



Surrogate model–based inverse parameter estimation in deep drawing using automatic knowledge acquisition

Matthias Ryser¹ · Felix M. Neuhauser² · Christoph Hein³ · Pavel Hora² · Markus Bambach¹

Received: 22 March 2021 / Accepted: 5 July 2021 / Published online: 4 August 2021
© The Author(s) 2021

Abstract

In this paper, we propose a new approach for the simulation-based support of tryout operations in deep drawing which can be schematically classified as automatic knowledge acquisition. The central idea is to identify information maximising sensor positions for draw-in as well as local blank holder force sensors by solving the column subset selection problem with respect to the sensor sensitivities. Inverse surrogate models are then trained using the selected sensor signals as predictors and the material and process parameters as targets. The final models are able to observe the drawing process by estimating current material and process parameters, which can then be compared to the target values to identify process corrections. The methodology is examined on an Audi A8L side panel frame using a set of 635 simulations, where 20 out of 21 material and process parameters can be estimated with an R^2 value greater than 0.9. The result shows that the observational models are not only capable of estimating all but one process parameters with high accuracy, but also allow the determination of material parameters at the same time. Since no assumptions are made about the type of process, sensors, material or process parameters, the methodology proposed can also be applied to other manufacturing processes and use cases.

Keywords Deep drawing · Regression analysis · Inverse identification · Digital twin · Surrogate modelling

1 Introduction

Deep drawing is one of the most used production processes in sheet metal forming and is used in a wide range of industries [1]. L.-E. Elend estimates that manual fine-tuning after tool manufacturing, called tryout, consumes up to 30% of the overall tool cost [2], which is confirmed by S. Schulte [3]. Birkert et al. estimate tryout cost to 32%, but required time to 44% compared to cost and time, respectively, for the overall tool [4]. Therefore, extensive work has been made to improve tool tryout using new methodologies.

The main modification during the tryout phase lies in the spotting of the tools to reach an approximately homogeneous pressure distribution between the tools and

the part. Essig et al. [5] used optical measurement and image processing methods to evaluate the thickness distribution of the blue color during spotting and were able to detect surface areas in contact by a quantitatively measurable and standardized method. Braedel [6] proposes a method to determine the individual forces in the hydraulic cylinders acting on the drawing cushion using structural FE analysis and therefore adjusting the pressure distribution between blank holder and die to reduce the number of iterative loops during tryout.

Wang et al. [7] identify the draw-in as the most important parameter during deep drawing which controls all forming characteristics like stresses and strains, drawing failures and surface defects. Hence, the authors propose the use of a simulation-based draw-in map which is used as reference during fine-tuning of the tools. In contrast, A. Prexl [8] found matching draw-ins necessary, but not sufficient and thus proposes a strain field–based procedure to predict manual tryout operations purely based on simulations by identifying reasonable fine-tuning operations based on their sensitivities on quality criteria. Harsch et al. [9] used a comparable procedure to virtually plan tryout operations on a trunk lid. The authors used the same procedure in [10–15] to find robust process windows, which can be

✉ Matthias Ryser
ryser@ivp.mavt.ethz.ch

¹ Advanced Manufacturing Group, ETH Zurich, Technoparkstrasse 1, 8005 Zurich, Switzerland

² Institute of Virtual Manufacturing, ETH Zurich, Technoparkstrasse 1, Zurich 8005, Switzerland

³ Audi AG, Auto-Union-Str. 1, Ingolstadt 85057, Germany

viewed as a similar task. Finally, AutoForm Engineering GmbH developed the TryoutAssistant [16] which uses AutoForm-Sigma simulations to calculate the sensitivities of the process parameters on the draw-in and proposes parameter corrections to reach the desired draw-in [17].

The use of the flange draw-in during drawing to infer part quality leads to the question where exactly the draw-in has to be measured. Fischer et al. [18–20] performed stochastic finite element simulations and applied correlation analysis between the draw-in and defined quality criteria in order to identify sensitive draw-in positions with respect to the quality of the final part. In this approach, the correlation between draw-ins at different locations is not taken into account which tends to lead to an excessive number of draw-in sensors. Neuhauser et al. [21, 22] propose the use of column subset selection to identify suitable sensor positions by selecting the most independent measurement locations regarding their sensitivities on quality criteria.

In summary, available tools and methods still lack a fully automated workflow to determine suitable process modifications during tryout of deep drawing tools. In this work, we propose a new concept for the support of tryout operations based on automatic knowledge acquisition [23]. The core of our methodology consists of a stochastic finite element simulation of the deep drawing process, which is used for regression analysis to learn the inverse relationship between process output and input in the form of a digital twin [24]. The regression models obtained can then be applied on a real world part to reconstruct current process parameters and infer parameter modifications by comparison with the nominal values from the robust optimum. Regarding the flow of information, this approach therefore looks very similar to the manual decision making made by humans. As a difference, the decision-making will be put in a purely mathematical framework. The two core questions that arise for the use of our approach and which will be answered for a case-study are the following:

1. Where do sensors need to be placed to maximise measured information content about the process?
2. Which process parameters can be observed by using the determined sensor positions?

In contrast to the approaches used in [25–27] to inversely determine material parameters by using numerical optimization of a FEM simulation in combination with experiments, our work aims to find a surrogate model that directly maps from the process output to its input. The most similar work to the best knowledge of the authors could be found by Senn and Link [28, 29]. Senn and Link trained neural networks to map from process output quantities like draw-ins to stresses in the blank during deep drawing. Despite the fact that their work deals with the observation of hidden states instead of input parameters, only one parameter, namely the

blank holder force, was varied during the simulation to generate data for model training and validation, which is not sufficient for our purpose.

The content of this paper is structured as follows. First, the mathematical methods and concepts which are important for the understanding of the work are explained. If the reader is already familiar with these concepts, it can be skipped. In the following chapter, the computational pipeline is presented. The proposed approach is then examined on an AUDI A8 L side panel frame to demonstrate its application on a real world part.

2 Models and methods

This chapter summarizes the most important mathematical concepts which are necessary to fully understand and reproduce the results of this work.

2.1 Principal component analysis (PCA)

Principal component analysis is a widely spread technique in machine learning and mainly used for data exploration of large data sets and formulation of predictive models. PCA was formulated first in 1901 by Pearson [30] and independently developed and formulated by Hotelling [31] in 1933. Given a set of data points in a higher dimensional space, the main idea of PCA is to subsequently approximate the data points by a linear function, where every function is chosen to be perpendicular to the previous ones. This process is repeated n times, where n is equal or less than the dimension of the original data set d . These functions form an orthogonal basis, whose directions are called principal components. By representing the original data in the coordinate system defined by the principal components, the dimension of the original data can be reduced while maintaining as much variance as possible [32, 33].

Let X be a matrix with n rows and d columns. Thus, X is of dimension $n \times d$. In addition, let us assume that the columns of X are standardized with zero mean and unit standard deviation. The principal components of X are then calculated in two steps. First, the sampling covariance matrix C of X is calculated by

$$C = \frac{1}{n-1} X^T X \quad (1)$$

where C is symmetric and contains the variances of the columns in its diagonal as well as the correlations of the columns in its off-diagonal terms. Second, eigendecomposition is performed on C to get d pairs of eigenvectors and eigenvalues

$$C = V^T C_d V \quad (2)$$

where C_d is a diagonal matrix with the eigenvalues of C on its diagonal and V is an orthogonal matrix with the eigenvectors of C in its columns. The columns of the eigenvector matrix V and the eigenvalues in C_d are sorted in decreasing order of the eigenvalues while still maintaining their correct pairings. The interpretation of C_d and V are as follows: While V contains the directions of the principal components of X in its columns, the eigenvalues in C_d represent the distribution of the source data's (X) energy among the direction of the principal components [34] (p. 61).

Using the linear transformation given by Eq. 3, the original data matrix X is mapped to a new space of uncorrelated coordinates defined by the principal component directions. Therefore, the columns of T are the principal components of X .

$$T = XV \tag{3}$$

While computing the linear transformation defined by Eq. 3, not all principal component directions in V have to be kept. By using only the p principal component directions in V corresponding to the p largest eigenvalues and thus maintaining a part of the variance, the reduced matrix T_p is calculated by

$$T_p = XV_p \tag{4}$$

which results in a $n \times p$ matrix instead of $n \times d$ for T . If p is chosen so that $p < d$, then the linear transformation defined by Eq. 4 results in a dimension reduction while maintaining as much variance of the original data in X as possible. The proportion of total variance which is preserved in T_p by using the p principal component directions belonging to the largest eigenvalues λ_i can be calculated by:

$$var_{preserved} = \frac{\sum_{i=1}^p \lambda_i}{\sum_{i=1}^d \lambda_i} \tag{5}$$

Finally, by using Eq. 5 and iteratively adding dimensions to p until a predefined proportion of the preserved variance is reached, the number of needed dimensions in the reduced space is determined.

The application of PCA on an example data set is shown in Fig. 1. First, data points are generated using $x = y$. Second, noise is added to the data. By applying PCA, the principal directions in the data are found which are the original direction of maximum variance (blue) and the direction perpendicular to it containing noise (red). Projecting the data points on the first principal component (blue) and neglecting the second principal component (red), which in fact is a dimensionality reduction from 2D to 1D, preserves 99.7% of the variance of the data in this case.

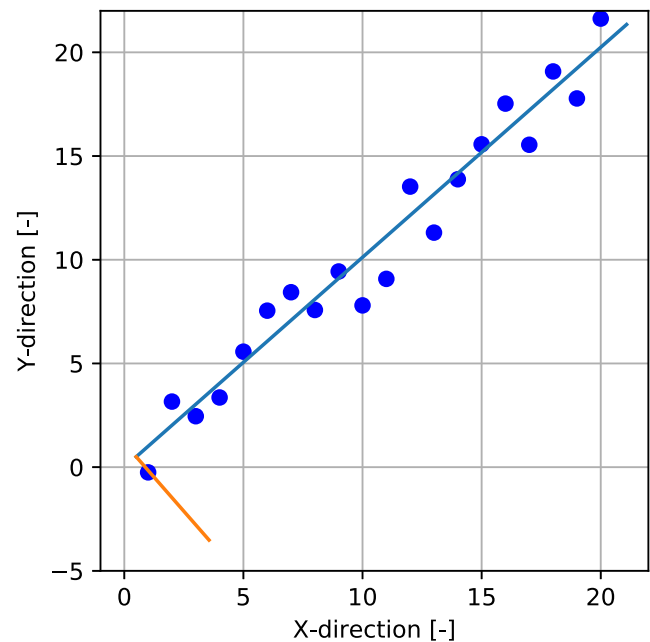


Fig. 1 Illustration of PCA: the directions of the first (blue) and second (red) principal components are revealed by using principal component analysis

2.2 Singular value decomposition (SVD)

Before discussing the singular value decomposition, the notion of singular values has to be introduced. Let X be a matrix of dimension $m \times n$. Taking the product of X with itself, $X^T X$ results in a matrix of dimension $n \times n$. Since $X^T X$ is a square matrix, it has n eigenvalues λ_i . Let's assume that the eigenvalues are sorted in descending order, so that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{n-1} \geq \lambda_n \geq 0$. The singular values σ_i of X are then defined as

$$\sigma_i = \sqrt{\lambda_i} \quad i = 1, 2, \dots, n \tag{6}$$

where λ_i are the eigenvalues of $X^T X$. The singular value decomposition is defined as the factorisation of the $m \times n$ matrix X

$$X = U \Sigma V^T \tag{7}$$

where U and V are orthogonal matrices with dimension $m \times m$ and $n \times n$ respectively and Σ is a $m \times n$ matrix whose diagonal contains the singular values of X [35], [36, p. 70]. The singular value decomposition is defined for every matrix, independent of its shape. One of the most important aspects of singular values lies in the fact that they can be used to determine if a matrix is rank deficient. For a matrix of rank r , it holds that

$$\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0 \tag{8}$$

which shows that the rank of a matrix is equal to the number of nonzero singular values. In addition, singular values do not only allow to determine and quantify the rank of a

matrix, they also provide a tool to determine how far a matrix is from being rank deficient. Theorem 2.5.3 from [36] states that the 2-norm distance of a matrix A to the set of all rank-deficient matrices equals the smallest singular value of the same matrix A (see also [36, p. 70]). Broadbent et al. describe this property as one of the most important properties of singular values [37], and it will be used for the solving of the column subset selection problem as can be seen in the following.

2.3 The column subset selection problem

The column subset selection problem consists of finding the most representative k columns of a matrix [38]. Subset selection algorithms are used for many different applications, among others, ranging from data compression by reducing the number of columns in large matrices to the solution for rank-deficient least squares problems.

Let X be a matrix of dimension $m \times n$ with $m \geq n$. The problem of column subset selection in X can be translated into finding a permutation matrix Π that sorts the columns of X as

$$X\Pi = [X_1 \quad X_2] \quad (9)$$

where X_1 contains the k representative and X_2 the $n - k$ remaining columns. Using matrices X_1 and X_2 , the objective of finding the best representation of X with a minimum number of column vectors can be expressed by the following two conditions [37]:

1. The column vectors in X_1 have to be as linear independent as possible.
2. The residuum of the best possible linear combination of X_1 has to be as close to X_2 as possible, that is, $r = \min_s \|X_1 s - X_2\|_2$ has to be minimised.

Many subset selection algorithms use a QR decomposition to decompose the data matrix X into an orthogonal matrix Q and an upper triangular matrix R since R has the same singular values as X , but numerical computations are easier to perform on R due to its diagonal form [37]. Taking into account that the i th column in $X\Pi$ corresponds to the i th column in R , we can simplify the problem by applying subset selection on R instead of X :

$$X\Pi = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \quad \text{with} \quad R = \begin{bmatrix} R_k & B_k \\ 0 & C_k \end{bmatrix}. \quad (10)$$

In the expression above, a partitioning of R is introduced with the notation in accordance to [37]. Thus, R_k is an upper triangular matrix of dimension $k \times k$, B_k is of dimension $k \times (n - k)$ and C_k is of dimension $(n - k) \times (n - k)$, where

k equals the number of representative columns to select. Inserting Eq. 9 into Eq. 10 leads to

$$X_1 = Q \begin{bmatrix} R_k \\ 0 \\ 0 \end{bmatrix}, \quad X_2 = Q \begin{bmatrix} B_k \\ C_k \\ 0 \end{bmatrix}. \quad (11)$$

Using this notation, the two mentioned objectives for finding independent columns in X can be translated into the following two rather mathematical objectives using the concept of singular values introduced in Section 2.2:

1. The smallest singular value of X_1 and thus of R_k needs to be maximised.
2. The largest singular value of X_2 and thus of C_k needs to be minimised.

Since the deterministic subset selection algorithm by Gu and Eisenstat [39] is used in this work, the algorithm is outlined in the following.

2.3.1 Deterministic subset selection algorithm by Gu and Eisenstat

In 1996, Gu and Eisenstat proposed a deterministic algorithm [39, Algorithm 4] for the column subset selection problem which is shown in Algorithm 1. This algorithm forms the best deterministic approximation of the two conditions mentioned above [37]. In Algorithm 1, matrices are named according to Section 2.3. Additionally, e_i denotes a column vector containing only zeros with the i th entry being equal to 1.

Since R is an upper triangular matrix and its singular values are identical to the ones of X , it holds that

$$|\det X| = |\det R| = |\det R_k| \cdot |\det C_k| \quad (12)$$

while the right side can be rewritten as

$$|\det R_k| \cdot |\det C_k| = \prod_{i=1}^k \sigma_i(R_k) \cdot \prod_{i=1}^{n-k} \sigma_i(C_k). \quad (13)$$

In principle, the algorithm proposed by Gu and Eisenstat looks for two columns i and j in $X\Pi$ that, after permuting them, increase the determinant of R_k . It has to be noted that in order to check if the determinant of R_k is increased by a permutation, the permutation itself does not need to be explicitly computed. Using [39, Lemma 3.1]

$$\left| \frac{\det \tilde{R}_k}{\det R_k} \right| = \sqrt{\left((R_k^{-1} B_k)_{i,j} \right)^2 + \|C_k e_j\|_2^2 \|e_i^T R_k^{-1}\|_2^2} \quad (14)$$

with $\det \tilde{R}_k$ being the determinant of R_k after permuting columns i and j , allows to determine if the permutation of columns i and j leads to an increase of the determinant, thus increasing computational efficiency. Since $\det X$ is invariant

to permutations in its columns, an increase of $|\det \mathbf{R}_k|$ must lead to a decrease of $|\det \mathbf{C}_k|$. Thus, the algorithm leads to

a deterministic solution that satisfies condition 1 and 2 as stated earlier as good as possible [37].

Algorithm 1 Column subset selection algorithm by Gu and Eisenstat [39, Algorithm 4].

Input: matrix \mathbf{X} , number of representative columns k , tolerance parameter f

Output: Permutation matrix $\mathbf{\Pi}$

Compute QR decomposition of matrix \mathbf{X} to initialise permutation matrix $\mathbf{\Pi}$ and \mathbf{R} ;

while there exist i, j so that $\sqrt{(\mathbf{R}_k^{-1} \mathbf{B}_k)_{i,j}^2 + \|\mathbf{C}_k \mathbf{e}_j\|_2^2 \|\mathbf{e}_i^T \mathbf{R}_k^{-1}\|_2^2} > f$ **do**

permute columns i and j of \mathbf{R} ;
 retriangularize \mathbf{R} with QR-decomposition ;
 permute columns i and j of $\mathbf{\Pi}$

end

3 Overview over computational pipeline

Using the models and methods described in Section 2, among others, a computational pipeline is built. It consists mainly of four parts which are shown in Fig. 2. Before applying the whole pipeline, a stochastic finite element simulation of the corresponding forming process is performed. In the first part of the pipeline, the sensor signals are modelled as a function of the simulation parameters. Based on the literature research mentioned in Section 1, draw-ins combined with local blank holder forces are used as sensor signals, since the latter are beneficial to increase the predictability of single parameters especially if the impact of the variation of different parameters on the draw-in is melded in the signal [22]. The function of drawing aids is briefly explained in the following paragraph. In the second step, the global sensitivities of the input on the output of these models are determined using fourier amplitude sensitivity testing (FAST). Third, by applying column subset selection on the global sensitivity matrix, independent sensor positions which maximise information content regarding their sensitivity vectors are determined. In the fourth step, the final surrogate models which model the material, process and tool parameters as a function of the independent sensor signals are calculated. These steps shall be outlined in this chapter in more detail. All programming was done in Python 3.7 using the scikit-learn library (version 0.21.1) [40].

Since draw-in measurements are well known but force measurements in drawing aids are not, the latter shall be briefly explained here. An example of drawing aids used in a cup drawing process is illustrated in Fig. 3. Drawing aids consist of a block that absorbs a part of the blank holder force parallel to the blank, allowing a change of the pressure distribution acting on the blank. The force acting on a drawing aid (denoted by F in Fig. 3) is influenced by many different parameters, including its height, stiffness, blank thickness, uplift force of drawbeads nearby and many

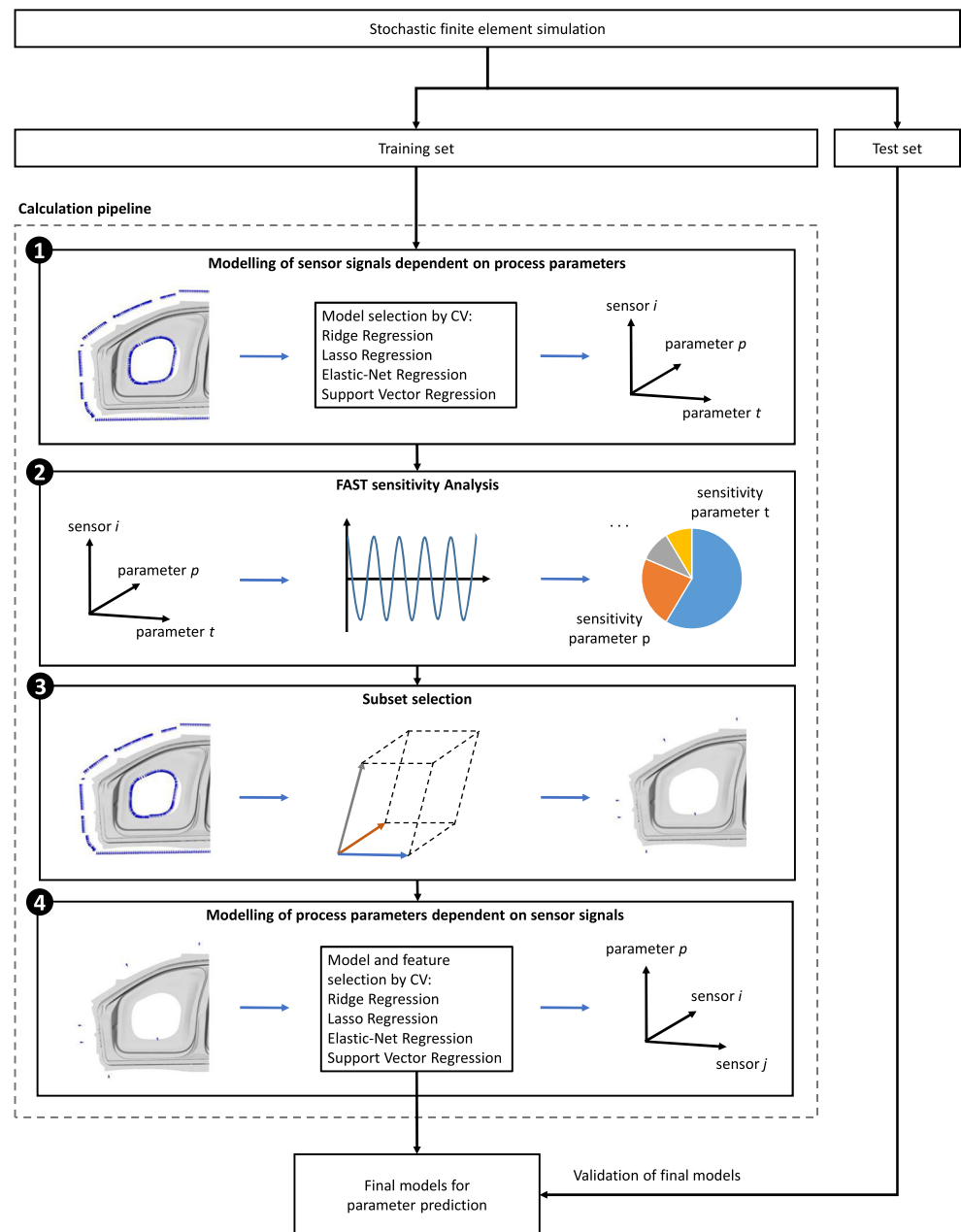
more. Measuring the force acting on different drawing aids in the closed tool allows to characterize the pressure distribution during the process. In the following, the force measurement in the drawing aids will also be referred to as local blank holder force measurement, due to its local effect. In AutoForm, drawing aids are called spacer and their force can be read out from the output file. It has to be noted that the drawing aid height itself is a process parameter, since it can be quickly adjusted from part to part. In this work, the drawing aid height remains constant during the process despite its elastic deformation due to the acting force.

3.1 Generation of training data

To generate training data for the modelling steps, a stochastic finite element simulation is performed using the AutoForm R8 solver. The pure computational approach to model fitting enables the generation of a large number of data points in a short amount of time. Furthermore, the simulative approach enables the variation of parameters which are difficult and costly to vary in reality without any additional cost.

From the set of simulations, three types of data are extracted in the form of matrices, as illustrated in Fig. 4. The design of experiments of the stochastic finite element simulation is stored in the matrix \mathbf{X}_{DoE} . The draw-in perpendicular to the part boundary and the local blank holder force signals are stored in the matrix \mathbf{X}_{drawin} and \mathbf{X}_{force} , respectively. All data matrices are of shape $n \times f$, where n corresponds to the number of simulations and f to the number of features. The number of columns f of \mathbf{X}_{DoE} therefore equals the number of independent parameters, whereas f of \mathbf{X}_{drawin} and \mathbf{X}_{force} equals the number of sensors times the number of timesteps at which the sensor signals are evaluated. The data of these matrices is then splitted into a training (90% of simulations) and a testset (10% of simulations). The training set is used for the

Fig. 2 Computational pipeline to estimate material, tool and process parameters based on sensor signals (blue arrows near part denote draw-in sensors as illustration)



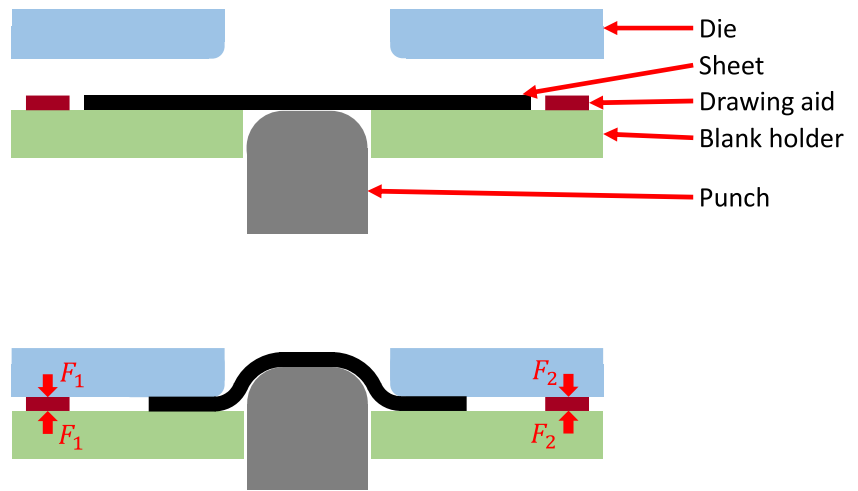
following steps whereas the testset is only used for the final model validation, as can be seen in Fig. 2.

3.2 Modelling of sensor signals

In the first part of the pipeline, all sensor signals are modelled as targets with the simulation parameters as predictor variables. Therefore, the three data matrices X_{DoE} , X_{drawin} and X_{force} are standardized and normalized using the standard deviation and the mean value columnwise. This preprocessing step of the simulation data is crucial since model complexity is regulated by using regularization. After preprocessing, polynomial feature transformation is

applied up to second order which on one hand allows to take into account nonlinear relationships between input and output. On the otherhand, the quadratic model also includes cross-terms which allow the model to express dependencies between different input variables. Afterwards regression for each sensor signal is performed. Instead of using just one model, a pool of different models is used whereas the best performing model as well as its model specific hyperparameters are selected by cross-validation (CV) for each sensor independently. It has to be noted that the degree of feature transformation itself acts as a hyperparameter, thus the quadratic model is only used if it improves the expressive power of the model compared to the linear one.

Fig. 3 Visualization of drawing aids on a cup drawing process with open tools (top) and during the process (bottom); the force acting on the drawing aids are denoted by F_1 and F_2

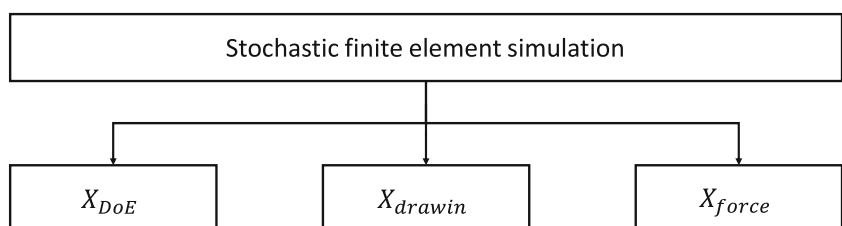


The principle of cross-validation is illustrated in Fig. 5. In a first step, the whole training set is divided into k subsets of approximately equal size. Each model is then fitted k times using $k - 1$ subsets (blue boxes in Fig. 5) for training and the remaining subset (green boxes in Fig. 5) for validation. As metric to quantify the goodness of fit on each validation set the R^2 -score is used. The final model performance is then calculated by averaging the R^2 -scores on each validation set according to Eq. 15. The model as well as the hyperparameters leading to the best R^2 -score are then selected. As number of folds, $k = 8$ is used.

$$Score_{tot} = \frac{1}{k} \sum_{i=1}^k Score_i \tag{15}$$

Senn and Link [28, 29] as well as Breitsprecher et al. [41] used neural networks for surrogate modelling in deep drawing whereas Morand et al. [42] and Huang et al. [43] used kriging models for this task. Our research [44] related to modelling of draw-in as well as drawing aid force signals showed highest modelling accuracy by using linear regression approaches which is in accordance with the results in [23, see Table 4], especially compared to neural networks for regression. The different regression types used in this work are therefore Ridge regression, Lasso regression, Elastic-Net regression and Support Vector regression. The regression approaches are very well documented in [45] as well as [46] in a compact manner.

Fig. 4 Three data matrices are extracted from the simulations



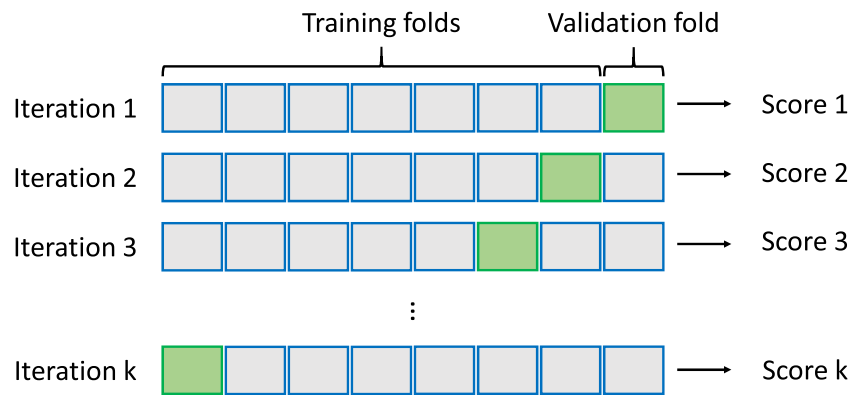
3.3 Sensitivity analysis

In the second part of the pipeline, the global sensitivities of the parameters on all sensors are determined by the Fourier Amplitude Sensitivity Test (FAST) which is documented in [47, p. 159 ff.]. The sensitivities are then written in a matrix S_{drawin} for the draw-ins and S_{force} for the force signals, where each row corresponds to a parameter and each column to a sensor. It has to be noted that the sensitivities of the parameters on the sensor signal for each time step are written into the same column, so that each column in S_{drawin} and S_{force} corresponds to exactly one sensor, which is a necessary condition for the following column subset selection step. Each entry $S_{t,i,j}$ therefore describes the sensitivity of parameter i on sensor j for timestep t . To take into account modelling uncertainty from the modelling step previously, the calculated sensitivities are scaled by a multiplication with the R^2 -score of each sensor model. To prevent sensitivities being scaled up by negative R^2 -scores, the sensitivities for a sensor are set to zero if the R^2 -score of its model is below zero.

3.4 Column subset selection

Before applying the subset selection algorithm on the sensitivity matrices S_{drawin} and S_{force} , the number of representative columns to select has to be determined. Therefore, PCA is applied on the matrices containing the draw-ins X_{drawin} as well as the forces X_{force} to get the

Fig. 5 Principle of model selection with cross-validation



number of principal directions for both sensor types. The preserved variance according to Eq. 5 has to be chosen in a way that it leads to a reasonable trade-off between preserved information content and higher costs due to the increasing number of needed sensors. As a rule of thumb, good results were obtained by using $var_{preserved} = 0.99$. Given the number of needed draw-ins and force sensors, Algorithm 1 is applied on S_{drawin} and S_{force} separately which leads to the most independent columns and therefore the sensor positions that maximise information content regarding their sensitivities. The tolerance parameter f in Algorithm 1 is set to 1, meaning that every increase of $|\det R_k|$ will lead to a permutation of the corresponding columns without any threshold. The columns in X_{drawin} and X_{force} corresponding to the selected sensors are then written into separate matrices which are called $X_{drawin,sel}$ and $X_{force,sel}$.

The interpretation of the working principle of the subset selection algorithm is visualized in Fig. 6. Since the columns in S_{drawin} and S_{force} contain the sensitivities of each independent parameter on each sensor signal, they can also be viewed as vectors in a space whose dimension equals the number of parameters. The subset selection algorithm then tries to find a given number of vectors k , that maximises the volume spanned by these vectors, which tends to select sensitivity vectors that are

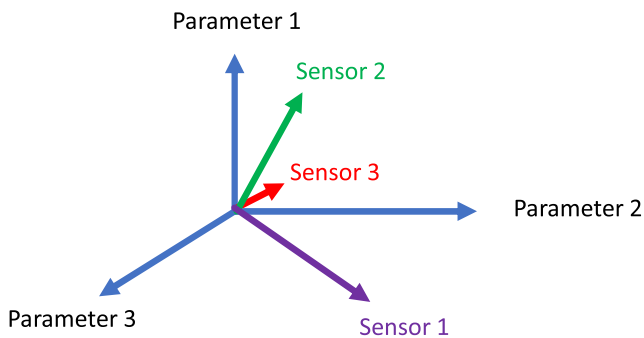


Fig. 6 Visualization of sensitivity vectors of Sensors 1, 2 and 3 expressed in the parameter space

as linearly independent as possible (vectors of Sensor 1 and Sensor 2 in Fig. 6). Since the scaled sensitivities from the previous step are automatically driven to zero due to the multiplication with the R^2 -score if the corresponding sensor is poorly modelled, the subset selection algorithm automatically performs a trade-off between sensors with high sensitivity and high modelling accuracy.

3.5 Modelling of process parameters

In the fourth part of the pipeline, the final models for the parameter estimation are determined. After standardising and normalising $X_{drawin,sel}$ and $X_{force,sel}$ analogously to the first part of the pipeline, regression analysis is performed using the matrices $X_{drawin,sel}$ and $X_{force,sel}$ as predictors and the parameters in X_{DoE} as targets. Again, cross-validation is used to select the best performing model as well as its model specific hyperparameters. The gridsearch is performed for every parameter independently. As evaluation metric, the R^2 -score is used. The number of folds is $k = 8$. The regression methods are again Ridge regression, Lasso regression, Elastic-Net regression and Support Vector regression.

However, there is one main difference compared to the cross-validation procedure used earlier. Here, cross-validation is done additionally over different combinations of predictor variables to perform feature selection. The reason behind this shall be examined in the following in more detail. A simplified example is considered in which the overall friction coefficient of a deep drawing process μ is modelled linearly dependent on the draw-in at only one position, evaluated at $m = 10$ defined blank holder positions d_1, d_2, \dots, d_{10} with d_{10} being the draw-in at the fully drawn part. Let's assume that to fit the model, only $n = 15$ experimental results are available. This model with $m = 10$ features would contain exactly $m + 1 = 11$ parameters, namely $\beta_0, \beta_1, \dots, \beta_{10}$, including the intercept β_0 and would look like

$$\mu(\mathbf{d}) = \beta_0 + \beta_1 d_1 + \beta_2 d_2 + \dots + \beta_{10} d_{10}. \quad (16)$$

Depending on the geometric complexity of the part and of the tools, it is very unlikely that all 10 features are required to predict the friction with high accuracy due to the direct relationship between friction and draw-in. At this point, it is assumed that all other parameters are identical. In fact, if a radial symmetric deep drawn part is considered, the friction coefficient could be estimated very well using only the final draw-in. Therefore, the remaining 9 features are mostly redundant and do not provide much additional information. These additional features still lead to a more complex model since more parameters are required which therefore leads to less certainty about the final model, assuming that the amount of training data remains the same.

In summary, it is beneficial to include the features, in this case the timesteps of the draw-in sensor, which add additional information and to neglect the features which do not contribute to the parameter estimation. As two positive side effects, the model complexity is reduced which increases its interpretability and the reduced number of parameters requires less data points for model fitting, resulting in lower total cost for data generation and modelling.

In this work, the subset of features is determined during cross-validation. Since the computation of every possible combination of sensor times during cross-validation increases exponentially with increasing number of sensors and timesteps, two assumptions are made to reduce the number of combinations:

1. The subset of timesteps of all sensors are identical.
2. Only directly subsequent timesteps are allowed.

Let's consider the example of a tool in which $s = 10$ sensors and $t = 10$ timesteps for each sensor are used. Hence, the total number of features is $m = s \cdot t = 100$. Since every timestep of each sensor can be used as feature or not, the total number of combinations in this case is

$$2^m \approx 1.27 \cdot 10^{30}. \tag{17}$$

It has to be noted that every model-hyperparameter-combination has to be fitted 2^m times during model selection. Furthermore, since cross-validation with $k = 8$ folds requires 8 times model fitting and model validation, the needed computation time increases even more. Testing all combinations is therefore not feasible from a computational point of view. Assumption 1 reduces the number of combinations to

$$2^t = 1024 \tag{18}$$

where all sensors use the same timesteps. By additionally including assumption 2 which requires that only subsets of

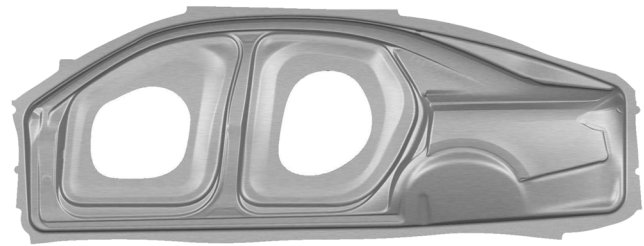


Fig. 7 Left side panel frame from the AUDI A8 L

sequential timesteps are allowed, the number of possible combinations decreases even more to

$$t + (t-1) + (t-2) + \dots + (t-(t-1)) = t^2 - \sum_{i=1}^{t-1} i = 55. \tag{19}$$

4 Case-study with results

In this section, the methodology presented above is examined on the left side panel frame of an AUDI A8 L shown in Fig. 7. The application and its interim results are documented in the same order as in Section 3.

4.1 Data generation with FEM

To investigate the behaviour of the draw-in and the local blank holder forces as a function of the material, tool and process parameters, a stochastic finite element simulation is performed. The setup of the simulation with its tools is shown in Fig. 8. The total drawing depth of the part equals 240 mm.

4.1.1 Material description

The side panel frame is drawn from an aluminum AA6016 sheet with 1.15-mm thickness, whose material properties are based on previous work by Neuhauser et al. [21, 22]. Nevertheless, the basic material parameters are outlined in the following. For the description of the flow behaviour, a combined approach using the flow curve approximation according to Swift [48] and Hockett-Sherby [49] is used according to Eq. 20.

$$\sigma_{comb} = \alpha \sigma_{HS} + (1 - \alpha) \sigma_{Swift} \tag{20}$$

with

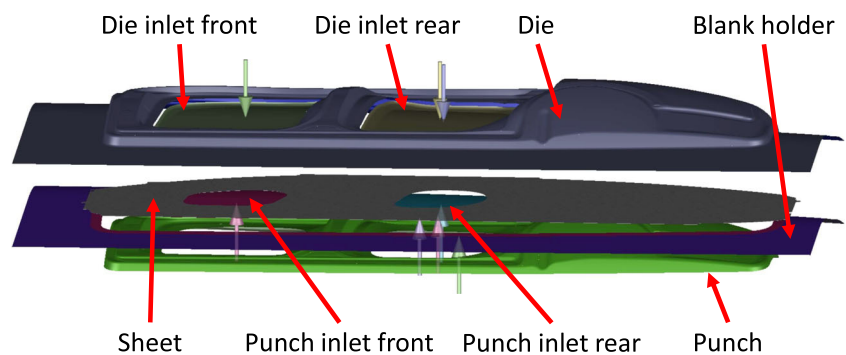
$$\sigma_{Swift} = C (\varphi_{pl} + \varphi_0)^m \tag{21}$$

and

$$\sigma_{HS} = \sigma_{sat} - (\sigma_{sat} - \sigma_i) \exp(-a\varphi_{pl}^p) \tag{22}$$

where φ_{pl} denotes the equivalent plastic strain. The flow curve parameters are defined in Table 1. For the description of the yield locus, the formulation according to Barlat and

Fig. 8 Tool setup for the AUDI A8 L in AutoForm



Lian [50] is used. The final yield locus is calculated based on the anisotropy values R_0 , R_{45} , R_{90} and the M-value. The used parameters of the yield locus are summarized in Table 1.

4.1.2 Variation of independent parameters

As independent variables, relevant parameters for tool tryout as well as material parameters are selected. The main process parameters of this tool are the blank holder force and the height of all seven drawing aids, resulting in eight independent parameters which can easily be modified. Additionally, drawbeads are adjusted during tryout to modify the local restraining of the blank. Therefore, the drawbeads are segmented at the main areas of the part, as can be seen in Fig. 9. db_{10} and db_{11} in Fig. 9 are not varied since if two drawbeads are present at a given location, only one of them is needed in order to modify the local restraining force on the blank. Thus, additional nine independent parameters, namely the height of the drawbeads db_1 up to db_9 are considered to be independent parameters. To take into account noise acting on the process, the global friction coefficient, variation of the blank thickness and material parameters are taken into account too. As material parameters, the yield stress and tensile strength are varied correlated (one independent parameter), and the R-values R_0 , R_{45} and R_{90} are also varied correlated (one independent parameter). Thus, the noise variables result in additional four independent parameters. Although the estimation of material parameters and blank thickness is not of main interest during tool tryout, their consideration is still necessary since the exact values of these noise variables are unknown. Thus, it is not only of interest to see if the relevant parameters for tryout are predictable, but if they

are predictable under uncertainty about material parameters, blank thickness and friction coefficient too.

Summing up the eight main process parameters, nine drawbead heights and four noise variables leads to 21 independent parameters in total. These parameters are varied during the stochastic finite element simulation and will later be predicted using surrogate models. All 21 parameters as well as their variation range during the simulation are summarized in Table 2.

The design of experiments is created using latin hypercube sampling [51]. Using twice as much data points as model coefficients for a quadratic model with a 21-dimensional input and adding validation (14.3%) and testset (10%), the number of needed data points equals 635. Therefore, the number of simulations was set to 635. All sensor signals were extracted in 9 increments, namely at 239, 100, 50, 30, 20, 10, 5, 2 and 1 mm above the bottom dead center of the blank holder movement.

4.2 Modelling of sensor signals dependent on process parameters

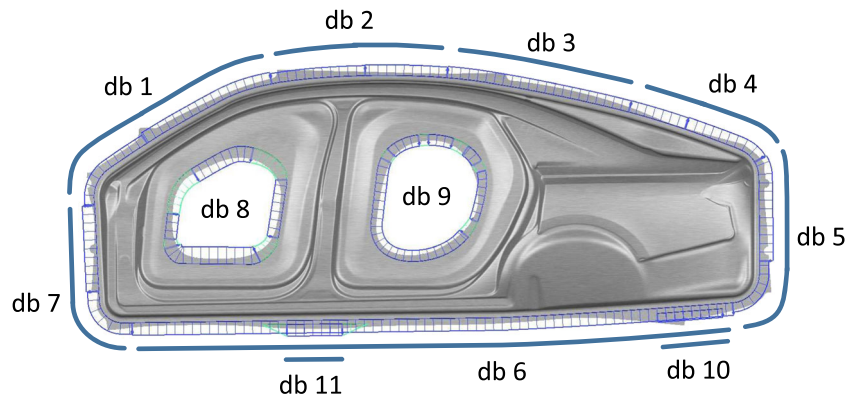
In Fig. 10, all possible sensor positions for the draw-in (cones) and the force (cubes) sensors are shown. The gaps between the sensor positions are mainly caused by the tool geometry which contains ribs and other geometrical features that restrict the possible positioning of the sensors in reality. In total, 772 possible draw-in sensors and 7 possible drawing aids with force sensors are placed.

The parameter grid used during cross-validation for sensor modelling is listed in Table 3. During cross-validation over all sensors, a mean R^2 -score of 0.971 and a median of 0.983 with standard deviation 0.035 is achieved. Out of the 779 sensors, 776 (including all force sensors)

Table 1 Parameter values used for the description of the flow curve and the yield locus

C [MPa]	σ_i [MPa]	σ_{sat} [MPa]	φ_0 [-]	m [-]	α [-]	p [-]	R_0 [-]	R_{45} [-]	R_{90} [-]	M-value [-]
481.2	120	310	0.01	0.312	9.15	0.94	0.72	0.43	0.62	8

Fig. 9 Segmentation of the drawbeads around the part



are modelled with a R^2 -score of 0.8 or higher. A histogram visualizing the distribution of the R^2 -scores is given with Fig. 11. The regions where the coefficient of determination reaches values below 0.9 belong mainly to areas where complex forming conditions are present, namely the area around the rear wheel, the window in the back as well as the area around the mirror.

4.3 Determination of sensor positions

Applying PCA on X_{drawin} and X_{force} separately which contain the sensor values of each simulation in its rows yields 19 draw-in and 7 force sensors. The proportion of preserved variance is set to 0.99. After calculating the number of needed sensors, their position is determined using Algorithm 1 with $k = 19$ for S_{drawin} and $k = 7$ for S_{force} . The progress of the subset selection algorithm can be controlled by visualizing the evolution of the absolute value of the determinant of R_k and C_k from Eq. 12. Since the algorithm progressively moves the most linearly independent columns of the sensitivity matrices to the left and the linearly dependent columns to the right, $|\det R_k|$ monotonically increases whereas $|\det C_k|$ monotonically decreases with increasing number of iterations. It has to be noted that the opposite is not possible for a properly working

algorithm since if the column considered would decrease the linear independence of the columns on the left, then the condition

$$\left| \frac{\det \tilde{R}_k}{\det R_k} \right| > f = 1 \tag{23}$$

would not be fulfilled. Therefore, the columns would not be swapped and the determinants would remain constant during this iteration. The absolute values of the determinant of R_k are visualized for the draw-ins in Fig. 12. Since the tool setup contains 7 force sensors which are all selected, the subset selection algorithm performs no permutations. Since every possible permutation of columns is tested, the selected columns represent the most linearly independent ones possible.

The finally calculated positions corresponding to the selected columns based on the subset selection algorithm are visualized in Fig. 13.

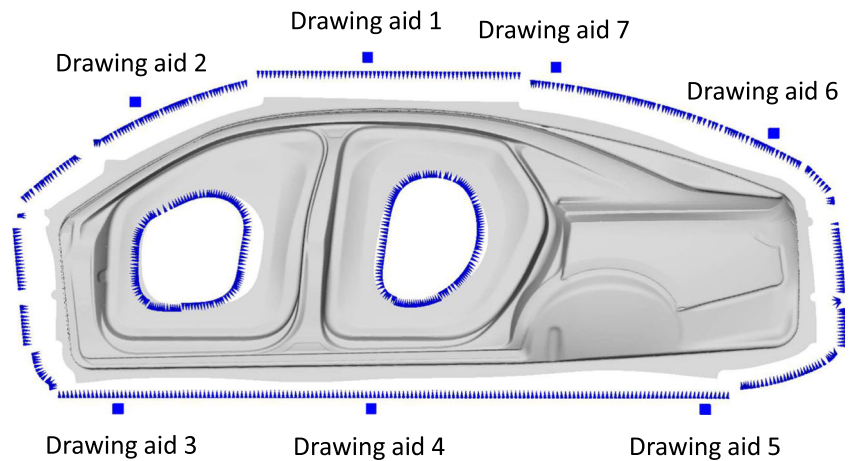
4.4 Modelling of parameters dependent on sensor signals

After determination of the sensor positions, the modelling of the material, process and tool parameters is performed. For model selection during cross-validation, the parameter grid

Table 2 Independent parameters used in the stochastic finite element simulation. The setup results in 21 independent parameters in total

Parameter	Minimum	Maximum	Correlated with
Blank holder force	1863.9 kN	3265.5 kN	-
Drawing-aid heights (7 independent)	1.15 mm	1.45 mm	-
Variation in drawbead heights (9 independent)	-1.25 mm	+1.25 mm	-
Blank thickness	1.14 mm	1.16 mm	-
Friction coefficient	0.07	0.11	-
Yield stress σ_0	107.2 MPa	137.2 MPa	-
Tensile strength R_m	207.9 MPa	237.9 MPa	σ_0
Anisotropy R_0	0.6	0.7	-
Anisotropy R_{45}	0.4	0.55	R_0
Anisotropy R_{90}	0.6	0.75	R_0

Fig. 10 All possible sensor positions: draw-in sensors (cones) and blank holder force sensors (cubes)



given in Table 4 is used. The grid could also be extended with more nonlinear models and even neural networks, but to show the basic principle of the workflow, only relatively simple regression approaches are used. For the validation of the final estimators, the testset depicted in Fig. 2 is used. The R^2 -scores achieved on the testset are shown in Fig. 14.

5 Interpretation and discussion of results

A more detailed inspection of the models selected by the gridsearch during sensor modelling in the first step in Fig. 2 reveals that, without any exception for all 779 sensors, quadratic models perform best for all draw-in sensors and linear models perform best for all force sensors. The former result makes intuitively sense due to two reasons: First, in contrast to the linear model, the quadratic model also contains terms to express cross-correlations between different input parameters. Cross-correlations exist for example between the blank holder force and the friction coefficient, since the sensitivity of the draw-in with respect to the blank holder force increases with higher friction

coefficient and vice versa. Second, the dependency between input parameters and draw-in is not necessarily linear, since for example the effect of large drawbead heights saturates at a certain level where material flow is almost prevented. The selection of only linear models for the force sensors is caused by the fact that the force acting on the drawing aids depend linearly on their deformation length which is given by the drawing aid height, since the drawing aids are deformed in the linear elastic range. Using a linear model, this linear relationship can almost perfectly be approximated. Therefore, by selecting only linear models, the workflow chooses the correct model type. The average coefficient of determination of 0.971 underlines the remarkably high accuracy of modelling achieved using polynomial regression.

Due to tremendous redundancy in the draw-in sensors, PCA determines the number of principal directions of all 772 draw-in sensors equal to 19 while preserving 99% of the variance which leads to a significant dimensionality reduction. On the other hand, since the force signals contain mainly information about the drawing aid heights but also about the blank holder force and blank thickness which turns out to be decorrelated for each force sensor, all 7 force sensors are selected. As can be seen in the sensor

Table 3 Grid for sensor modelling during cross-validation

Approach	Hyperparameters	
Ridge	λ	[0.0001, 0.001, 0.01, 0.1, 1]
	feat. degree	[1, 2]
Lasso	λ	[0.0001, 0.001, 0.01, 0.1, 1]
	feat. degree	[1, 2]
Elastic-net	λ	[0.0001, 0.001, 0.01, 0.1, 1]
	feat. degree	[1, 2]
	L1-ratio	[0.25, 0.5, 0.75]
SVR	C	[1e-4, 1e-3, 1e-2, 1e-1, 1, 2]
	ϵ	[1e-4, 1e-3, 1e-2, 1e-1, 1]
	kernel type	[linear]
	feat. degree	[1, 2]

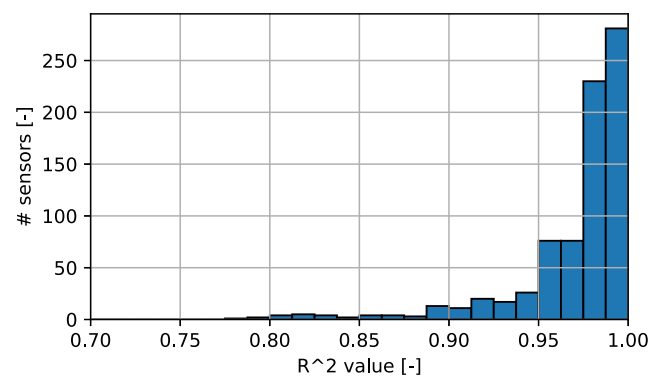


Fig. 11 Distribution of R^2 -scores for all 779 sensors

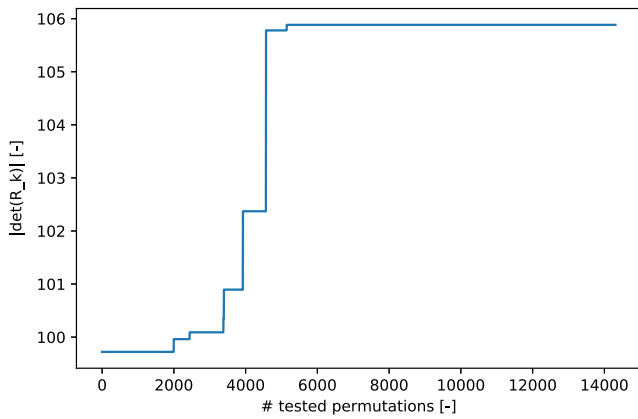


Fig. 12 $|\det(\mathbf{R}_k)|$ over iterations during subset selection for the draw-in signal; larger value means more sensitive sensor signal in summary

positioning shown in Fig. 13, the sensor positions are chosen to be distributed over the whole part. Drawbead *db2* is the only drawbead which is not observed by a draw-in sensor nearby, but according to Fig. 14 the height of *db2* is still estimated with high accuracy. The reason behind this lies in the fact that increasing restraining force of *db2* not only reduces the draw-in nearby, but also increases the draw-in for example on the inlets. These global effects of the restraining force are easily captured by the sensitivity analysis, resulting in a minimum number of sensors. Furthermore, the algorithm puts heavy weight regarding sensor positioning on the rear wheel, which is also the area with the largest absolute draw-in. This area reacts sensitive especially on changes of the blank holder force or friction coefficient. In general, Fig. 13 shows that the algorithm tends to place sensors near the attributes to be observed. Although the big picture remains the same, it has to be noted that the exact sensor positioning also depends on the split of training and testset due to high correlation in the sensor signals in combination with the stochasticity of the data. The selected sensor layout therefore is not necessarily unique.

Fig. 13 Selected sensor positions by the subset selection algorithm (two draw-in sensors are placed at the edge of the front mirror)

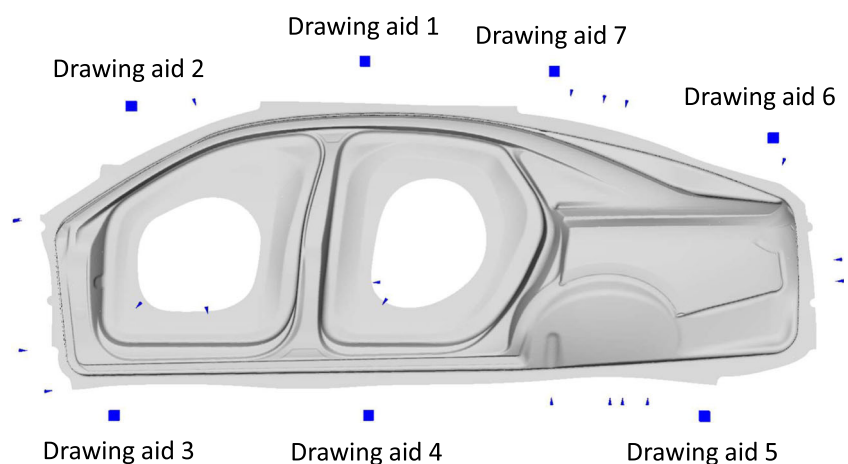


Table 4 Grid for parameter modelling during cross-validation

Approach	Hyperparameters	
Ridge	λ	[0.0001, 0.001, 0.01, 0.1, 1]
Lasso	λ	[0.0001, 0.001, 0.01, 0.1, 1]
Elastic-net	λ	[0.0001, 0.001, 0.01, 0.1, 1]
	L1-ratio	[0.25, 0.5, 0.75]
SVR	C	[1e-4, 1e-3, 1e-2, 1e-1, 1, 2]
	ϵ	[1e-4, 1e-3, 1e-2, 1e-1, 1]
	kernel type	[linear, polynomial]
	kernel degree	[1, 2]

The result depicted in Fig. 14 shows that 20 out of 21 parameters reach a coefficient of determination of 0.9 or higher which is denoted by the red line. The average R^2 -score achieved equals 0.934. The parameters predicted with the highest accuracy are the seven drawing aid heights. This behaviour can be explained by the physical behaviour of the drawing aid, which is deformed in a linear elastic way during the process. Due to the linear relationship between the drawing aid deformation given by its drawing aid height and the acting force, the initial drawing aid height can be easily predicted by the model by the acting force on each drawing aid as well as its stiffness, whereas the latter equals the slope of the linear curve and is thus fitted during the regression analysis. The overall result shows that not only is the model capable of predicting process parameters, it even allows to estimate flow curve parameters as well as R -values dependent on draw-ins and local blank holder forces which allows the estimation of fluctuations in material parameters. For a parameter being predictable by using the proposed methodology, the following two conditions need to be fulfilled:

- The influence of a process parameter has to be unambiguously identifiable in the sensor data. If the variation of a parameter has no influence on the

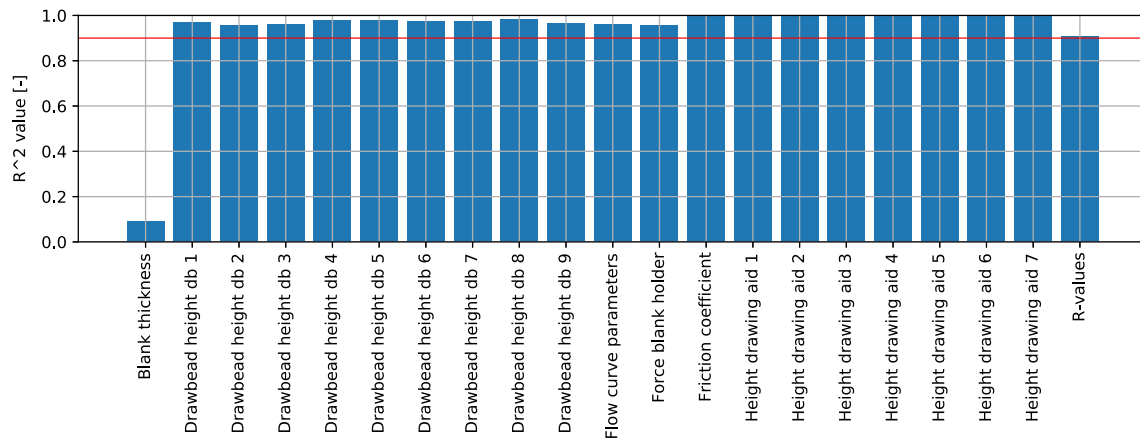


Fig. 14 R^2 -scores achieved at estimating process and material parameters on the testset; red line denotes R^2 of 0.9 which is used as baseline

measured sensor values or if the influence of two independent parameters is melded together, the inverse problem can not be solved. This issue becomes clear by looking as an example at a linear mapping with 4-dimensional input and output, which can be described by a multiplication of the input with a transformation matrix containing the model weights. The solving of the inverse problem would correspond to inverting the transformation matrix, which turns out to be singular and thus non-invertible if one row contains only zeros or is linearly dependent on another row. The solution of the inverse problem is in this case either not defined or not unique. This issue implies that a perfectly accurate surrogate model of the forward problem (step 1 in Fig. 2) not necessarily leads to an accurate inverse model (step 4 in Fig. 2).

- The absolute sensitivity of each parameter on the sensor signals has to be of comparable order compared to the other parameters. If the sensitivities of the parameters on the sensor signal are of very different orders of magnitude, the modelling error on the sensor signal of sensitive parameters can exceed the total influence of a less sensitive parameter on the sensor signals. Therefore, the prediction of the latter is not possible from a statistical point of view.

The prediction of the blank thickness is the only parameter that does not pass the benchmark with an R^2 -score of 0.093. Considering the definition of the coefficient of determination

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (24)$$

where \hat{y}_i denotes the model prediction for data point i , y_i the underlying truth for data point i , \bar{y} the mean of all y_i for all data points i and n denotes the number of data points, the

result means that the mean squared error in the prediction of the blank thickness is as large as approximately 90.7% of the variance of the blank thickness and therefore the prediction does not contain much relevant information about the underlying truth at all.

The reason for this result is analyzed using the two requirements defined above for the inverse parameter identifiability. According to the first criteria, either the blank thickness has no influence on the draw-in and the force sensors (option one) or the influence is melded together in the sensor signal with another parameter (option two). Option two is discarded since in this case there would be another parameter which would also show a low prediction accuracy. Thus, option one is considered in more detail. Therefore, the whole methodology is applied on a separate stochastic finite element simulation using the same part in which the blank thickness is the only independent parameter. The result shows that the blank thickness is indeed identifiable using two draw-in sensors with an R^2 -score of 0.942. The first criteria is thus fulfilled.

Analysis of the second criteria leads to different orders of the sensitivities of the blank thickness compared to the other parameters on the sensors. The variation of the blank thickness according to Table 2 leads to a variation range of the draw-in of approximately 0.1–0.3%, whereas the variation of the drawbeads leads to a variation range of the draw-in of approximately 5–15%, depending on the location where the draw-in is measured. The sensitivity of the drawbeads in the given range on the draw-in is therefore 50 times as large as the sensitivity of the blank thickness and of different order of magnitude. Thus, the second criteria is not fulfilled. It is concluded that in general blank thickness is predictable too using draw-ins and local blank holder forces, but its lower sensitivity on the sensors prohibits an accurate prediction in the given range while using more sensitive parameters at the same time.

6 Conclusion

In this work, a new approach for solving the inverse problem and thus for the observation of input parameters in deep drawing based on its output, is presented. The approach is based on simulation data, which is first used to determine information maximising sensor positions regarding their sensitivities on the input parameters of the process. The simulation data is then used to train predictive models which express the process and material parameters as a function of the sensor data. For model training, only virtually generated data is used. The models are then evaluated on a separate testset.

The result shows that 20 process parameters out of 21 are predicted with sufficient accuracy characterized by a R^2 -score of 0.9 or higher. The prediction of the blank thickness fails due to the fact that its variation in the given range has almost no influence on the draw-in as well as local blank holder force sensors. The following conclusions are made:

- For the case-study presented, 19 draw-in and 7 blank holder force sensors were sufficient for the task. The application of sensitivity analysis in combination with principal component analysis and subset selection enabled a tremendous reduction in dimensionality of the data from 779 to 26 sensors while preserving 99% of the variance in the data. The methodology proposed allows the identification of suitable sensor positions for draw-in as well as blank holder force measurements.
- The proposed regression models are capable of predicting the mentioned process as well as material parameters as long as the two criteria of unique identifiability and comparable scale of sensitivities mentioned above are fulfilled.
- The prediction accuracy reached for 20 out of 21 parameters is shown to be sufficient to apply the model in tryout operations with negligible computational (<1 s for prediction) cost in comparison to an inverse parameter identification based on stochastic optimization of a finite element model. It is worth mentioning that the stochastic finite element simulation used in this work has to be carried out anyway during tool engineering for analysing process robustness.
- Precise identification of material and process parameters is limited by a lack of sensitivity of the parameters on the sensor signals. Low sensitivities can limit the inverse prediction even if a precise surrogate model of the deep drawing process is found, as was observed for the blank thickness in this work. In our case, the forward model of the process reached an average R^2 -score of 0.971 with a minimum score of 0.781 for the worst sensor model, whereas the inverse model reached an average R^2 -score of 0.934 with a minimum score of 0.093 for the worst parameter.

The stochastic finite element simulation used in this work consists of 635 simulations. The simulation time in AutoForm was $\approx 3h$ per simulation and core with an Intel(R) Xeon(R) E5-1650 CPU and using all 6 cores in parallel, resulting in a total calculation time of around 14 days and 1.67 TB of data. The calculation time for all models proposed in this work was $\approx 7h$ using again all 6 cores in parallel.

Acknowledgements The authors are grateful to AUDI AG for the support and for making this project possible. We would also like to thank AutoForm Engineering GmbH for providing the licenses and using their FEM software AutoForm.

Author contribution Matthias Ryser: conceptualization, methodology, investigation, writing, formal analysis, visualization. Felix M. Neuhauser: conceptualization, methodology. Christoph Hein: Conceptualization, review and editing. Pavel Hora: supervision, review, formal analysis. Markus Bambach: review.

Funding Open Access funding provided by ETH Zurich. This project was supported by Audi AG.

Data availability Due to the ongoing project, no data can be provided at present.

Declarations

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication All authors consent to the publication of the manuscript.

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Balla VK, Deckers E, Jonckheere S, Pluymers B, Desmet W (2021) Incommensurate vibro-acoustic performance due to in-process blank holder force variation during deep drawing process. *Appl Acoust* 172:107618. <https://doi.org/10.1016/j.apacoust.2020.107618>
2. Elend LE (2001) Einsatz elastischer Niederhaltersysteme zur Erweiterung der Prozeßgrenzen beim Tiefziehen. PhD thesis, University of Hanover

3. Schulte S (1999) Umformsimulation als fester Bestandteil des Produktentstehungsprozesses bei Audi. In: Umformtechnischen Kolloquium, vol 25, Hannover, pp 263–277
4. Birkert A, Haage S, Straub M (2013) Umformtechnische Herstellung komplexer Karosserieteile: Auslegung von Ziehanlagen. Springer. <https://doi.org/10.1007/978-3-642-34670-5>
5. Essig P, Liewald M, Bolay C (2020) Contact area evaluation of digitalized spotting images as a criterion for die tryout. *Procedia Manuf* 47:855–860
6. Braedel M (2009) Optimierung des Werkzeugeinarbeitungsprozesses für Karosserieaußenhautteile. PhD thesis, Technical University of Munich
7. Wang C, Zhang JJ, Goan N (2005) Draw-in map—a road map for simulation-guided die tryout and stamping process control. In: AIP Conference proceedings, american institute of physics, vol 778, pp 66–69. <https://doi.org/10.1063/1.2011195>
8. Prexl AAK (2013) Entwicklung einer Methodik zur simulationsgestützten Inbetriebnahme von Karosseriewerkzeugen. PhD thesis, Technical University of Munich
9. Harsch D, Heingärtner J, Hortig D, Hora P (2016) Virtual tryout planning in automotive industry based on simulation metamodels. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, vol 159, pp 012007
10. Harsch D, Fischer P, Berisha B, Heingärtner J, Renkci Y, Hora P (2018) Considering fluctuations of material properties, stainless steel 1.4301, on manufacturability of kitchen sinks. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, vol 418, pp 012113
11. Harsch D, Heingärtner J, Hortig D, Hora P (2017) Observability of quality features of sheet metal parts based on metamodels. *Comput Plast XIV Fund Appl*:692–703
12. Harsch D, Heingärtner J, Hortig D, Hora P (2016) Process windows for sheet metal parts based on metamodels. In: *Journal of Physics: Conference Series*. IOP Publishing, vol 734, pp 032014
13. Harsch D, Heingärtner J, Renkci Y, Hora P (2017) Influence of scattering material properties on the robustness of deep drawing processes. In: 10th Forming Technology Forum. Model Based Control for Smart Forming Processes
14. Harsch D, Heingärtner J, Renkci Y, Hora P (2018) Metamodel-based methods to verify the feasibility of a process control in deep drawing. In: *Journal of Physics: Conference Series*. IOP Publishing, vol 1063, pp 012185. <https://doi.org/10.1088/1742-6596/1063/1/012185>
15. Harsch D (2019) Digitale Zwillingmodelle zur optimalen Auslegung und Regelung von bedingt robusten Blechumformprozessen. PhD thesis, ETH Zurich
16. AutoForm-Tryout Assistant. <https://www.autoform.com/de/produkte/autoform-forming/autoform-tryoutassistant/>, Accessed: 15 April 2020
17. Stippak M, Carleer B (2019) Bridging the gap between engineering and tryout of sheet metal forming parts by a smart solution. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, vol 651, pp 012011
18. Fischer P, Harsch D, Heingärtner J, Renkci Y, Hora P (2016) Inline feedback control for deep drawing applications. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, vol 159, pp 012006
19. Fischer P, Harsch D, Heingärtner J, Renkci Y, Hora P (2017). In: Approaches for control in deep drawing. In: Proceedings of the Forming Technology Forum. Enschede, The Netherlands, pp 43–48
20. Fischer P, Harsch D, Heingärtner J, Renkci Y, Hora P (2018) Implementation of feedback control in kitchen sink production. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, vol 418, pp 012110. <https://doi.org/10.1088/1757-899X/418/1/012110>
21. Neuhauser F, Struck R, Mautz H, Manopulo N, Hora P (2019) Sensor placement design strategy and quality estimation in modern car body production using stochastic finite element methods. *Procedia Manuf* 27:104–111. <https://doi.org/10.1016/j.promfg.2018.12.051>
22. Neuhauser FM (2020) Potentials of artificial intelligence in production. PhD thesis, ETH Zurich
23. Röhner S, Breitsprecher T, Wartzack S (2011) Acquisition of design-relevant knowledge within the development of sheet-bulk metal forming. In: Proceedings of the 18th International Conference on Engineering Design, vol 6, pp 108–120
24. Lim KYH, Zheng P, Chen CH (2020) A state-of-the-art survey of Digital Twin: techniques, engineering product lifecycle management and business innovation perspectives. *J Intell Manuf* 177:1313–1337. <https://doi.org/10.1007/s10845-019-01512-w>
25. Chamekh A, Belhadjsalah H, Hambli R, Gahbiche A (2006) Inverse identification using the bulge test and artificial neural networks. *J Mater Process Technol* 177(1-3):307–310
26. Pereira A, Prates P, Oliveira M, Fernandes J (2020) Inverse identification of the work hardening law from circular and elliptical bulge tests. *J Mater Process Technol* 279:116573. <https://doi.org/10.1016/j.jmatprotec.2019.116573>
27. Hippke H, Berisha B, Hora P (2020) A full-field optimization approach for iterative definition of yielding for non-quadratic and free shape yield models in plane strain. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, vol 967, pp 012084
28. Senn M, Link N (2010) Hidden state observation for adaptive process controls. In: Proceedings of the Second International Conference on Adaptive and Self-adaptive Systems and Applications. Citeseer, pp 52–57
29. Senn M, Link N (2012) A universal model for hidden state observation in adaptive process controls. *Int J Adv Intell Syst* 4(3-4):245–255
30. Pearson K (1901) Liii. On lines and planes of closest fit to systems of points in space. *The London. Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2(11):559–572. <https://doi.org/10.1080/14786440109462720>
31. Hotelling H (1933) Analysis of a complex of statistical variables into principal components. *J Educ Psychol* 24(6):417–441. <https://doi.org/10.1037/h0071325>
32. Jolliffe I (2002) Principal component analysis. Springer, New York
33. Shlens J (2014) A tutorial on principal component analysis. arXiv:14041100
34. Myers R (2019) Data management and statistical analysis techniques. Springer Dordrecht
35. Stewart GW (1993) On the early history of the singular value decomposition. *SIAM Rev* 35(4):551–566
36. Golub GH, Van Loan CF (2012) Matrix computations, vol 3. JHU press, Baltimore
37. Broadbent ME, Brown M, Penner K, Ipsen I, Rehman R (2010) Subset selection algorithms: Randomized vs. deterministic. *SIAM Undergrad Res Online*:50–71
38. Boutsidis C, Mahoney MW, Drineas P (2009) An improved approximation algorithm for the column subset selection problem. In: Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms. SIAM, pp 968–977
39. Gu M, Eisenstat SC (1996) Efficient algorithms for computing a strong rank-revealing qr factorization. *SIAM J Sci Comput* 17(4):848–869
40. Scikit-learn library for machine learning. <https://scikit-learn.org/0.21/>, Accessed: 20 May 2020
41. Ashhab MS, Breitsprecher T, Wartzack S (2014) Neural network based modeling and optimization of deep drawing–extrusion combined process. *J Intell Manuf* 25(1):77–84

42. Morand L, Helm D, Iza-Teran R, Garcke J (2019) A knowledge-based surrogate modeling approach for cup drawing with limited data. In: IOP Conference Series: Materials Science and Engineering. IOP Publishing, vol 651, pp 012047
43. Huang C, Radi B, El Hami A (2016) Uncertainty analysis of deep drawing using surrogate model based probabilistic method. *Int J Adv Manuf Technol* 86(9):3229–3240. <https://doi.org/10.1007/s00170-016-8436-4>
44. Ryser M, Bambach M (2021) Comparison of linear regression and neural networks as surrogates for sensor modeling on a deep drawn part. The 13th International Conference on the Technology of Plasticity (accepted)
45. Bishop CM (2006) Pattern recognition and machine learning. Springer, New York
46. Murphy KP (2012) Machine learning: a probabilistic perspective. MIT press, Cambridge
47. Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Saisana M, Tarantola S (2008) Global sensitivity analysis: the primer. Wiley
48. Swift H (1952) Plastic instability under plane stress. *J Mech Phys Solids* 1(1):1–18. [https://doi.org/10.1016/0022-5096\(52\)90002-1](https://doi.org/10.1016/0022-5096(52)90002-1)
49. Hockett J, Sherby O (1975) Large strain deformation of polycrystalline metals at low homologous temperatures. *J Mech Phys Solids* 23(2):87–98. [https://doi.org/10.1016/0022-5096\(75\)90018-6](https://doi.org/10.1016/0022-5096(75)90018-6)
50. Barlat F, Lian K (1989) Plastic behavior and stretchability of sheet metals. Part I: a yield function for orthotropic sheets under plane stress conditions. *Int J Plast* 5(1):51–66. [https://doi.org/10.1016/0749-6419\(89\)90019-3](https://doi.org/10.1016/0749-6419(89)90019-3)
51. McKay MD, Beckman RJ, Conover WJ (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Techn Amer Stat Assoc* 21(2):239–245. <https://doi.org/10.2307/1268522>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.