide the predictors key performance indicators (reports, Mean Absolute Error, etc.) (Figure 1).

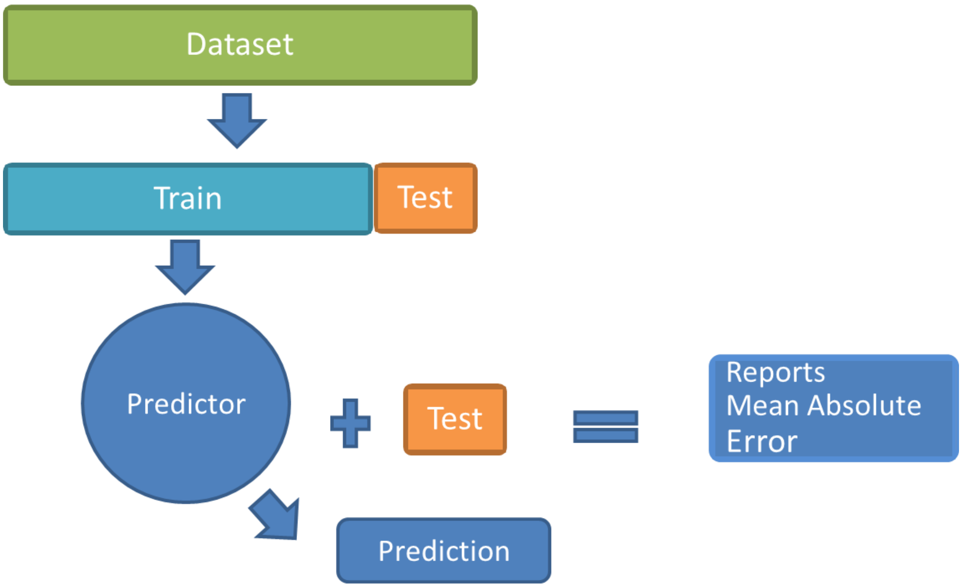


Figure 1 Basic training process

Incremental learning

Incremental learning is suitable for cases where a single BiW is studied, and variations of this BiW are generated gradually during this product's development cycle. This way, the predictive model can always be up to date with the latest developments of the model with minimum additional time investment.

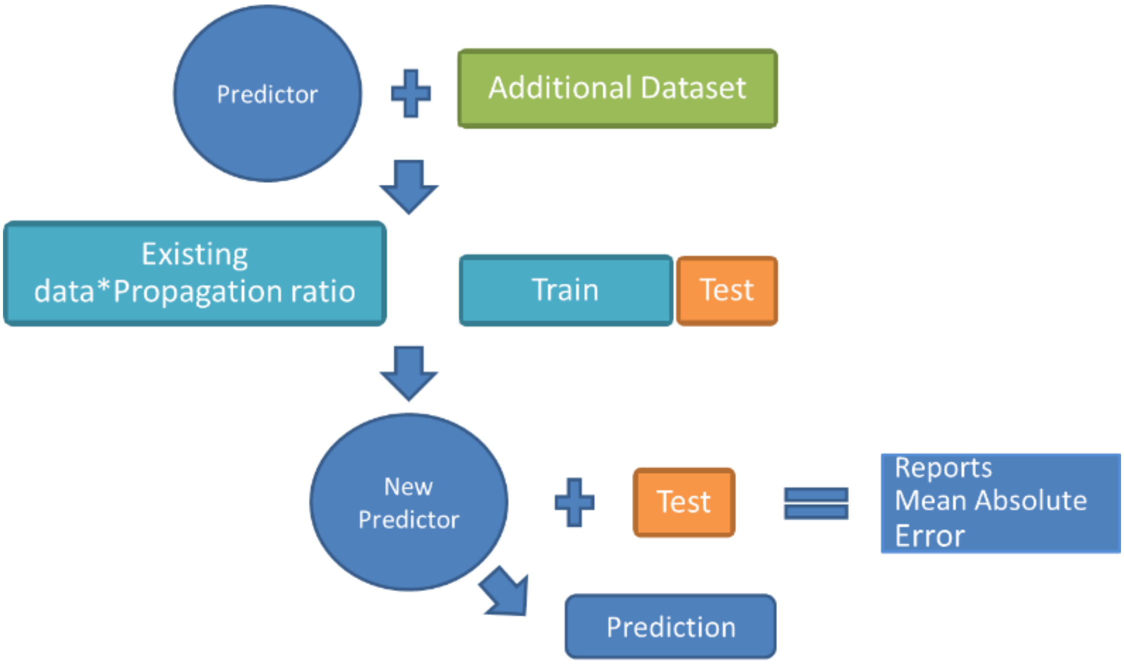


Figure 2 Incremental training process

An existing predictor that supports Incremental learning is selected to be updated with new data. The new Dataset is split to Train and Test partitions, similar to the Basic scenario. Existing data from the selected predictor pass to the new predictor with a user defined propagation ratio. This ratio states the percentage of old data that will pass to the new predictor. This data are already processed so no time is consumed to train the new predictor on past data.
A propagation ratio of 0% means that the training will use only the new data. The contribution of the old data is only to the initial state of the predictor. A propagation ratio of 100% means the model will use all the old data along with the new. It can be expected that the smaller the propagation ratio, the faster the training will be, and the easier old data will be forgotten. On the other hand, the higher the propagation ratio, the slower the training will be, and the less easily old data will be forgotten.
At the end, the training of the new Train set completed and the new predictor, containing old and new data is tested to produce its Key Performance Indicators.

Retrain

The Retrain scenario is similar to the basic scenario. The retrain process is applied on an existing Predictor (default or incremental) and uses all the data from the existing predictor additionally to the newly selected data, to train a new predictor.

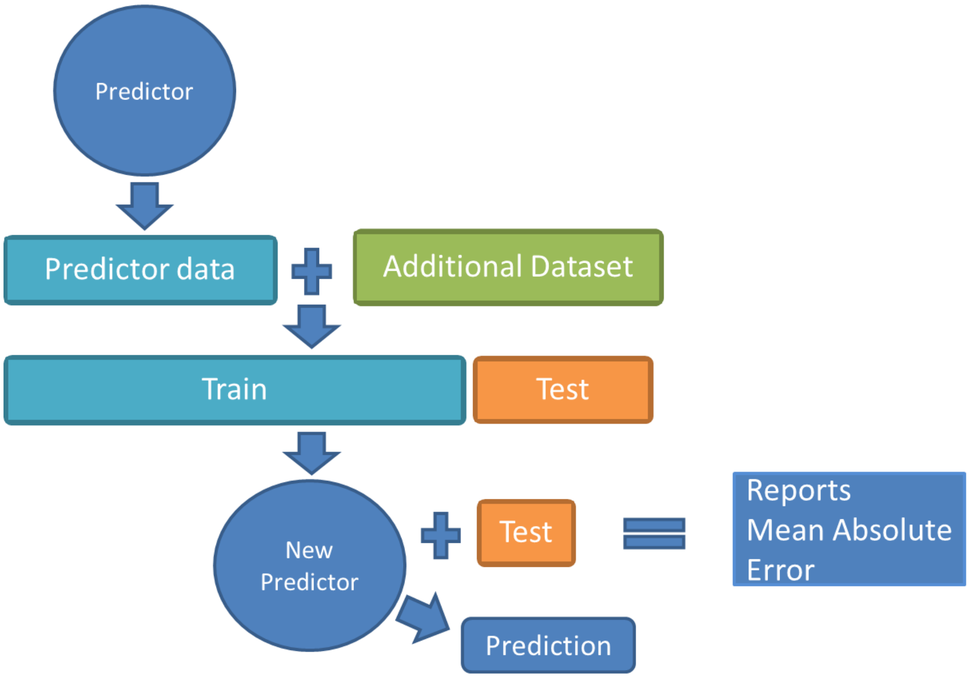


Figure 3 Retrain process

The difference to the Incremental learning with 100% propagation ratio is that, in the Incremental learning, “hyperparameter tuning” and model selection do not take place. These are performed only on the creation of the initial predictor with the first batch of data. In Retrain, however, the entire ML pipeline (including model selection and hyperparameter tuning) is executed again on the combined dataset. This means that, if the new data is completely unrelated to the initial data batch, then “Retrain” is a more suitable option.

Example 1

In cases of Simulation Runs from the same model DOE, the Machine Learning toolkit in KOMVOS can yield accurate predictions with both DV-based and Feature-based predictors. However, if the selected dataset contains runs from various models, the feature-based scenario is the only applicable option.
Let us consider five different BiW models of different cars. Using ANSA's morphing capabilities and the Optimization tool, 20 Simulation Runs were generated for each BiW, by modifying several variables, such as geometric properties (width, height, pillars position etc.) and material properties (thickness, elasticity) (Figure 1). Half of these Runs have been used in the D1 and the other half in the D2 dataset. So, D1 contains 10 Runs from 5 BiWs (50 Simulation Runs), and D2 contains another 50.

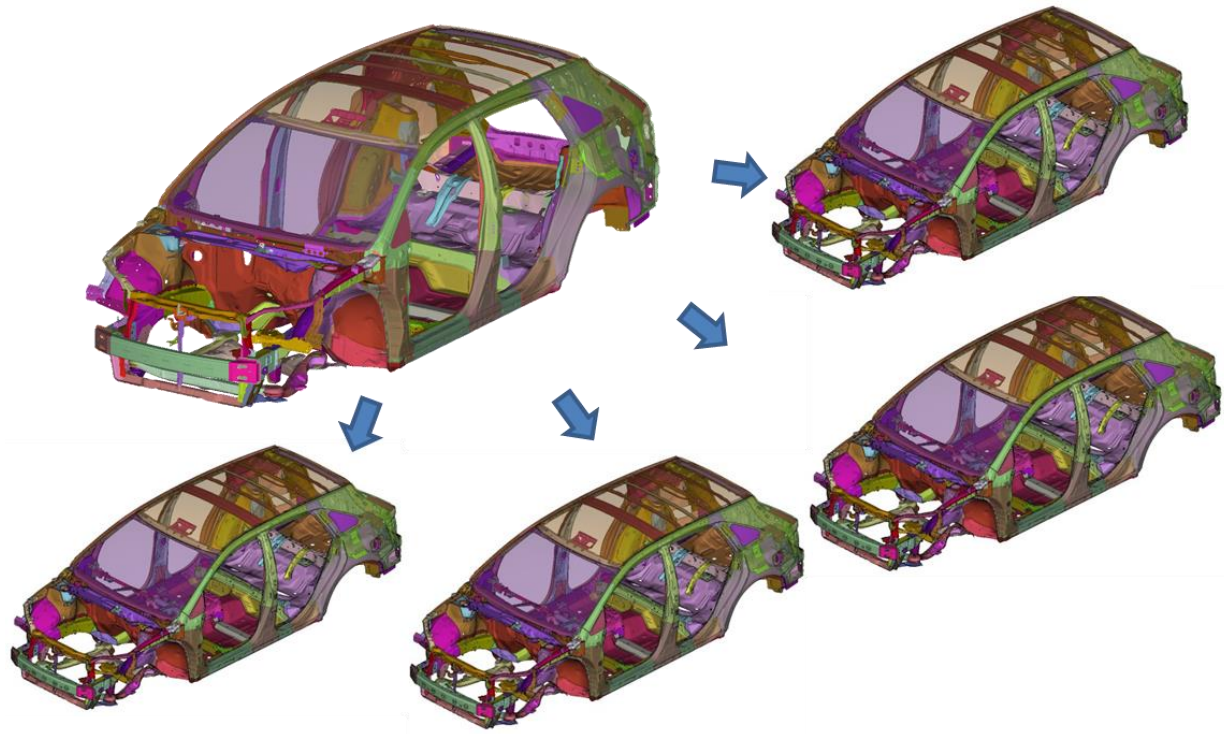


Figure 4 One BiW produces multiple variations using morphing

Basic scenario

For this basic scenario, the predictor model was trained on data from the D1 dataset. Then, the Predictor (trained predictive model) was tested and evaluated on predicting the First torsional values for the test set of D1 (runs that were not used for training) and from D2 (that was completely unknown to the Predictor).

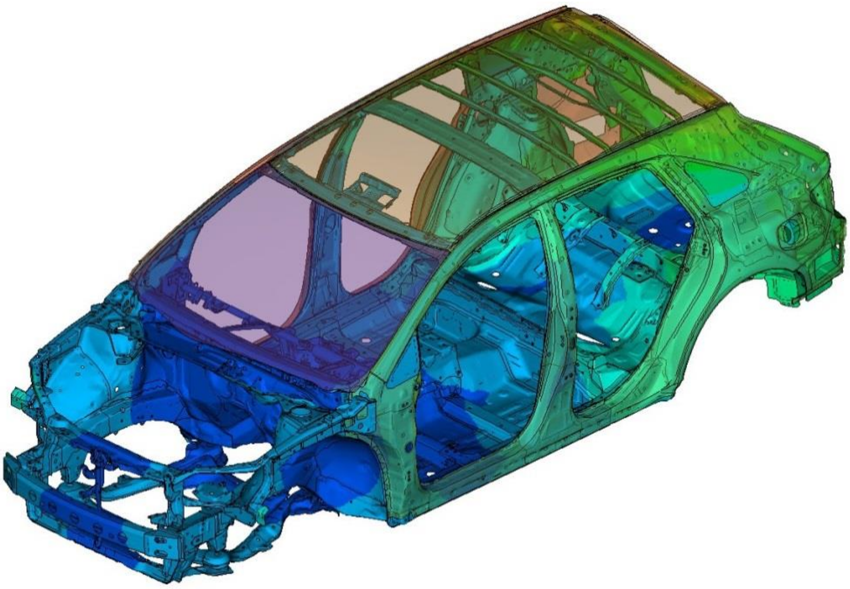


Figure 5 BiW Displacements at First Torsional mode

Table 1 Basic predictor

Dataset	Mean Absolute Error (MAE)
D1	**0.2147**
D2	**0.5297**

On all experiments presented in this document, the train-test ratio is 70/30. The error (MAE) is measured in Hz as the predicted value is measured in Hz (First torsional mode Frequency value).

Outcome

It can be observed from this example that the predictor of the basic scenario is able to accurately predict the first torsional mode values from the D2 dataset. This was possible as the distributions of the two datasets are similar because, even though they contain Runs from multiple BiWs, these BiWs are the same across the two datasets.

Example 2

In the second example, three training scenarios (basic, incremental and retrain) have been tested and compared in terms of accuracy and training time. For all three scenarios a passenger car's BiW was used as the initial simulation model.
Using ANSA's morphing capabilities and the Optimization tool, several simulation runs were generated by modifying several variables of the BiW, like width, height, thickness, material properties etc., to create the first dataset D1 (50 Simulation Runs). The same method was used to create a second dataset D2 (50 additional Simulation Runs), with the difference that the Design Variables range is modified to higher values. Therefore, the distribution of both the target value (1st torsional mode) and of the extracted features of datasets D1 and D2 are distinct, even though the base model and the design variables are the same.

Basic Scenario

For the basic scenario, the ML model was again trained using the D1 dataset. Then, the Predictor was tested and evaluated on predicting the first torsional values for simulation runs from D1 (on runs that were not used for training) and from D2 (that was completely unknown to the Predictor).

Table 2 Basic predictor

Dataset	Mean Absolute Error (MAE)
D1 test set	**0.1734**
D2 entire set	**6.7628**

The Mean Absolute Error (MAE) is measured in Hz as the predicted value is measured in Hz (first torsional mode frequency value). In this case, training this basic model required 14 minutes.

Incremental Scenario

For this scenario, a predictor was created with D1 and opting for "Incremental Support", called the initial incremental predictor. Then, the predictor was incrementally trained with the D2 dataset. For demonstration purposes, this is performed multiple times using a few variations of the propagation ratio. The propagation ratio is a setting that defines the percentage of old training data that will be reused for training together with the "unseen" data as they arrive.
The initial predictor was evaluated on the test set of D1 and on the entire D2 dataset. At the next stage, after incrementally training with D2, the resulting predictor was evaluated on the test set of D1 and on the test set of D2. Essentially, we only evaluate on data not seen during training, as is expected.

Table 3 Initial Incremental Predictor

Dataset	Mean Absolute Error (MAE)
D1 test set	**0.3783**
D2 entire set	**75.8189**

Table 4 Incremental Predictor – next stage

Dataset	MAE with propagation ratio 20%	MAE with propagation ratio 50%	MAE with propagation ratio 100%
D1 test set	**1.1357**	**0.5042**	**0.2316**
D2 test set	**0.8176**	**0.5674**	**0.7014**

Retrain Scenario

For this scenario, a basic model was trained initially using D1 and then it was retrained with D2. Once the predictor was retrained, it was evaluated on the test set of the combined dataset D1+D2.

Table 5 Retrain Predictor

Dataset	Mean Absolute Error (MAE)
D1+D2 test set	**0.4171**

The Mean Absolute Error (MAE) is measured in Hz as the predicted value is measured in Hz (first torsional mode frequency value). Overall training time was ~1hour (28 minutes for the initial model and an additional 31 minutes for the retraining with D2).

Outcome

The Basic scenario's predictive approach did not manage to predict with accuracy the first torsional value of the D2 dataset that was “unseen” to the predictor, as a higher Mean Absolute Error values presented. The Retrain Scenario's predictive approach managed to predict values from both datasets (D1, D2) with higher accuracy. However, the double training time was required for the retraining based on both datasets. The Incremental scenario's predictor managed to predict values from both datasets with high accuracy while maintaining a short training time making it the optimal choice for this type of experiments.

Training Time

The training times of the presented experiments are as follows:

Table 6 Training time

Scenario	Training time Linux	Training time Windows
Basic	**14'**	**28'**
Incremental Initial	**15'**	**20'**
Incremental	**3'**	**3'-5'** (depending on the propagation ratio)
Retrain	**14'**	**31'**

Conclusions

From these two examples, we can conclude that each scenario is better suited for different types of experiments since there is a tradeoff between training time and achieved accuracy of a predictor.

Incremental training can be proved very helpful in cases where a simulation model is developed and most likely this will be soon updated. This approach will provide accurate prediction results with reduced training time to update the predictor.

In cases of various available simulation models and different models between datasets, the Basic or Retrain predictive training approaches provide more accurate results.

About BETA CAE Systems International AG
BETA is a simulation solutions provider, dedicated to the development of state-of-the-art software systems for CAE. For almost 30 years, we have been developing tools and delivering services for the front-runners in numerous sectors by listening to their needs and taking up even the most demanding challenges. For more information on BETA CAE systems, our products, and our services, visit www.beta-cae.com

Headquarters
Platz 4
CH-6039 Root D4,
Switzerland
+41 415453650

Email: ansa@beta-cae.com
URL: www.beta-cae.com

BETA
SIMULATION SOLUTIONS

physics on screen