

HyperMoVal: Interactive Visual Validation of Regression Models for Real-Time Simulation

H. Piringer¹, W. Berger¹, and J. Krasser²

¹VRVis Research Center, Vienna, Austria

²AVL List GmbH, Graz, Austria

Abstract

During the development of car engines, regression models that are based on machine learning techniques are increasingly important for tasks which require a prediction of results in real-time. While the validation of a model is a key part of its identification process, existing computation- or visualization-based techniques do not adequately support all aspects of model validation. The main contribution of this paper is an interactive approach called HyperMoVal that is designed to support multiple tasks related to model validation: 1) comparing known and predicted results, 2) analyzing regions with a bad fit, 3) assessing the physical plausibility of models also outside regions covered by validation data, and 4) comparing multiple models. The key idea is to visually relate one or more n-dimensional scalar functions to known validation data within a combined visualization. HyperMoVal lays out multiple 2D and 3D sub-projections of the n-dimensional function space around a focal point. We describe how linking HyperMoVal to other views further extends the possibilities for model validation. Based on this integration, we discuss steps towards supporting the entire workflow of identifying regression models. An evaluation illustrates a typical workflow in the application context of car-engine design and reports general feedback of domain experts and users of our approach. These results indicate that our approach significantly accelerates the identification of regression models and increases the confidence in the overall engineering process.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—Line and curve generation

1. Introduction

Strict emission rules and steadily increasing demands on performance force car manufacturers to constantly improve the design of powertrain systems [BDHK05]. Different types of numerical simulations have thus become a key technology to analyze complex systems like engines. Especially in early stages of the design process, 1D Computational Fluid Dynamics (CFD) simulations are widely used. Compared to very time consuming 3D CFD simulations, 1D CFD simulations are magnitudes faster and produce much less data. This enables to study the design space by running thousands of simulation runs for different locations of the parameter space. Matkovic et al. [MJJK05] describe, how interactive visualization can support the analysis of such data.

Some tasks, however, require mathematical models that can make predictions in real-time. When testing strategies for Engine Control Units (ECUs), for example, their behav-

ior is analyzed during a simulated drive, which requires simulating the physical behavior. As even 1D CFD simulations are far too slow for such real-time tasks, manufacturers use *surrogate models* instead which are based on statistical regression rather than on physical equations [Hei09].

The identification of surrogate models requires known results at sampled positions of an input parameter space for training and validation. These results come from measurements or from 1D CFD simulations. A particular model is trained to predict one result (e.g., torque) given values for particular attributes (e.g., speed and load). While the process of training a model requires no interaction, most types of regression models have numerous parameters (e.g., kernel type and tolerance values for support vector regression [HTF09]) that must be set before. The complex interplay between these parameters makes finding good training parameters a challenging task. Different models may also perform best for

different subsets of input attributes. Sometimes, implausible data must be filtered before the training.

Before making critical decisions based on the predictions of a surrogate model, it is therefore very important to validate its quality and to possibly improve it in further iterations. This validation involves comparing known results to predictions by the model [Sar94]. Besides looking at criteria that can be derived automatically (e.g., the maximal residual), it is essential for application experts to get a feeling for the model itself and to be able to analyze its behavior also for regions not covered by training data. For these reasons, a fully automatic workflow for model validation is insufficient. Instead, it must be complemented by an interactive visual approach to increase the confidence in the overall process based on the domain knowledge of the engineers.

1.1. Design goals

Based on interviews with application experts and observations of their workflow, we derived the subsequent set of design goals for an interactive visual approach for the validation of surrogate models:

- Relate known results to predictions of models.
- Support a quick identification of deviating regions.
- Convey a feeling of the behavior of a model around any point in space.
- Allow a comparison of multiple models.
- Scale to models with a large number of input parameters.
- Be highly interactive without perceivable delays.
- Be tightly integrated into the workflow of model identification by restricting the analysis to an arbitrary subset of validation data and to update the visualization immediately at modifications of the model.

While the application background of this paper is the development of powertrain systems, the identification and the validation of regression models is of general importance in many application domains [Sne77]. The problem can be formulated as assessing the suitability of a multidimensional scalar function $y = f(X)$ to approximate known values of y at a set of different points for X , where y is a scalar value (called the dependent variable) and X is a point in n -dimensional space (referred to as independent variables).

1.2. Contributions

The contributions of this paper are as follows:

- A *design study* of HyperMoVal, an interactive visualization technique for validating regression models based on a combined visualization of n -dimensional functions and known validation data.
- A tight integration of HyperMoVal within an *interactive workflow* for iterative identification and validation of regression models involving multiple linked views.
- An *evaluation* of the proposed techniques within the application context of engine design based on a case study of an exemplary workflow and the report of user feedback.

2. Related Work

Numerous visualization techniques address the important issue how to visualize multivariate data [GTC01]. Most of these techniques operate on generic n -tuples, for example parallel coordinates [Ins85] and scatterplot matrices [Cle93].

Besides such general purpose techniques, various approaches have been proposed to visualize multidimensional scalar functions $y = f(x_1, x_2, \dots, x_n)$. A direct visualization is trivial for $n = 1$ using line graphs and $n = 2$ using surface plots. The case $n = 3$ is typically addressed using volume rendering and isosurfacing. However, larger values of n are challenging. Most visualization approaches thus reduce the number of parameters to 1, 2, or 3 per plot by assuming specific values for the rest. These values can typically be changed by animation or by interaction.

Worlds within Worlds [FB90], for example, uses a hierarchy of nested coordinate systems. At each level of the hierarchy, the origin of the coordinate system specifies up to three parameters for all further levels until the number of independent variables is reduced to support a direct visualization of the function. A related approach by Mihalisin et al. [MTS91] nests axes hierarchically to plot function values against all combinations of a sampled subset of the parameter space. Jayaraman and North propose a focus+context visualization of multi-dimensional functions [JN02] based on a radial layout of slices. Each slice shows the behavior of the function as sampled along rays emanating from a focal point into a certain direction for one independent variable.

HyperSlice by van Wijk and van Liere [vWvL93] shows all 2D orthogonal slices of a function around an n -dimensional focal point. Each slice represents the function by varying two parameters while assuming the values of the focal point for the rest. This is done for all pairs of variables and the resulting plots are arranged using a matrix layout. Dos Santos and Brodrie extended the idea of HyperSlice to three dimensions to what they call HyperCell [dSB02]. Instead of using a fixed matrix layout, they provide a graph interface that supports the creation of 1-, 2-, or 3-dimensional plots (called cells) and they support multiple focal points. Recently, Nouanesengsy et al. [NSSV09] proposed an approach to analyze multidimensional scalar functions based on projecting line segments into 2D plots. The key idea is to plot the distance of each line segment from user-defined points against the respective function values.

All of these techniques are useful for understanding certain aspects of multidimensional scalar functions, and particularly HyperSlice served as an important and widely accepted starting point for our work. However, the goal of these approaches is to visualize a function as such, not to validate it as an approximation model by means of known data. As motivated in Sec. 1, a validation requires a direct comparison of the approximation model to results from measurements or physically-based simulations [Sar94].

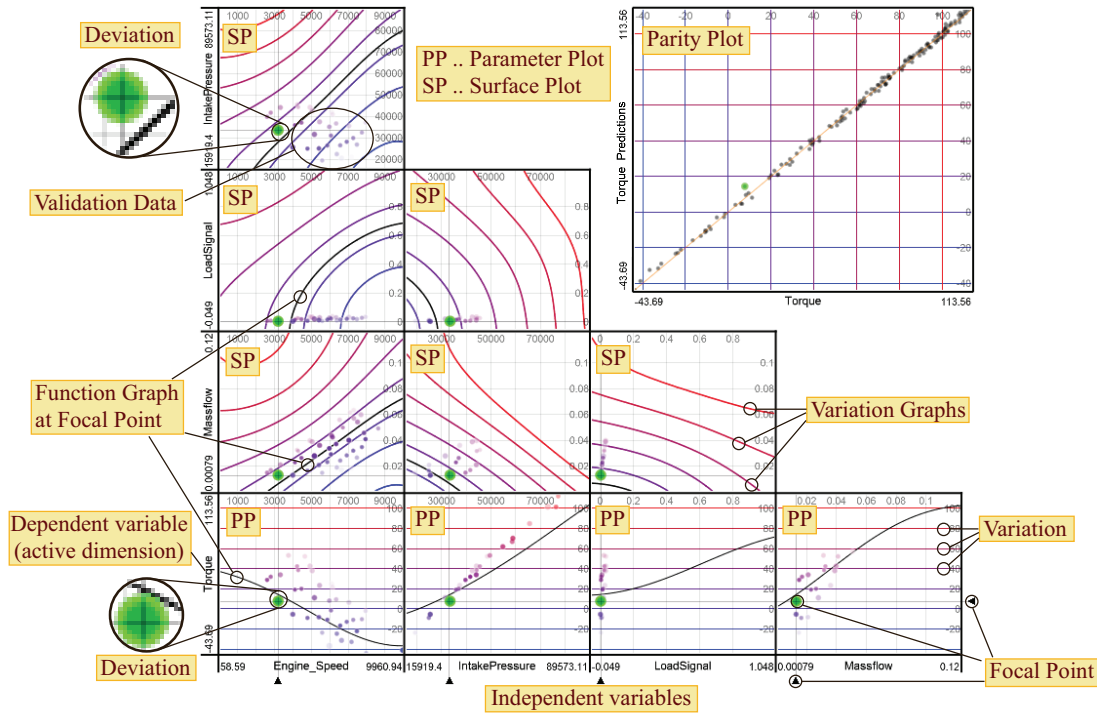


Figure 1: The layout of HyperMoVal for a real model predicting torque given four parameters. The focal point F is set to a validation data point with a significant deviation. The matrix contains all paraxial 2D slices at F in the 5D model space.

Moreover, the identification of regression models based on techniques from machine learning is typically an iterative process [BL04]. Each iteration involves parameter definition, training, and validation. Supporting this entire workflow is an important topic which goes far beyond the mere visualization of functions. Recent work shows significant progress towards supporting the process of model building using interactive visualization, but this work is restricted to particular models like clusters [NHM*07] or linear trends [GWR09]. There is no approach to support the identification and validation of general regression models according to the design goals listed in Sec. 1.1.

3. Interactive Model Validation

This section introduces HyperMoVal as our approach to an interactive visual validation of surrogate models. We first describe the combined visual encoding of model and validation data before discussing the supported interaction techniques.

3.1. Visual Encoding

The key idea of our approach is to provide a combined visualization of a regression model and known validation data in order to assess the match between both. The model is an analytically given scalar function with n independent variables and the validation data is given as an arbitrary number of $n+1$ -tuples, i.e., a value for each parameter of the function and one for the result. In order to visualize such potentially high-dimensional data, we layout multiple projections

to low-dimensional space as paraxial slices in a similar way as HyperSlice [vWvL93]. This approach treats all dimensions equally and supports the creation of plots which are familiar to engineers. An implication is to maintain a focal point F as $n+1$ -tuple, which defines values $F_i, i \in \{1..n+1\}$ for all dimensions not shown by a particular plot. F may take any position inside an $n+1$ -dimensional hyperrectangle S representing the space considered for visualization. S is defined as the Cartesian product of intervals defined for each of the $n+1$ dimensions. S is initially set to contain all values of the validation data, but its extents can be modified by the user (see Sec. 3.2).

Each pair of dimensions defines one plot (see Fig. 1). The result dimension of the function specifies the y-axis of the bottom row of the matrix. These plots (called *parameter plots*) thus display the explicit function graph for all independent parameters. All other plots (called *surface plots*) show pairs of independent variables, where the function is represented as contour at the iso-value specified by F . Only the lower triangle of the matrix is displayed for performance reasons and to gain free space for other purposes.

In addition to the function graphs of the model, each plot also displays projections of the validation data as points. In order to support a quick identification of badly approximated regions, the *size* of each point reflects its absolute residual to the prediction by the model at the particular position in parameter space. Large points indicate regions with a significant deviation whereas small points suggest a good match.

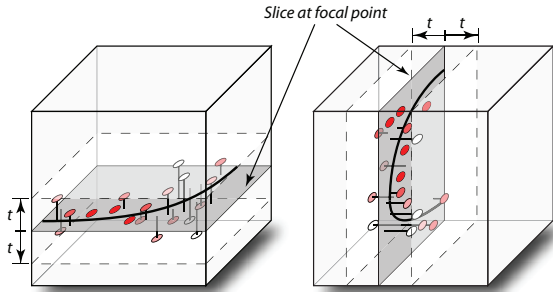


Figure 2: The region around two slices in which points are considered relevant for the respective plot. The color intensity depends on the distance to the slice.

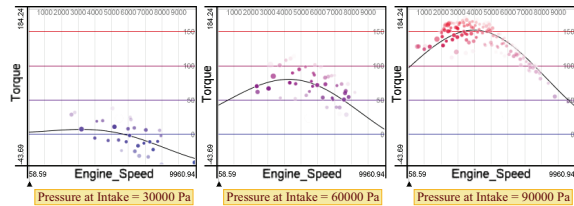


Figure 3: Altering F in one dimension changes the function graph and the relevance of data points for all plots not displayed in that dimension. Continuous modifications make the points fade in and out smoothly around the graph.

Although the precision conveyed by size is lower than for other visual attributes, it is sufficient for the purpose of assessing the prediction quality.

3.1.1. Task-Specific Point Relevance

As each plot is a slice through the visualization space S at F , it is important to consider which points should be displayed by a plot. A precise assessment of the fit by a model is only possible for validation points on that slice. One approach could thus hide all other points. In this case, at most one point will typically be visible. An important exception, though, is validation data which is structured as – possibly unequally spaced – n -dimensional grid. This is a common case in the design of experiments (e.g., full-factorial designs). Provided that F is set to a point of the grid, parameter plots then show all points in the same line in the respective dimension. 2D surface plots, however, require a precise match in the result dimension and typically display F only (this is different for 3D plots as discussed below).

In general, it is often reasonable to additionally display points close to (but not on) the slice of a plot, e.g., to provide context information for navigation. It may even be necessary to show all points in order to assess the coverage of the parameter space by validation data like in a normal scatterplot matrix. For this reason, HyperMoVal supports a task-specific definition of what is considered a *relevant range* around each slice (see Fig. 2). Formally, let P denote a point inside the $n+1$ -dimensional hyperrectangle S and let V be a 2D visual-

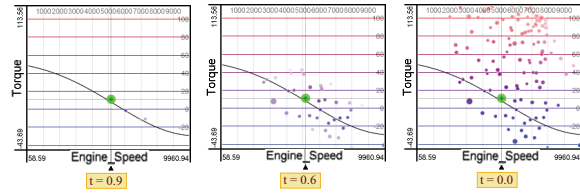


Figure 4: Altering the relevance threshold t changes the shown subset of data points around the visualized slice.

ization of the dimensions x_j and x_k , then the relevance r of P with respect to V is defined as

$$r_V(P) = 1 - \text{Max}_{i \in \{1..n \setminus x_j, x_k\}} \left(\frac{|P_i - F_i|}{|S_i|} \right)$$

where S_i denotes the length of S in the i -th dimension. After computing $r_V(P)$, a linear mapping determines the saturation with which P is drawn. This mapping is controlled by a threshold t with $0 \leq t < 1$ so that $r_V(P) \leq t$ means white (i.e., the background color) and 1 means maximal saturation. The transition conveys the distance to the slice and generates a smooth fading effect when changing F , as illustrated by Fig. 3. Modifying t enables to define relevance depending on the task (see Fig. 4): values close to (or at) 1 show only a narrow range around the slice to precisely support a visual assessment of the fit. For values close to 0, our approach resembles a normal scatterplot matrix.

3.1.2. Sensitivity Analysis

Another important task is to analyze the sensitivity of the surrogate model with respect to changes in single dimensions. To support this task, one dimension can be set active at a time, which has two effects: 1) each plot displays a family of function graphs (called *variation graphs*) for a user-defined number of equally sized steps along the displayed range of the active dimension; 2) the active dimension determines the hue of validation data points and variation graphs via a transfer function. By default, we use a transition from blue to red. This ensures a good luminance contrast to the white background and it is perceptually distinct from the modulation of saturation by point relevance. Users may also choose other transfer functions.

Coupling variation and coloring has proven beneficial with respect to usability. The main reason is that grid lines of plots showing the active dimension define meaningful steps for variation graphs and may also be used as legend for the color coding. The most frequent – and default – case is to set the result dimension active, which generates multiple iso-contours of the same surface in each surface plot (as in Fig. 1). Setting a function parameter as active dimension generates multiple explicit graphs in all other parameter plots. It also shows contours at the same iso-value for multiple surfaces in each surface plot that does not display the active dimension (Fig. 5). Mapping the active dimension to hue emphasizes relations between points and graphs with respect to this dimension. It also reveals deviations when points are afar from graphs despite having the same hue.

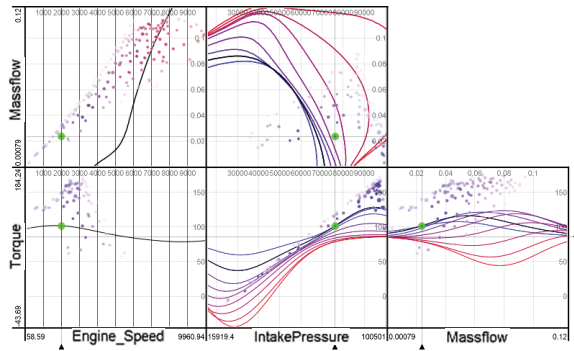


Figure 5: Varying the parameter "Engine Speed" generates a family of iso-contours and explicit function graphs.

3.1.3. 3D Visualization

While all plots are described as 2D so far, surface plots optionally provide a 3D visualization as well. In this case, the result dimension of the model is mapped to height (see Fig. 6). The validation data is shown as a 3D scatterplot of points which are scaled and colored as described for the 2D case. The computation of relevance excludes the result dimension in order to display points around the entire surface. Being less restrictive, more data points are thus shown in 3D than in 2D. The function of the surrogate model is represented in multiple ways: iso-contours are generated for F and all variation graphs of the active dimension as line strips similar to the 2D case, except that each contour is located at the height of the respective iso-value.

In addition, the surface of the surrogate model for F is either shown as wireframe or as shaded, opaque surface. While the wireframe representation avoids problems caused by occlusion and thus keeps all validation points visible, opaque surfaces convey a better 3D impression at the cost of occluding points either above or beneath the surface – depending on the viewing position. When switching between 2D and 3D, a smooth transition of the view point avoids change blindness in a similar way as described by Elmqvist et al. [EDF08]. The main reason for integrating a 3D visualization is the great popularity of this representation with engineers in our target domain. Initial observations have shown that 3D plots are particularly used for assessing models with few parameters and for checking physical relationships between parameters. 2D surface plots are rather used for studying abstract relationships for high-dimensional models.

3.1.4. Parity Plot

Differences between predicted and known results are also shown in a separate scatterplot (see Fig. 1). This plot maps known results to the X-axis and predicted results to the Y-axis for each point of the validation data. Such plots are called *parity plots* and are common in many fields of science and engineering. A straight line at the main diagonal indicates a good match whereas deviations to either side convey

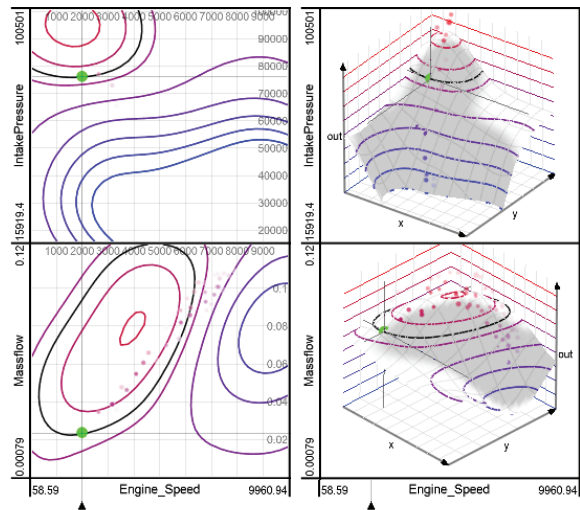


Figure 6: 2D and 3D visualizations of the same surface plots. Due to the additional visual dimension, the set of relevant points is typically larger in 3D.

the amount and the sign of residuals. The thickness of the covered area around the main diagonal provides an overview of the overall fit and outliers remain visible. Both types of information could also be conveyed differently (as shown in later sections). The reasons for integrating parity plots are the equal scaling of both axes and the popularity of this plot with experts in our target field of application.

3.1.5. Comparison of Multiple Models

Another design goal concerns the *comparison* of multiple regression models predicting the same result dimension. The independent variables may be different, in which case our approach represents the union of the independent variables of all visualized models. Each plot (except the parity plot) shows function graphs or iso-contours for each model that covers the dimensions of both axes. Different line stippling is used to discriminate the graphs of multiple models (see Fig. 7). Deviations, however, will in general be different for multiple models. In order to avoid overloading the visualization, the point size thus refers to the residuals with respect to one active model while disregarding all others. Analogously, the parity plot is only shown for the active model and 3D surface plots show the – wireframe or opaque – surface of the active model only. For these reasons, other models are rather context information when assessing the match between the active model and the validation data. However, visualizing multiple models is particularly useful when comparing their behavior outside the space covered by validation data (e.g., when assessing the plausibility of extrapolations).

3.2. Interaction

Interaction is an important aspect of HyperMoVal to support different tasks. *Modifying the focal point F* enables to navigate the visualization space S . We discriminate local and

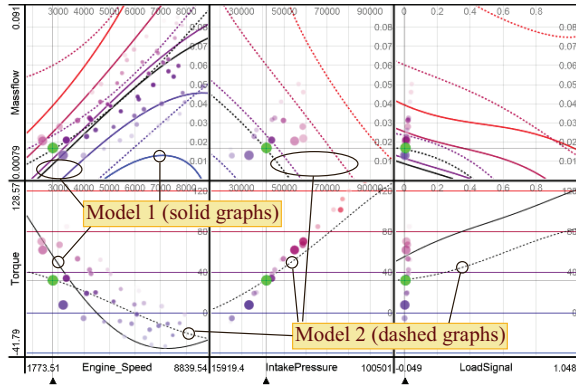


Figure 7: Comparison of two models predicting torque. The models are discriminated by the line style (dashed and solid) and show a very different prediction quality at F . "Intake-Pressure" is a parameter of Model 2 only.

global modifications: *local modifications* affect a subset of the dimensions at a time. They involve altering the value of a single dimension by dragging handles at the border of the matrix or altering two dimensions by moving a crosshair representing F in any 2D plot. *Global modifications* define values for all dimensions at once by setting F according to a particular point. This is done by clicking on a visible validation point in any 2D plot including the parity plot. It supports a quick navigation to points with a significant residual.

As another type of interaction, the extents of S can be modified individually for each dimension. Similar to the Projection matrix [TSDS96] and dynamic filtering [AS94], narrowing the range of one dimension removes all validation points outside this range in all plots. Changing the range also allows the user to analyze every part of the function in as much detail as necessary or to scroll across function graphs. Extending ranges to contain space outside the region covered by validation data is crucial in our context to assess the plausibility of *extrapolations* by a model.

Further interactions which are not discussed in detail involve changes of the viewpoint in 3D plots, altering the threshold t as discussed in Sec. 3.1, and setting functions and dimensions to be active. When validating models with many parameters, it is helpful to temporarily hide certain independent parameters and thus gain more space for the rest.

4. Integrated Workflow for Model Identification

HyperMoVal as described in Sec. 3 can be regarded a design study that is conceptually independent of properties of a surrounding software system. A stand-alone implementation of HyperMoVal already fulfils all design goals except the last of Sec. 1.1. On the other hand, it is not always practical for a single view to cover all potentially relevant tasks during a complex workflow like model building. Fortunately, linking multiple views is agreed to be a solution to this problem. This section therefore describes the integration of Hy-

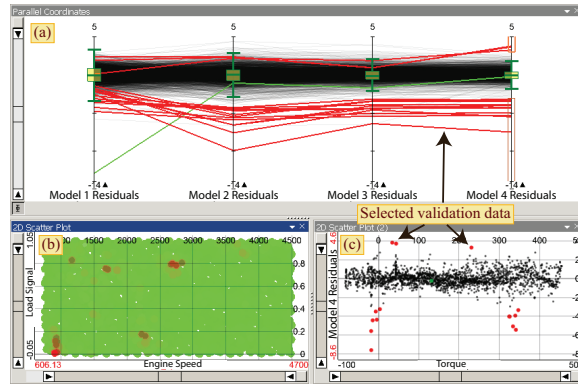


Figure 8: Linked views of derived model attributes: (a) comparing the deviations of different models by brushing large residuals; (b) mapping absolute deviations of one model to color in a 2D representation of the parameter space; (c) plotting residuals (Y-axis) against known results (X-axis).

perMoVal within a system of multiple linked views. We also discuss how to use different views to support the entire workflow of model identification. It should be emphasized, however, that the focus of this paper is on model validation while many aspects related to model identification are up to future work (see Sec. 7).

Our integration is based on two concepts: 1) defining arbitrary subsets of validation data in any view and 2) representing predicted results and residuals of a particular model as derived data attributes. Besides HyperMoVal, the types of involved views may include most standard visualizations for multivariate data, for example scatterplots and parallel coordinates. Their purpose is to visualize the validation data and to support the definition of arbitrary subsets thereof based on view-specific brushing techniques like rubberband- or lasso-selections. HyperMoVal supports restricting the set of validation data points to any subset defined by brushing other views. This enables to filter validation data that is implausible (e.g., incorrect measurements) or undesirable due to design constraints. Unlike the filtering offered by HyperMoVal, such selections may include dimensions which are not covered by any model and they may be based on composite queries of arbitrary complexity [Wea09].

As second concept, predicted results and residuals of a particular model are added as additional data dimensions for each point of the validation data. Such *derived data columns* are available for view parameterization like any attribute of the validation data itself. This offers a variety of possibilities: for example, residuals of multiple models may be compared using parallel coordinates (Fig. 8a). They may indicate the amount of deviation in a scatterplot relating model parameters (Fig. 8b). Other views may provide statistics as quantitative results, e.g., the maximal and the average prediction error of a model for a certain subset of the validation

data [PBH08]. Derived columns may also be used as criterion for selections as discussed above.

Validation is only a part of the overall workflow of model identification [Sar94]. In our application context, the first step is the definition of all information that is necessary for training the model. This involves selecting dimensions as input parameters, specifying model and training parameters, and choosing the training data itself. After the (automatic) training, the model needs to be validated and potentially compared to other models as supported by our approach. The goal of this validation is to determine whether a particular model is appropriate, or whether further refinements are necessary. In the latter case, an additional iteration starts by changing the set of input dimensions, modifying training parameters, and performing another training [BL04].

Our approach to support this highly iterative process is based on integrating parameter specification, training, and validation within a single tool. This has two benefits: 1) brushing various views as discussed above can also be used to define the set of training data interactively, e.g., to filter wrong results based on domain knowledge of the user. 2) Changing a certain model by executing a new training immediately updates the information related to this model in all views. This includes function graphs in HyperMoVal, derived attributes like residuals in all views, and it triggers a re-evaluation of selections which are based on derived attributes of the model. The advantage of this tight integration is the speed at which different settings of training parameters can be tried and compared to each other, which may reduce the necessary time of the overall workflow significantly.

5. Implementation

HyperMoVal and the workflow described in Sec. 4 have been implemented in *visplore*, a system for visual exploration, which offers different visualizations (e.g., 2D and 3D scatter plots, parallel coordinates, histograms, etc.) that can be linked by ad-hoc selections and derived data columns. All parts are written in C++ and use OpenGL for rendering.

HyperMoVal supports immediate visual feedback during continuous user interactions like changing the focal point F . For this purpose, it implements the multi-threading architecture as described in previous work [PTMB09]. When drawing function graphs, a dedicated visualization thread samples the model progressively. This provides an immediate preview while generating the final image including variation graphs in each plot may take a few seconds on standard PCs for models of more than two parameters. Each plot also layers the visualization in order to prioritize the visualization of F and to cache and reuse visual results in image space.

Internally, models are objects which implement an interface that allows the integration of any type of regression model. Currently, our implementations supports support vector regression (SVR), and it uses the library LIB-

SVM [CL] both for model evaluation and training. The user interface for defining training parameters is a simple dialog.

6. Evaluation

To evaluate our approach on multiple levels [Mun09], this section first describes an application scenario to illustrate a typical workflow in car engine design. We then report general feedback collected by interviewing application experts of the target domain. This evaluation has been done in collaboration with three experts in the field of engine design, whose educational background is in mechanical engineering and industrial mathematics. As employees of AVL List GmbH, a company providing hardware and software for the development of powertrain systems, their responsibility is partly customer support and training, and partly the development of the simulation core. All of them are experienced users of our approach who have been testing, using, and training it for months.

6.1. Application Scenario

The goal of the scenario is to identify a surrogate model predicting torque based on the results of 400 simulation runs of a real-world car engine. This example uses the same data for training and validation. Knowing that torque is primarily a function of engine speed and load signal, the first step is to check whether a 2D model on these attributes already provides sufficient accuracy. Statistical summaries (first row of Fig. 9e), however, indicate a large mean deviation and a substantial maximal error. An analysis in HyperMoVal (Fig. 9a) reveals an outlier in the data. The engineer explains this point as non-converged simulation run and decides to exclude it from further steps. Besides the outlier, large points in regions near steep changes in torque generally reveal the model as insufficiently accurate.

In the next iteration, the engineer decides to increase the complexity of the model by adding the pressure at the intake manifold (pOIM) as third parameter which is known to impact torque. While this new model turns out to be much better, statistics still show a considerable maximal deviation (second row of Fig. 9e). Brushing absolute residuals larger than 5 Newton meters (Nm) in a histogram (Fig. 9b) reveals that the main problem is a jump in torque at low values for both speed and load. As this insight still does not explain, why the match is bad in this region, the engineer uses HyperMoVal to analyze the local model behavior in detail (Fig. 9c). This shows that the gradient of the model is not sufficient to approximate the sudden jump in torque. A potential explanation is that pOIM does not contribute information for this region since the values for pOIM are very similar as shown by the bottom right plot of Fig. 9c.

The engineer decides to add one more dimension as model parameter. Since there are multiple physically meaningful candidates, ranking them by the amount of correlation to torque facilitates the selection (Fig. 9d). For this purpose,

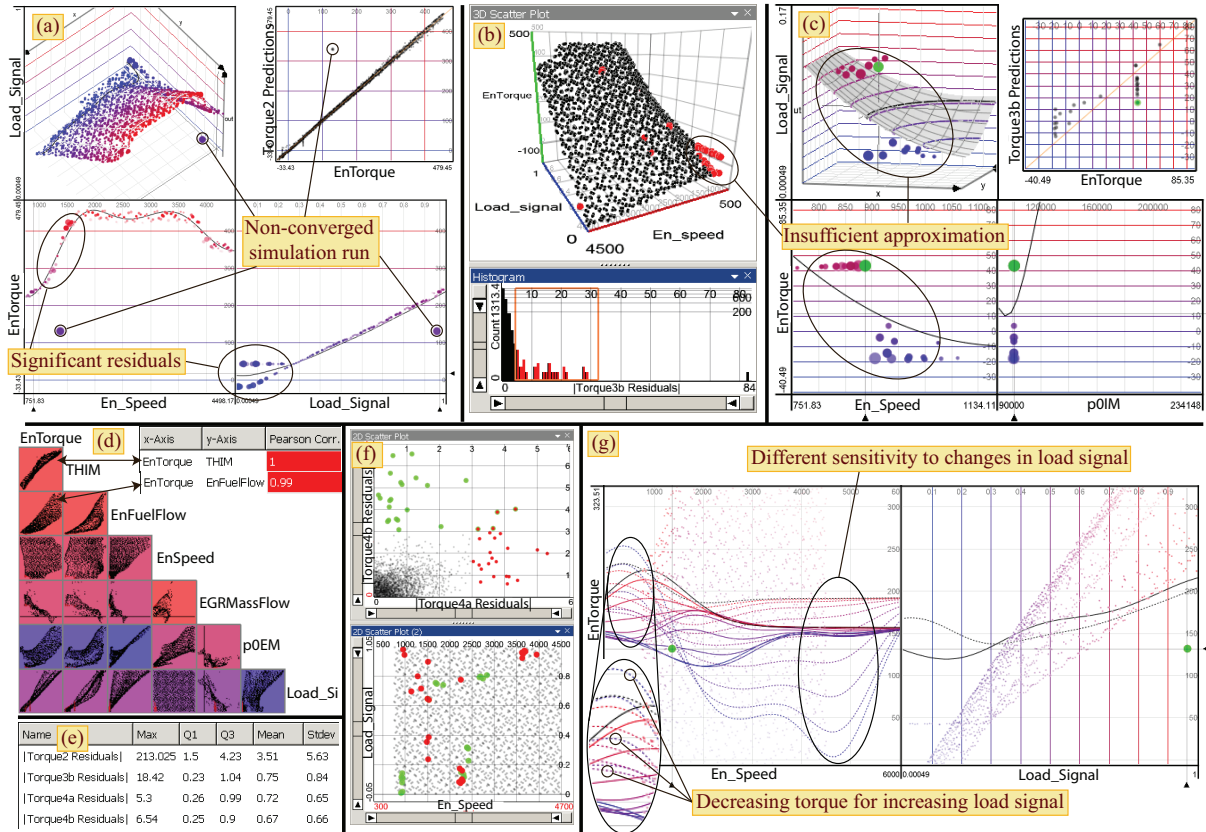


Figure 9: An exemplary workflow for model building. (a) Assessing a 2D model also reveals implausible validation data; (b + c) a 3D model still does not sufficiently cover a jump in torque for low values of speed; (d) ranking potential input parameters by means of correlation to the result; (e) quantifying multiple models with respect to their residuals (in Nm); (f) comparing residuals for two model candidates; (g) comparing the candidates with respect to the plausibility of extrapolations.

the engineer brushes the validation points in the mismatching region; he then opens a scatterplot matrix that is ranked and colored by the correlation of the selected points to torque [PBH08]. Having correlation coefficients close to 1, the temperature at the intake manifold (THIM) and fuel mass flow (EnFuelFlow) turn out to be equally suitable candidates.

In order to try both variants, the engineer creates two further models: The model "Torque4a" has THIM as fourth parameter while "Torque4b" depends on EnFuelFlow. The statistical quantification shows that both models are better than the previous ones in particular with respect to the maximal error which can now be considered sufficiently small (Fig. 9e). Brushing residuals greater than 3 Nm for both models in a scatterplot shows an error distribution that is slightly more clustered for model "Torque4b" (green points) in a linked view plotting speed against load signal (Fig. 9f).

As final step, the engineer uses HyperMoVal to compare the plausibility of both models (Fig. 9g) outside the range covered by training data. He extends the displayed interval for speed to 0 to 6000 rotations per minute. He also sets the focal point to the non-converged simulation run to ensure

that the models compensate for this error. Setting load signal as active dimension generates variation graphs in steps of 10 percent. Despite their comparable error statistics, the shapes of the function graphs differ significantly. Unlike "Torque4a" (solid graphs), the model "Torque4b" (dashed graphs) turns out to be highly sensitive to changes in load signal at high speeds, which is considered undesirable. Even worse, "Torque4b" predicts significant decreases in torque for increasing load signal at low speeds, which is implausible regardless of the values of other model parameters (like pOIM). As a result, "Torque4a" is the better choice.

6.2. User Feedback

Based on their own experiences and on the feedback gained from customer training, the overall feedback of our interviewees was very positive. Their former workflow for model identification involved an in-house tool and standard software in engineering like Matlab [Mat]. Compared to these software packages, our interviewees appreciated that HyperMoVal supports analyzing particular questions and changing scenarios instantaneously, e.g., skimming through the model behavior for different validation points. In their experience,

standard software like Matlab requires a setup-time of several minutes for writing and modifying some lines of code for each step of an analysis, and they provide limited interactivity for their visualizations. As a result, HyperMoVal and its integration with other views allows engineers to validate models in detail on average one magnitude faster. As an application expert stated, "*the design [of HyperMoVal] and the high interactivity encourage the analysis of reasons for mismatches. It also facilitates comparisons between models which would otherwise have been omitted in many cases due to the involved effort. Being able to validate a model in detail before usage significantly increases the quality and the confidence in the entire process.*"

An objection was that most engineers are still not used to the concept of interaction. While the improvements in speed (e.g., when rotating a 3D surface) have been noticed immediately, most users tend to map their old workflow to the new tool. This particularly applies to the concept of linking different views as discussed in Sec. 4, which may require several days for familiarization. On the other hand, features which are common in other engineering applications are also expected in new approaches and thus required to gain acceptance. For HyperMoVal, this includes the support for 3D plots, parity plots, and the ability to change the transfer function to a rainbow color map, which are still popular with engineers despite their known flaws [BT07]. Perhaps the most challenging aspect of the design process was thus to find a good balance between preserving familiar methods and introducing techniques that are powerful yet unfamiliar to engineers (e.g., linked views). Nevertheless, our interviewees confirmed that all features of HyperMoVal are useful for specific tasks as discussed in former sections.

Our approach is distributed as part of the software suite of the company AVL List GmbH. As such, it is potentially available to several thousand users. On the other hand, the application of surrogate models is at the beginning and reliable evidence about adoption rates is not yet available. First estimates assume an application by 5 to 10 companies.

7. Discussion and Future Work

The validation of regression models comprises *three levels of detail* as reflected in the application scenario of Sec. 6.1: (1) Statistical summaries of the entire validation data provide a coarse yet compact information about the global prediction quality. (2) Derived attributes like residuals describe the quality as one value per data point. They enable the identification of local characteristics like badly fitted regions, but are often insufficient to explain such characteristics. (3) A combined visualization of validation data and function graphs of the model provides most information for a certain point of the parameter space. The shape and the gradient of the function offer detailed reasons yet with very local scope.

HyperMoVal is designed to support domain experts at this latter level, while the integrated workflow described in Sec. 4

serves the first two levels. Stand-alone implementations of HyperMoVal may decide to additionally cover the levels 1 and 2 by integrating quantitative model characteristics like the statistics of Fig. 9e or additional options for coloring, e.g., mapping residuals to color as in Fig. 8b. While respective extensions would be straightforward, our implementation employs linked views for these tasks.

Concerning *scalability*, HyperMoVal has been tested with several thousand validation data points. Multi-threading and the use of graphics hardware ensure immediate visual feedback. Interaction concepts like narrowing the relevant region support the perceptual scalability. HyperMoVal has also been tested for models with more than ten parameters, although such surrogate models are rare in practice. In this case, hiding some dimensions is usually tolerable and provides more space for the rest. An upper limit with respect to the number of compared models is four due to the difficult distinction of function graphs and a significant cluttering.

We believe that HyperMoVal is a good example for combining computation and interactive visualization to support a complex task, which is a main issue of Visual Analytics. The optional integration to multiple linked views fosters a tight loop of computation-based training and visualization-based validation of regression models. Splitting the data into training and validation data is an important aspect in practice [Sne77] and is supported using selections.

However, further steps towards model identification are a key aspect of *future work*. In particular the identification of good training parameters is an optimization problem on its own. Currently, users with little background in machine learning (as are most designers of car engines) need to manually try many combinations of parameters like the cost value or the gamma value of the SVR. This is neither efficient nor does it ultimately guarantee an optimal set of training parameters. Semi-automatic approaches could integrate optimization techniques to quickly identify promising training parameters. Other plans for future work include a long-term field study of the adoption by different groups of engineers, as well as further navigation concepts for the focal point. Finally, we intend to evaluate our approach within other application areas to assess its general applicability.

8. Conclusion

This paper introduced HyperMoVal as interactive visualization to support various tasks during the validation of regression models. The combined visualization of the n -dimensional model and the validation data provides a direct comparison of known and predicted results and it enables to analyze regions with a bad fit in detail. The simultaneous analysis of families of graphs for multiple models helps to assess and compare the physical plausibility of the function behavior. The matrix-based layout of 2D and 3D slices scales to high-dimensional functions, which can easily be navigated using different interaction techniques. Providing

model-related attributes like residuals as derived dimensions optionally complements the analysis in other multivariate views. Linking these views via ad-hoc queries may not only be used to filter implausible validation data; it also supports a user-defined selection of training data as a first step towards a workflow for model building that tightly integrates domain knowledge. User feedback suggests that our approach significantly accelerates the identification of high-quality surrogate models for simulating engine physics in real-time. Application experts consider it an important technique for increasing the confidence in the entire process of car engine design. Motivated by these results, we believe that HyperMoVal may also be beneficial to numerous other application areas in science and engineering.

9. Acknowledgements

This work has been supported by the Austrian Funding Agency (FFG) within the scope of the project AVISOM (No. 818060). Thanks go to Johann Wurzenberger and Ivo Prah for supporting the evaluation and providing important feedback, and to M. Eduard Gröller and Krešimir Matković for constructive comments and discussion.

References

- [AS94] AHLBERG C., SHNEIDERMAN B.: Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. In *CHI '94: Proc. of the SIGCHI Conf. on Human Factors in Computing Systems* (1994), ACM, pp. 313–317. 6
- [BDHK05] BOECKING F., DOHLE U., HAMMER J., KAMP-MANN S.: Passenger Car Common Rail Systems for Future Emission Standards. *Motortechnische Zeitschrift (MTZ)* 66, 7–8 (2005), 552–557. 1
- [BL04] BERRY M., LINOFF G.: *Data Mining Techniques, 2nd Edition*. Wiley, 2004. 3, 7
- [BT07] BORLAND D., TAYLOR R.: Rainbow Color Map (Still) Considered Harmful. *IEEE Computer Graphics and Applications* 27, 2 (2007), 14–17. 9
- [CL] CHANG C., LIN C.: LIB-SVM. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, Last visited on Feb. 17th 2010. 7
- [Cle93] CLEVELAND W.: *Visualizing Data*. Hobart Press, 1993. 2
- [dSB02] DOS SANTOS S., BRODLIE K.: Visualizing and Investigating Multidimensional Functions. In *VISSYM '02: Proc. of the Symposium on Data Visualisation* (2002), Eurographics Association, p. 173 pp. 2
- [EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the Dice: Multidimensional Visual Exploration Using Scatterplot Matrix Navigation. *IEEE Trans. on Visualization and Computer Graphics* 14, 6 (2008), 1539–1148. 5
- [FB90] FEINER S., BESHERS C.: Worlds within Worlds: Metaphors for Exploring n-Dimensional Virtual Worlds. In *UIST '90: Proc. of the 3rd ACM SIGGRAPH Symposium on User Interface Software and Technology* (1990), ACM, pp. 76–83. 2
- [GTC01] GRINSTEIN G., TRUTSCHL M., CVEK U.: High-Dimensional Visualizations. In *Workshop on Visual Data Mining, 7th Conf. on Knowledge Discovery and Data Mining (KDD)* (2001), pp. 77–87. 2
- [GWR09] GUO Z., WARD M., RUNDENSTEINER E.: Model Space Visualization for Multivariate Linear Trend Discovery. In *VAST '09: Proc. of the 2009 IEEE Symposium On Visual Analytics Science And Technology* (2009). 3
- [Hei09] HEINZLE R.: *Machine Learning Methods and Their Application to Real-Time Engine Simulation*. Dissertation, Johannes Kepler Universität Linz, 2009. 1
- [HTF09] HASTIE T., TIBSHIRANI R., FRIEDMAN J.: *The Elements of Statistical Learning, 2nd Edition*. Springer, 2009. 1
- [Ins85] INSELBERG A.: The Plane with Parallel Coordinates. *The Visual Computer* 1, 2 (1985), 69–91. 2
- [JN02] JAYARAMAN S., NORTH C.: A Radial Focus+Context Visualization for Multi-Dimensional Functions. In *VIS '02: Proc. of the IEEE Conf. on Visualization* (2002), IEEE Computer Society, pp. 443–450. 2
- [Mat] MATHWORKS T.: Matlab. <http://www.mathworks.com/>, Last visited on Feb. 17th 2010. 8
- [MJJK05] MATKOVIC K., JELOVIC M., JURIC J., KONYHA Z.: Interactive Visual Analysis and Exploration of Injection Systems Simulations. In *Proc. of the IEEE Conf. on Visualization 2005* (2005), IEEE Computer Society, pp. 391–398. 1
- [MTS91] MIHALISIN T., TIMLIN J., SCHWEGLER J.: Visualizing Multivariate Functions, Data, and Distributions. *IEEE Computer Graphics and Applications* 11, 3 (1991), 28–35. 2
- [Mun09] MUNZNER T.: A Nested Model for Visualization Design and Validation. *IEEE Trans. on Visualization and Computer Graphics* 15, 6 (2009), 921–928. 7
- [NHM*07] NAM E., HAN Y., MUELLER K., ZELENYUK A., IMRE D.: Clustersculptor: A Visual Analytics Tool for High-Dimensional Data. In *VAST '07: Proc. of the 2007 IEEE Symposium On Visual Analytics Science And Technology* (2007), IEEE Computer Society, pp. 75–82. 3
- [NSSV09] NOUANESSENGSY B., SEOK S., SHEN H., VIELAND V.: Using Projection and 2D Plots to Visually Reveal Genetic Mechanisms of Complex Human Disorders. In *VAST '09: Proc. of the 2009 IEEE Symposium On Visual Analytics Science And Technology* (2009). 2
- [PBH08] PIRINGER H., BERGER W., HAUSER H.: Quantifying and Comparing Features in High-Dimensional Datasets. In *International Conf. on Information Visualisation (IV08)* (2008), IEEE Computer Society, pp. 240–245. 7, 8
- [PTMB09] PIRINGER H., TOMINSKI C., MUIGG P., BERGER W.: A Multi-Threading Architecture to Support Interactive Visual Exploration. *IEEE Trans. on Visualization and Computer Graphics* 15, 6 (2009), 1113–1120. 7
- [Sar94] SARGENT R.: Verification and Validation of Simulation Models. In *WSC '94: Proc. of the 26th Conf. on Winter simulation* (1994), Society for Computer Simulation International, pp. 77–87. 2, 7
- [Sne77] SNEE R.: Validation of Regression Models: Methods and Examples. *Technometrics* 19, 4 (November 1977), 415–428. 2, 9
- [TSDS96] TWEEDIE L., SPENCE R., DAWKES H., SU H.: Externalising Abstract Mathematical Models. In *CHI '96: Proc. of the SIGCHI conference on Human factors in computing systems* (1996), ACM, p. 406 pp. 6
- [vWvL93] VAN WIJK J., VAN LIERE R.: HyperSlice: Visualization of Scalar Functions of Many Variables. In *Proc. of the 4th Conf. on Visualization* (1993), IEEE Computer Society, pp. 119–125. 2, 3
- [Wea09] WEAVER C.: Conjunctive Visual Form. *IEEE Trans. on Visualization and Computer Graphics* 15, 6 (2009), 929–936. 6