Journal of Manufacturing Systems 67 (2023) 281–295

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Technical paper

A methodology for data-driven adjustment of variation propagation models in multistage manufacturing processes

Rubén Moliner-Heredia, Ignacio Peñarrocha-Alós, José Vicente Abellán-Nebot *

Department of Industrial Engineering Systems and Design, Universitat Jaume I, Av. Sos Baynat s/n., Spain

ARTICLE INFO

Keywords:
Zero defect manufacturing
Multistage manufacturing processes
Model adjustment
Model calibration

ABSTRACT

In the current paradigm of Zero Defect Manufacturing, it is essential to obtain mathematical models that express the propagation of manufacturing deviations along Multistage Manufacturing Processes (MMPs). Linear physical-based models such as the Stream of Variation (SoV) model are commonly used, but its accuracy may be limited when applied to MMPs with a large amount of stages, mainly because of the modeling errors at each stage that are accumulated downstream.

In this paper we propose a methodology to calibrate the SoV model using data from the inspection stations and prior engineering-based knowledge. The data used for calibration does not contain information about the sources of variation, and they must be estimated as part of the model adjustment procedure. The proposed methodology consists of a recursive algorithm that minimizes the difference between the sample covariance of the measured Key Product Characteristic (KPC) deviations and its estimation, which is a function of a variation propagation matrix and the covariance of the deviation of the variation sources. To solve the problem with standard convex optimization tools, Schur complements and Taylor series linearizations are applied. The output of the algorithm is an adjusted model, which consists of a variation propagation matrix and an estimation of the aforementioned variation source covariance.

In order to validate the performance of the algorithm, a simulated case study is analyzed. The results, based on Monte Carlo simulations, show that the estimation errors of the KPC deviation covariances are proportional to the measurement noise variance and inversely proportional to the number of processed parts that have been used to train the algorithm, similarly to other process estimators in the literature.

1. Introduction

Zero Defect Manufacturing (ZDM) paradigm has gained traction on the quality management agenda since 2010, mainly moved by the trend of digitalization and the Industry 4.0 [1]. In a position paper, Psaromatis et al. [2] defined the ZDM paradigm as "a holistic approach for ensuring both process and product quality by reducing defects through corrective, preventive, and predictive techniques, using mainly data-driven technologies and guaranteeing that no defective products leave the production site and reach the customer, aiming at higher manufacturing sustainability". According to recent reviews [1,3,4], ZDM is considered beyond traditional quality improvement approaches where data-driven approaches are applied to predict the future existence of a fault and implement actions to avoid or minimize its consequences. For this purpose, sensors and in-line inspection systems are deployed along the production line to enable process monitoring and control [2].

A comparison of traditional quality improvement strategies and ZDM strategies can be seen in Fig. 1, adapted from [1]. Traditional quality strategies rely on data from inspection stations at downstream and the feedback quality control loop may present issues such as delays in defect identification and difficulties in root-cause analysis. However, ZDM strategies use in-line inspection to proactively identify defects or potential defects and look for actions for minimize their impact downstream by implementing feedforward quality control loops.

Data-driven technologies used in ZDM lead to a more efficient quality assurance with capabilities for early detection of product defects and the identification of fault root causes. Some key enabling technologies for ZDM are digital twins which let the implementation of digital counterparts of product/process/systems [1,3]. In this field, the integration of data-driven and engineering-driven models is a key aspect of research for the development of reliable models with physical interpretation, where causality and root cause analysis can be conducted.

Current trends on ZDM promoted by the European Factories of the Future Research Association (EFFRA) and other institutions are

* Corresponding author.
E-mail addresses: rmoliner@uji.es (R. Moliner-Heredia), ipenarro@uji.es (I. Peñarrocha-Alós), abellan@uji.es (J.V. Abellán-Nebot).

https://doi.org/10.1016/j.jmsy.2023.02.005
Received 11 July 2022; Received in revised form 26 January 2023; Accepted 6 February 2023
Available online 17 February 2023

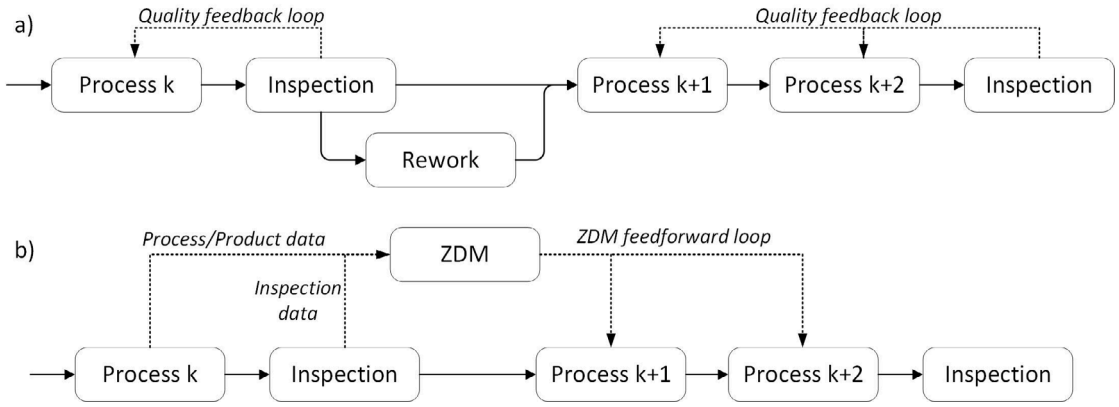


Fig. 1. A comparison of traditional quality improvement strategies (a) and ZDM strategies (b).

encouraging engineers to develop strategies for modeling, monitoring and controlling the output quality of complex manufacturing systems where several stages are inter-related to manufacture the product [5]. Examples of these processes, called Multistage Manufacturing Processes (MMP), are automotive body assemblies, machining lines for conducting multiple operations under different part orientation and fixtures, dielectric layer formulation processes in semiconductor industries and tile manufacturing processes [6–9]. To understand the complexity of these systems, consider the MMP of automobile assemblies where a typical body-in-white structure of the car is composed of 100–150 sheet metal parts. In this multistage assembly process one may find around 80–120 assembly stations where more than 1500 fixture locator are used to place the parts and more than 4000 welding points are executed [6]. As it can be expected, the amount of stages, variation sources and the complexity of the interactions at each stage makes the quality assurance of these processes a challenging task.

In order to model error propagation in MMP, the Stream-of-Variation (SoV) methodology has been successfully applied since the beginning of 21st century [6]. The kinematic relationships among fixtures, workpieces, tools (e.g. cutting tools or welding guns), are linearized and represented in matrix form which results in a variation propagation model in the form of a linear state-space model, where workpiece deviations in a given stage depend on the deviations caused by the variation sources in that stage (named fixture, machining or welding errors) and on the deviations of certain features manufactured in previous stages (named datum errors). This methodology was first developed to model the behavior of rigid sheet metal assembly processes, defining the main types of errors [10], which was later extended to compliant parts [11], and extended afterwards to 3D using Differential Motion Vectors (DMVs) [12]. The behavior of compliant composite parts for single and multistage assembly processes has also been modeled [13,14]. Multistage machining processes have also been investigated using SoV models. Originally defined using DMVs in [7], subsequent research has modeled the effects of machining-induced variations [15], general-purpose workholding devices [16] and general fixture layouts [17]. The concept of feedforward quality control loops from ZDM paradigm was investigated using the SoV model and in-line measurements. The prediction of part deviations at downstream stages let implement tooling compensation actions to correct or minimize their effects [18,19], or let apply quality rework loops [20]. These techniques require flexible manufacturing, optimal measurement sensor location [21] and appropriate models of the manufacturing system, which must be analyzed to verify its diagnosability [22] and compensability [23].

However, the accuracy of these models is limited due to linearizations and approximations, especially when there is a large amount of stages since these linearization-induced errors accumulate when calculating the error propagation model. Additionally, these methods can only take into account the general configuration of the manufacturing process, considering ideal geometries for each component of the process; in reality, each component may present slight differences depending on the manufacturer and brand. Thus, some elements of the complete model of the MMP may present divergences with respect to the real behavior of the process. To overcome this limitation, the physical model based on SoV methodology requires an adjustment or calibration using data from the process. This type of approaches where engineering-based models are calibrated have been studied in the literature under the term “hybrid modeling” and it has been identified as a big trend in the field of modeling manufacturing systems [24,25].

In this paper, we present a methodology to reduce the aforementioned divergences by slightly adjusting a physical variation propagation model of a MMP using collected measurements from the process and engineering knowledge. The complexity and dimensionality of this adjustment requires a numerical solution using optimization solvers. The adjustment is performed by minimizing the difference between the sample covariance of the output quality measurements and the expected covariance calculated with the model and the covariance of the variation sources, taking into account that only a given variation source covariance range is known. Prior knowledge, such as inspection measurement uncertainty and ranges of variation sources, is assumed to be known from backup data and/or equipment specifications. This knowledge is used to determine the optimization bounds. The nonlinear behavior of the objective function and the bounding conditions require convexification transformations and iterated optimizations in order to obtain a convergent solution using a convex optimization solver.

The main contribution of this paper is the definition of a methodology that combines physical models, data-driven methods and engineering knowledge to obtain an improved input–output variation propagation model of an MMP. For this purpose, different linearization methods have been applied to the objective function in order to ensure that a convex optimization solver can provide a solution within a finite time.

This paper is structured as follows. Section 2 presents the literature review related to model adjustment using engineering models and data. Section 3 introduces the dimensional error propagation model usually applied in MMPs and formulates the calibration problem under different assumptions. Section 4 describes the proposed methodology for model adjustment and Section 5 presents several indexes to evaluate the performance of the resulting model. A case study is shown in Section 6 to illustrate the application of the proposed methodology and finally, Section 7 presents the conclusions of the paper.

2. Literature review

Mechanistic models, also known as first principle models, are based on physical laws that define the behavior of the system. These models

are generally expensive in development, since one needs to derive equations from physical laws and, in many cases, simplification of the system is necessary to end with a practical mathematical expression. On the other hand, data-driven models also known as empirical or black-box models are based on the data obtained from the actual system in sufficient quantity and quality to estimate the relationships among inputs and outputs. While these models can be rapidly built with great accuracy using the data obtained from the actual system, the main drawbacks are the lack of physical interpretation of the model and thus, the limitation of its use on the working region where the data was acquired and under the same conditions. Merging physical-based and data-driven models, i.e. engineering knowledge and data, is becoming the next big trend in research and industry, since it leads to models with higher accuracy and interpretability and thus, to a better decision-making. In the literature, this field is commonly named hybrid modeling or grey-box modeling. Interesting reviews on this topic can be found in [24,26,27]. One type of hybrid modeling is model calibration, where "a low-fidelity mechanistic model is calibrated using both low-fidelity data (from the mechanistic equation) and the high-fidelity data (from the experiment) to generate high-fidelity output" [24]. When dealing with model calibration the following problems may arise [28]: (i) various combinations of input parameters may yield comparable fits to observed data, which is defined as the identifiability issue, (ii) the observed data contain some degree of error or uncertainty, and (iii) the most appropriate measure of agreement between observed data and model response is no obvious.

Let y be the output of the system, where the input variables are defined as $x = [x(1), \dots, x(p)]$. Consider the physical model defined as $f(x, \theta)$, where $\theta = [\theta(1), \dots, \theta(q)]$ are the calibration parameters. Then, the model error is defined as

$$e(x, \theta) = y(x) - f(x, \theta). \tag{1}$$

The adjustment of the calibration parameters seeks to minimize the sum of squared errors which may be a non-convex optimization problem and a formalized mathematical method using optimization techniques to minimize the error criterion is required. Methods from system identification based on Kalman filter have also been proposed for model calibration and are particularly well-suited when new data is available over time and process parameters may vary [28]. In the field of model calibration of computer models, Kennedy and O'Hagan presented in [29] a framework for model calibration that is considered a landmark work. In their research, the real output of the system is defined as

$$y(x) = f(x, \theta) + \delta(x) + \epsilon, \tag{2}$$

where $\delta(x)$ is a discrepancy function to capture errors between the model and the true output (model bias), and ϵ is the random error that captures the measurement noise and the effect of unaccounted variables in the system. Since computer models are considered inexact, if discrepancy is not included in model calibration, parameter estimation may be biased and over-confident parameter estimates are obtained. Model calibration of Eq. (2) is conducted using parametric or non-parametric methods. In parametric methods both $f(x, \theta)$ and $\delta(x)$ are modeled in a parametric way and thus, a specific functions with a finite number of parameters are defined in advance. Some authors have presented parametric methods to understand why a discrepancy exists in order to fix the wrong assumptions made in the model. For instance, Joseph and Melkote [30] proposed an engineering-driven parametric model to capture the discrepancy. In their work, the discrepancy function is further analyzed using simple main effects analysis and graphical plots in order to identify and correct the wrong assumptions of the model and make it more interpretable. Their approach was applied for modeling the cutting forces in a laser-assisted mechanical micro-machining process.

On the other hand, non-parametric methods are mainly based on Gaussian Process models for both the discrepancy function and a surrogate for the engineering model. Kennedy and O'Hagan [29] proposed a Bayesian method to compute the posterior prediction distribution of the physical process. Many other researchers proposed modifications from this work to improve model calibration using other non-parametric methods. For instance, Plumlee et al. [31] proposes a discrepancy function prior that is orthogonal to the gradient of the engineering model, which results in an improvement of the posterior distribution. As a brief comparison [32], parametric models need of prior knowledge to ensure that the model assumption can be explained physically, which may be difficult if there is no enough engineering knowledge of the process. However, model assumptions in non-parametric models are much relaxed although leads to less physical interpretation. Furthermore, the rate of convergence is much slower than the standard rate of parametric methods.

One important issue in model calibration is identifiability. Identifiability refers to "whether the single true value of a model's calibration parameters can theoretically be inferred based on the available data" [33]. The main problem is that different combination of parameters and discrepancy values can result in similar output values which leads to an identifiability issue of the calibration parameters θ. The identifiability issue cannot be solved through additional experimental observations, but the use of more informative prior knowledge of the system in terms of more accurate prior information about calibration parameters or including engineering-driven information in the discrepancy function can improve the identifiability [34,35]. The use of multiple responses in the model can also improve this issue [36]. Another way to circumvent this problem is to apply engineering knowledge to obtain a sparse model for the calibration problem as presented in Wang et al. [37]. In their work, the authors studied how to calibrate a computer model to match model predictions with the experimental observations of the structural load in a composite fuselage. In their proposal, most engineering values are considered close to their corresponding optimal values and only a subset of the model parameters require an adjustment. These parameters are called sensible variables to differentiate from the parameters that do not require adjustment because they have no influence on the output or because they are assumed to be close to the optimal one given the engineering knowledge. The complex computer model is then approximated by a model in linear form and as a result, the parameter identification becomes a convex optimization problem.

Previous methods apply the model calibration algorithm to a computer model where the cost of running is expensive, and thus, there is limited data from the model. In Liu [38], it is assumed that sufficient amount of data generated from both physical operations and computational experiments is available to perform parameter calibration. They presented the calibration problem as an stochastic optimization procedure to iteratively fine-tune the calibration parameters via a mini-batch optimization process. They formulated the parameter calibration problem as

$$\theta^* = \arg\min_{\theta} L(\theta), \tag{3}$$

where $L(\theta)$ is any form of loss function, although they applied the $L2$ norm due to its mathematical tractability. Given a data set $(y_i, x_i), i = 1, \dots, N$ collected from the physical system, the empirical loss becomes

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} [y_i(x_i) - f(x_i, \theta)]^T \cdot [y_i(x_i) - f(x_i, \theta)], \tag{4}$$

which refers to a nonlinear least squares estimation problem which is solved by a gradient descent algorithm with stratified sampling.

In the field of modeling MMPs, Sun et al. [39] studied the dimensional errors in multistage machining processes and they proposed a semi-parametric model to integrate engineering knowledge and measurement data for model calibration. In their model, the dimensional error at each stage depends on the previous dimensional errors and the tolerance range due to processing at each stage. Furthermore, a discrepancy function and random errors complete the model for

dimensional error propagation. In their work, the discrepancy function is closely related to machine tools and fixtures at different cutting tool positions and it is assumed to be a second order function. The Bayesian framework is then applied to estimate the parameters of the system.

In this paper, we deal with the problem of calibrating a model of error propagation in multistage manufacturing systems. Unlike previous research works, the calibration problem presented has specific characteristics that make their calibration challenging. First, the model is derived using the SoV methodology and thus, the model is a linear model derived from kinematic linearizations of fixture dimensions. Second, the data used for calibration is related to dimensional deviations acquired in quality control by inspection stations, but there is no data about the sources that produce these deviations. In other words, in Eq. (1) we have no information about x (i.e., fixture or tooling deviations) for a given set of inspection data, $y(x)$. However, we assume that a solid engineering knowledge about ranges of operation of the system is available, so prior information about values of θ, and x are given. Our problem is to adjust the parameters θ of the SoV model (i.e., the engineering model), which is defined as $f(x,\theta)$, in order to get a better explanation of the data acquired in the system, minimizing the discrepancy between the model and the data.

Notation

Let us define $A \in \mathbb{R}^{n\times n}$ as a matrix, and $a \in \mathbb{R}^{n}$ as a vector. When we refer to the structure of the model matrices, $A(i)$ and $a(i)$ refer to the values of A and a of the ith processed part, respectively. $a_{k,n}$ refers to the state of A for the nth locator or KPC deviation at stage k. Letter Σ represents a covariance matrix, and σ^2 an element of that matrix. Also, letter S represents a sample covariance matrix. Letters Σ, σ^2 and S can be accompanied by a subscript (e.g. Σ_z), which refers to a given assigned term z. Thus, $\sigma_a^2[q]$ refers to the variance of the qth element of a. Additionally, $A[p,q]$ refers to the element located on the p row and q column of A. A subscript after a dimensional counter (n_z) also refers to an assigned name z.

The diagonal of a square matrix is extracted using operator $diag(\cdot)$. Operator $diag^{-1}(\cdot)$ applied to a vector generates a diagonal square matrix whose diagonal contains the aforementioned vector. Operator $vec\,(A) \in \mathbb{R}^{n^2}$ returns the vectorization of A as a column. Given a symmetric A, operator $svec\,(A) \in \mathbb{R}^{n^2}$ returns the vectorization of the elements within and below the diagonal of A, expressed as a column. The Hadamard product of A and A is expressed as $A^{\circ 2}$.

When we explain numerical algorithms, $A_{(l)}$ and $a_{(l)}$ refer to the values of A and a during the l iteration. Expected values are denoted as $\boldsymbol{E}\{\cdot\}$. Let us define function $b = f(a)$, where $b \in \mathbb{R}^1$. The partial derivative of b with respect to vector a is expressed as $\frac{\partial b}{\partial a}$.

3. Problem statement

The objective of this paper is to present a methodology to adjust a physical linear input–output model of dimensional variation propagation in MMPs with a large amount of stages and/or with components that, due to their configuration, cannot be reliably modeled, using collected measurement data from shopfloor and engineering knowledge. Given that physical models of these MMPs may present divergences with respect to the real behavior of the process due to modeling approximations and differences between idealized and real components, the adjustment is performed by adapting the physical model to minimize these divergences.

In this problem, the following assumptions are considered:

Assumption 3.1. Data from the variation sources are not available, although approximated ranges of their covariances are available. This is a reasonable industrial practice since accuracy of fixtures and manufacturing operations may be estimated from technical specifications or capability studies.

Assumption 3.2. Model divergences are modeled as disturbances expressed as linear functions of the variation sources. This assumption is explained by the nature of the SoV model. This model is based on deviations of reference points using linear approximations provided that the deviations are much smaller than nominal values [6]. Therefore, a linear function can be used to model the divergences.

Assumption 3.3. The data used to adjust the model is collected from a MMP that is under statistical control (i.e. faultless process); thus, no other disturbances are considered in this methodology. This is also a common practice, for instance, prior to setting up a control chart for quality control, the process is verified to be under statistical control.

The output of the proposed adjustment methodology will be: (i) adjusted coefficients of the dimensional variation propagation model and (ii) the estimation of the actual covariance of the variation sources. The variation propagation model considered in this paper and additional assumptions to clarify the scope of the research are exposed in this section.

3.1. Variation propagation model and assumptions

The Stream-of-Variation methodology defines the variation propagation model of an M-stage MMP as a state-space model [6]. This model describes the effect of the variation sources on the dimensional deviations of the workpiece features, and consequently, how these feature deviations affect Key Product Characteristics (KPC), which are the most important dimensional and geometrical properties of a processed part, as they directly impact the output quality of the product. The model presents the form

$$x_k(i) = A_{k-1}\cdot x_{k-1}(i) + B_k\cdot u_k(i) + w_k(i), \tag{5a}$$

$$y_k(i) = C_k\cdot x_k(i) + v_k(i), \tag{5b}$$

where $k = \{1,\ 2,\ ...,\ M\}$ refers to the stage index. The feature deviations of part $i = \{1,\ ...,\ N\}$ after stage k are expressed by $x_k(i)$. The values of the variation sources that affect part i at stage k are represented by $u_k(i)$, and $w_k(i)$ represents unmodeled errors of the process. The deviation of the KPCs from nominal values (KPCd) that are inspected in stage k are represented by $y_k(i)$. Measurement noise is represented by $v_k(i)$. Matrices A_k, B_k and C_k are defined by the process layout and its characteristics and can be derived from process planning data as shown in [7].

This model can be rearranged into a linear input–output model of the MMP:

$$y_M(i) = \Gamma\cdot u(i) + v_M(i) + \omega(i), \tag{6}$$

where $y_M(i)$ is $y_k(i)$ at stage M, containing the n_y KPCd measurements in that stage ($y_M(i) \in \mathbb{R}^{n_y}$), $v_M(i)$ is $v_k(i)$ at stage M ($v_M(i) \in \mathbb{R}^{n_y}$), $u(i)$ contains all the n_u states of the variation sources of the MMP ($u(i) \in \mathbb{R}^{n_u}$), $\omega(i)$ includes the unmodeled disturbances, and Γ ($\Gamma \in \mathbb{R}^{n_y\times n_u}$) is the variation propagation matrix, which relates the impact of all variation sources on the KPC deviations using a linear relationship. Matrix Γ is the result of the organized products of matrices A, B and C of each stage [7] and it defines the general behavior of the MMP.

As explained in Assumption 3.3, no faults nor non-linear disturbances will be present for our adjusting methodology; thus, $\omega(i)$ will be omitted from now on. The divergences due to errors when developing physical models are considered proportional to the variation sources and thus, included within matrix Γ, as commented in Assumption 3.2.

From Eq. (6), vectors $y_M(i)$, $v_M(i)$ and $u(i)$ are then defined as

$$y_M(i) = \begin{bmatrix} y_{M,1}(i) \\ \vdots \\ y_{M,n_y}(i) \end{bmatrix},\ v_M(i) = \begin{bmatrix} v_{M,1}(i) \\ \vdots \\ v_{M,n_y}(i) \end{bmatrix},\ u(i) = \begin{bmatrix} u_1(i) \\ \vdots \\ u_{n_u}(i) \end{bmatrix}, \tag{7}$$

where $1,\ ...,\ n_u$ in $u_1,\ ...,\ u_{n_u}$ refer to an arbitrarily assigned numeration of the variation sources of the whole MMP.

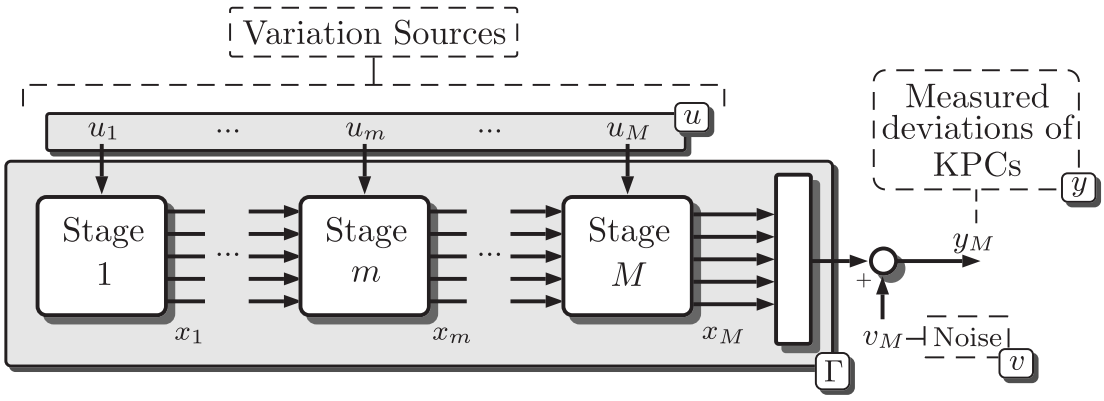


Fig. 2. Diagram of the variation source propagation in a multistage process.

Assumption 3.4. Matrix Γ is time invariant, so it remains constant during the natural working time of the MMP.

Thus, Eq. (6) is generalized into

$$y = \Gamma \cdot u + v, \tag{8}$$

and we define u, v and y as vectors containing the values of the states for each deviation source, measurement noise and inspected KPC deviations, respectively, for any given amount of parts:

$$u = \begin{bmatrix} u_1 \\ \vdots \\ u_{n_u} \end{bmatrix}, \; v = \begin{bmatrix} v_{M,1} \\ \vdots \\ v_{M,n_y} \end{bmatrix}, \; y = \begin{bmatrix} y_{M,1} \\ \vdots \\ y_{M,n_y} \end{bmatrix}. \tag{9}$$

Note that subscript M is now omitted, as it is implied.

In Fig. 2 we present a diagram that summarizes the different concepts we have exposed until now.

Assumption 3.5. The expected value of the states of the variation sources u and the measurement noise v is zero.

$$\boldsymbol{E}\{u\} = \begin{bmatrix} \boldsymbol{E}\{u_1\} \\ \vdots \\ \boldsymbol{E}\{u_{n_u}\} \end{bmatrix} = \mathbf{0}_{n_u \times 1}, \quad \boldsymbol{E}\{v\} = \begin{bmatrix} \boldsymbol{E}\{v_{M,1}\} \\ \vdots \\ \boldsymbol{E}\{v_{M,n_y}\} \end{bmatrix} = \mathbf{0}_{n_y \times 1}. \tag{10}$$

Assumption 3.6. Both u and v are independent variables:

$$\boldsymbol{E}\{u \cdot v^\top\} = \mathbf{0}_{n_u \times n_y}. \tag{11}$$

Taking into account the previous assumptions a Variance Variation Propagation Model (VVPM) is established, which models the behavior of the covariance matrices of the variation propagation model. Defining variables Σ_y, Σ_u and Σ_v as

$$\Sigma_y = \boldsymbol{E}\{yy^\top\}, \tag{12a}$$

$$\Sigma_u = \boldsymbol{E}\{uu^\top\}, \tag{12b}$$

$$\Sigma_v = \boldsymbol{E}\{vv^\top\}, \tag{12c}$$

where $\Sigma_y \in \mathbb{R}^{n_y \times n_y}$, $\Sigma_u \in \mathbb{R}^{n_u \times n_u}$ and $\Sigma_v \in \mathbb{R}^{n_y \times n_y}$, we define the VVPM as

$$\Sigma_y = \Gamma \; \Sigma_u \; \Gamma^\top + \Sigma_v. \tag{13}$$

Assumption 3.7. The linear input–output model in Eq. (6) is considered a stationary process [40], thus, the aforementioned variances will be constant during the normal operation of the multistage process, and no faults are present as stated in Assumption 3.3.

Additionally, we also consider the following assumptions:

Assumption 3.8. The variation sources are independent, so covariance matrix Σ_u is diagonal:

$$\Sigma_u = \begin{bmatrix} \sigma_u^2[1] & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_u^2[n_u] \end{bmatrix}, \tag{14}$$

where $\sigma_u^2[q]$ represents the variance of the qth variation source (u_q). We call $\vec{\Sigma}_u$ to the variable that collects the same terms of the diagonal of Σ_u.

$$\vec{\Sigma}_u = \left[\sigma_u^2[1], \; ..., \; \sigma_u^2[n_u]\right]^\top \equiv diag\left(\Sigma_u\right). \tag{15}$$

Assumption 3.9. Measurement noises are independent. Thus, covariance matrix Σ_v is diagonal:

$$\Sigma_v = \begin{bmatrix} \sigma_v^2[1] & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_v^2[n_y] \end{bmatrix}, \tag{16}$$

where $\sigma_v^2[p]$ represents the variance of the pth measuring instrument. We call $\vec{\Sigma}_v$ to the variable that contains the same terms of the diagonal of Σ_v.

$$\vec{\Sigma}_v = \left[\sigma_v^2[1], \; ..., \; \sigma_v^2[n_y]\right]^\top \equiv diag\left(\Sigma_v\right). \tag{17}$$

Assumption 3.10. The variance of the measurement noises is always known. It is calculated from its precision, obtained through the instruments' handbook or by performing a calibration of the instrument. Thus, covariance matrix Σ_v, and consequently, $\vec{\Sigma}_v$ are known terms in this paper.

Assumption 3.11. Covariance matrix Σ_y is a full symmetric matrix. We call $\vec{\Sigma}_y$ to the variable that contains the same terms of the diagonal of Σ_y.

$$\vec{\Sigma}_y \equiv diag\left(\Sigma_y\right). \tag{18}$$

Attending to Appendix A, we can state the following relationship for the diagonal elements of the covariance of the KPCd measurements.

$$\vec{\Sigma}_y = \Gamma^{\circ 2} \vec{\Sigma}_u + \vec{\Sigma}_v. \tag{19}$$

Assumption 3.12. The elements of Σ_y are magnitude orders greater than the elements of Σ_v.

3.2. Problem formulation

With the definitions and assumptions above we can reformulate the problem statement as follows. Given that:

- the MMP behaves as a linear input–output model (8) that leads to a VVPM model (13) assuming random independent variables;
- the process allows physical modeling and through geometrical analysis (as SoV methodology) we can obtain an initial estimation for variation propagation matrix in (8) that is close to the real one, but not enough to be used for diagnosis or control purposes;
- we can state some geometrical premises, and we have some engineering knowledge about the process that allows us to establish some bounds about the parameters of the model;
- a set of KPCd measurements from the real process can be acquired in standard conditions without any excitation for identification purposes (e.g. at the inspection station) and those measurements are affected by measurement noises with known variances;

then, obtain an accurate estimate for Γ and Σ_u in model (13). This model can be used latter, for instance, to estimate source variation covariance Σ_u to detect changes and state fault detection and isolation (see [40,41]), or to apply feedforward control (see [18]).

4. Proposed methodology for model adjustment

In this section we detail our proposal to adjust the variation propagation model for MMPs. The proposed methodology can be summarized as:

1. Obtain the variation propagation model according to the SoV methodology [6]. This is the initial model estimation for Γ, obtained through geometrical and engineering knowledge.
2. Set constraints to model values. According to engineering knowledge, fixed values or uncertainty ranges of model coefficients can be set up.
3. Acquire real data from the MMP to be modeled. For simplicity we assume that the data acquired is at the last stage of the process where a quality inspection is conducted, but the procedure can be applied at any intermediate stage.
4. Obtain an initial estimated value for the variation source covariance matrix, i.e., for Σ_u.
5. Set constraints to admissible ranges for the variation source covariance matrix during normal operation based on engineering knowledge (e.g., admissible tool wear in machining, admissible locating error due to assembly and wear, etc.).
6. Calibrate the model (i.e., Γ and Σ_u adjustments) by an optimization procedure that takes into account the following:
   - The model must minimize the difference between the acquired measured output variances and the predicted one through the model.
   - The optimization must use as an initial guess the values estimated in steps 1 and 4.
   - The optimization must search for the model respecting the knowledge set in steps 2 and 5.

In the next subsections we detail and formalize the steps of this proposed methodology. For now on Γ and Σ_u will refer to the real values to be estimated, and $\hat{\Gamma}$ and $\hat{\Sigma}_u$ will refer to their estimates, while Γ_0 and $\Sigma_{u,0}$ will refer to their initial estimation.

4.1. Step 1. Engineering-based modeling

As presented above, the SoV model is a engineering-based model to estimate dimensional variation propagation in MMPs. Given the manufacturing process plan (e.g., sequence of machining or assembly stages, machined/welded features, surfaces used for locating the part in the fixture, etc.), the variation propagation model is built in the form of a state space model with the definition of matrices A_k, B_k and C_k from Eq. (5). The state space model is expressed in the form of a linear input–output model as shown in Eq. (8), where Γ represents a reorganized product of those matrices. We call Γ_0 to this initial estimation of Γ. The internal elements of Γ_0 ($\Gamma_0 \in \mathbb{R}^{n_y \times n_u}$) present the following form:

$$\Gamma_0 = \begin{bmatrix} \Gamma_0[1,1] & \dots & \Gamma_0[1,n_u] \\ \vdots & \ddots & \vdots \\ \Gamma_0[n_y,1] & \dots & \Gamma_0[n_y,n_u] \end{bmatrix}. \tag{20}$$

Since modeling errors at each stage are accumulated, Γ_0 matrix may not present accurately the dimensional error propagation, especially when the number of stages of the MMP increases. In this paper we assume that, due to approximations and differences between idealized and real components, Γ_0 will differ from Γ, and, in some cases, some of the elements could notably differ due to, for instance, error during modeling phase and, thus, we need some adjustment.

4.2. Step 2. Model constraints

Now we propose several constraints that must be taken into account when searching for the model parameters. The formulation of these constraints depends on the available engineering knowledge, and they are detailed for any element in matrix Γ, denoted as $\Gamma[a,b]$.

1. **Sign of the elements of Γ.** The direction of the effect of each variation source on each KPC deviation can be defined in most cases. Thus, elements of matrix $\hat{\Gamma}$ can be constrained with the corresponding sign.

   $$C_1 : \Gamma[a,b] \geq 0 \;\; \| \;\; \Gamma[a,b] \leq 0. \tag{21}$$

2. **Bounds of Γ_0.** The upper and lower limits for Γ can be defined as a variation of the initial model Γ_0 within some given deviations.

   $$C_2 : \Gamma_0[a,b] + \mu_1 \leq \Gamma[a,b] \leq \Gamma_0[a,b] + \mu_2. \tag{22}$$

   In practice, this means we assume that the initial model estimation Γ_0 based on physical modeling is, to some extent, close to Γ. The elements that are obtained during the modeling phase by using approximations and linearizations, will require of these constraints, and, the rougher the approximations, the higher the values in μ_1 and μ_2 to be used.
3. **Null values.** Elements of Γ can be constrained to zero if it is clear that some variation sources cannot affect the corresponding KPC deviations.

   $$C_3 : \Gamma[a,b] = 0. \tag{23}$$

   This constraint can be also applied to fix some elements based on accurate engineering knowledge in Γ.
4. **Related terms.** If supported by geometrical assumptions, some different elements in Γ can be constrained to be proportional or related to each other.

   $$C_4 : \lambda_1 \Gamma[c,d] \leq \Gamma[a,b] \leq \lambda_2 \Gamma[c,d], \tag{24}$$

   where λ_1 and λ_2 are scalar values close to the expected relation between the aforementioned elements $[a,b]$ and $[c,d]$ of Γ.

With straightforward manipulations, constraints (21) to (24) can be reformulated into:

$$A_\Gamma \cdot vec(\Gamma) \leq b_\Gamma, \qquad C_\Gamma \cdot vec(\Gamma) = d_\Gamma, \tag{25}$$

where A_Γ and C_Γ are matrices, and b_Γ and d_Γ column vectors that contain the terms of the reorganized aforementioned constraints. The number of columns in A_Γ and C_Γ are equal to $n_y \cdot n_u$, i.e., the number of different elements in Γ, while the number of rows in A_Γ and b_Γ are equal to the number of inequality constraints, and the number of rows in C_Γ and d_Γ are equal to the number of equality constraints. The set of constraints formulated by Eq. (25) will bound the search for the final model estimation $\hat{\Gamma}$.

4.3. Step 3. Process data acquisition

To adjust the engineering model we use a set of data acquired from the MMP during production. This data set consists of KPCd measurements at the last stage ($k = M$). Given a batch of N processed parts, the corresponding KPCd measurements of the parts processed in that batch are expressed as matrix $Y \in \mathbb{R}^{n_y \times N}$:

$$Y = \begin{bmatrix} y_M(1) & \dots & y_M(N) \end{bmatrix} = \begin{bmatrix} y_{M,1}(1) & \dots & y_{M,1}(N) \\ & \ddots & \\ y_{M,n_y}(1) & \dots & y_{M,n_y}(N) \end{bmatrix}. \tag{26}$$

We can compute the sample covariance of the data set Y using the formula:

$$S_y = \frac{1}{N-1}\left(Y \cdot Y^\top\right). \tag{27}$$

We call $\vec{S}_y$ to the variable that extracts the values contained in the diagonal of S_y.

$$\vec{S}_y \equiv diag\left(S_y\right). \tag{28}$$

Thus, $S_y \in \mathbb{R}^{n_y \times n_y}$, and $\vec{S}_y \in \mathbb{R}^{n_y}$.

4.4. Step 4. Initial estimation of variation source covariance

As stated in Assumption 3.10, we know the theoretical covariance of the measurement noise Σ_v, which is directly related to the precision of the measuring instrument and thus, obtainable from the instruments' specifications. The order of magnitude of Σ_v is stated in Assumption 3.12 to be notably lower than the expected values of Σ_y, as required to perform faithful measurements.

Under Assumption 3.8, which states the independence of the variation sources, we search for a diagonal matrix $\vec{\Sigma}_u$. Thus, making use of Eq. (19), we can obtain an initial estimation of the variance of the variation sources, expressed as $\vec{\Sigma}_u$ (15). From the approximation

$$\vec{S}_y \approx \vec{\Sigma}_y \approx \Gamma_0^{\circ 2}\vec{\Sigma}_u + \vec{\Sigma}_v, \tag{29}$$

which is based on Eq. (19), we can then derive an estimation for the diagonal of the covariance matrix using least squares as

$$\vec{\Sigma}_{u,0} = \left(\Gamma_0^{\circ 2\top}\Gamma_0^{\circ 2}\right)^{-1}\Gamma_0^{\circ 2\top}\left(\vec{S}_y - \vec{\Sigma}_v\right). \tag{30}$$

4.5. Step 5. Constraints related to variation source covariance ranges

The covariance matrix for variation source Σ_u must be positive definite and, if some previous knowledge of the process is available, we can state the following bounds on the possible values for the variation source. The knowledge for the variation source can be obtained from vendor's specifications (e.g. accuracy of fixture locators) as well as from some backup data if, for instance, the fixtures have been already used and analyzed in another manufacturing or assembly usage. Furthermore, as stated before in Assumption 3.8, we force this matrix to be diagonal to reinforce the independence of the variation sources. With these premises, we state the following constraints for the variation source covariance matrix:

1. **Independence of the variation sources.** Matrix Σ_u can be expressed only with its diagonal elements (called $\vec{\Sigma}_u$) through

$$C_5 : \Sigma_u = diag^{-1}(\vec{\Sigma}_u). \tag{31}$$

2. **Positive variance.** The values of $\vec{\Sigma}_u$ are positive, as it represents the variance of the variation sources.

$$C_6 : \vec{\Sigma}_u \geq \mathbf{0}. \tag{32}$$

3. **Backup data.** The expected limits of $\vec{\Sigma}_u$ can be bounded if backup data from other similar processes is available.

$$C_7 : \vec{\Sigma}_{uBU_1} \leq \vec{\Sigma}_u \leq \vec{\Sigma}_{uBU_2}, \tag{33}$$

where $\vec{\Sigma}_{uBU_1}$ and $\vec{\Sigma}_{uBU_2}$ are vectors containing backup data.

Additional engineering-based constraints can be included if deemed necessary.

Constraints (31) to (33) can be reformulated into:

$$A_u \cdot \vec{\Sigma}_u \leq b_u, \qquad C_u \cdot \vec{\Sigma}_u = d_u, \tag{34}$$

where A_u and C_u are matrices, and b_u and d_u column vectors, which contain the terms of the reorganized aforementioned constraints. The number of columns in A_u and C_u are equal to n_u, the number of rows in A_u and b_u are equal to the number of inequality constraints, and the number of rows in C_u and d_u are equal to the number of equality constraints. The set of constraints formulated by Eq. (34) will bound the search for the final model estimation $\hat{\vec{\Sigma}}_u$.

4.6. Step 6. Model calibration through optimization

As our goal is to obtain a model that can be latter used, for instance, for diagnosis purposes or feedforward quality control, that model must be able to predict the KPCd measurement covariances (Σ_y) using Eq. (19) with high accuracy. For that reason, the initial idea for model adjustment is the minimization of the difference between the real covariance Σ_y and its estimated value $\hat{\Sigma}_y$ by tuning our model parameters, i.e., $\hat{\Gamma}$ and $\hat{\vec{\Sigma}}_u$. Matrix $\hat{\Sigma}_y$ is a function of our estimated parameters and according to Eq. (19) this function is expressed as

$$\hat{\Sigma}_y(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \hat{\Gamma} diag^{-1}(\hat{\vec{\Sigma}}_u)\hat{\Gamma}^\top + \Sigma_v, \tag{35}$$

where Σ_v is known, as stated in Assumption 3.10.

On the other hand, matrix Σ_y is not known and we approximate its value to the sample covariance S_y from Eq. (27). Then, the model adjustment seeks to minimize the difference between S_y and its predicted value $\hat{\Sigma}_y$ from Eq. (35). As S_y and its predicted value are symmetric matrices, we use the operator *svec*, that gathers unique matrix elements in a column vector. With these premises, we propose the following optimization problem to adjust the model parameters, where the decision variables are matrix $\hat{\Gamma}$ and vector $\hat{\vec{\Sigma}}_u$:

$$(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \arg\min_{\hat{\Gamma}, \hat{\vec{\Sigma}}_u} \|e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)\|_2^2 \tag{36a}$$

$$\text{s.t. } A_\Gamma \cdot vec(\hat{\Gamma}) \leq b_\Gamma, \tag{36b}$$

$$C_\Gamma \cdot vec(\hat{\Gamma}) = d_\Gamma, \tag{36c}$$

$$A_u \cdot \hat{\vec{\Sigma}}_u \leq b_u, \tag{36d}$$

$$C_u \cdot \hat{\vec{\Sigma}}_u = d_u, \tag{36e}$$

where $\|\cdot\|_2$ refers to the vector 2-norm, and thus, $\|\cdot\|_2^2$ refers to the summation of all the squared elements of the previous vector, that can be expressed as

$$\|e(\cdot)\|_2^2 = e(\cdot)^\top e(\cdot). \tag{37}$$

Function $e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)$ constructs the vector containing the difference between unique elements in the sampled output covariance S_y, and the estimated one $\hat{\Sigma}_y$ through $\hat{\Gamma}$ and $\hat{\vec{\Sigma}}_u$, i.e.,

$$e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u) = svec(S_y - \hat{\Sigma}_y(\hat{\Gamma}, \hat{\vec{\Sigma}}_u)), \tag{38}$$

with $\hat{\Sigma}_y$ defined in Eq. (35). As $e(\cdot)$ is the result of a symmetrical vectorization, $e(\cdot) \in \mathbb{R}^{n_{Sy}}$ is a column vector, with $n_{Sy} = n_y(n_y+1)/2$.

It should be noted that the function to be minimized in Eq. (36a), i.e., $\|e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)\|_2^2$, depends on the elements included in the decision matrix $\hat{\Gamma}$ and vector $\hat{\vec{\Sigma}}_u$ and it becomes a 6th order polynomial on the

decision variables. The number of decision variables is $n_u \cdot (n_y + 1)$. The proposed model parameters estimation through problem (36a) requires solving a polynomial optimization of high degree with lots of decision variables. There are several methods and available tools to solve polynomial optimization problems through semidefinite programming relaxations that may converge to the optimal value (see [42–44]). In order to solve such a complex optimization problem, we propose to use as initial guess Γ_0 for the model and $\Sigma_{u,0}$ for the variation source obtained in previous steps. We also want to remark that the initial guess Γ_0 and $\Sigma_{u,0}$ do not necessarily must fulfill constraints (25) and (34). If that is the case, one should recompute the initial estimate, as some of the aforementioned tools require an initial feasible estimated solution.

In an attempt of using those methods and the available tools for our problem, we have run into numerical problems derived of the high required computational burden for the degree and number of variables involved in our problem. As a polynomial optimization problem, it is a nonlinear optimization problem, so we could also use general (and expensive) nonlinear solvers, that may fail for the same reasons above.

Then, in this work, we propose in the following section an alternative approach to solve our polynomial optimization problem using both a reformulation of the problem to decrease the polynomial order, and a sequence of approximations through linearization. The idea is to formulate the problem in such a way that nonlinear solvers (that may not find a solution, or that may be too expensive or hard to tune) are avoided, and only generic standard convex optimization tools (that lead to a unique solution if the problems are properly formulated) are used. However, this will require some iteration procedure over convex problems to reach a solution.

4.6.1. Optimization solution through convexification

Here we present our proposal to numerically solve problem (36) iteratively. First, let us define index j to denote the number of iteration. Then, let us define new decision variables that represent the difference between an initial guess and the obtained solution at each iteration of the optimization problem as

$$\begin{cases} \Delta\Gamma_{(j)} = \hat{\Gamma}_{(j+1)} - \hat{\Gamma}_{(j)}, \\ \Delta\hat{\Sigma}_{u(j)} = \hat{\bar{\Sigma}}_{u(j+1)} - \hat{\bar{\Sigma}}_{u(j)}. \end{cases} \tag{39}$$

Now let us express small changes in the error function as a function of small changes in decision variables through a first order Taylor series approximation as

$$e(S_y, \hat{\Gamma}_{(j+1)}, \hat{\bar{\Sigma}}_{u(j+1)}) \approx e(S_y, \hat{\Gamma}_{(j)}, \hat{\bar{\Sigma}}_{u(j)}) + K_{\Gamma,j} \cdot vec(\Delta\Gamma_{(j)}) + K_{u,j} \cdot \Delta\hat{\Sigma}_{u(j)} \tag{40}$$

with $K_{\Gamma,j}$ and $K_{u,j}$ matrix gains that can be computed numerically as

$$\begin{aligned} K_{\Gamma,j} &= \frac{\partial e(S_y, \hat{\Gamma}, \hat{\bar{\Sigma}}_u)}{\partial\left(vec(\hat{\Gamma})\right)}\bigg|_{\hat{\Gamma}=\hat{\Gamma}_{(j)}, \vec{\Sigma}_u=\hat{\bar{\Sigma}}_{u(j)}}, \\ K_{u,j} &= \frac{\partial e(S_y, \hat{\Gamma}, \hat{\bar{\Sigma}}_u)}{\partial\left(\hat{\bar{\Sigma}}_u\right)}\bigg|_{\hat{\Gamma}=\hat{\Gamma}_{(j)}, \vec{\Sigma}_u=\hat{\bar{\Sigma}}_{u(j)}} \end{aligned} \tag{41}$$

With the use of Schur complements (see Appendix B.1) and the linearization of the error function (see Appendix B.2), the optimization problem in Eq. (36) can be solved iteratively through the solution of convex optimization problems as follows.

1. Set $j = 0$, $\hat{\Gamma}_{(0)} = \Gamma_0$ and $\hat{\bar{\Sigma}}_{u(0)} = \vec{\Sigma}_{u,0}$.
2. Obtain $\Delta\Gamma_{(j)}$ and $\Delta\hat{\Sigma}_{u(j)}$ by solving the following convex optimization problem

$$(\Delta\Gamma_{(j)}, \Delta\hat{\Sigma}_{u(j)}) = \arg\min_{\Delta\Gamma, \Delta\hat{\Sigma}_u, t} t_{(j)} \tag{42a}$$

$$s.t. \begin{bmatrix} t_{(j)} & \varepsilon_{(j+1)}^{\top} \\ \varepsilon_{(j+1)} & I_{n_{Sy}} \end{bmatrix} \succeq 0, \tag{42b}$$

$$A_\Gamma \cdot vec(\hat{\Gamma}_{(j)} + \Delta\Gamma_{(j)}) \leq b_\Gamma, \tag{42c}$$

$$C_\Gamma \cdot vec(\hat{\Gamma}_{(j)} + \Delta\Gamma_{(j)}) = d_\Gamma, \tag{42d}$$

$$A_u \cdot \left(\hat{\bar{\Sigma}}_{u(j)} + \Delta\hat{\Sigma}_{u(j)}\right) \leq b_u, \tag{42e}$$

$$C_u \cdot \left(\hat{\bar{\Sigma}}_{u(j)} + \Delta\hat{\Sigma}_{u(j)}\right) = d_u, \tag{42f}$$

where

$$\varepsilon_{(j+1)} = e(S_y, \hat{\Gamma}_{(j)}, \hat{\bar{\Sigma}}_{u(j)}) + K_{\Gamma,j} \cdot vec(\Delta\Gamma_{(j)}) + K_{u,j} \cdot \Delta\hat{\Sigma}_{u(j)}, \tag{42g}$$

is the linear approximation of $e(S_y, \hat{\Gamma}_{(j+1)}, \hat{\bar{\Sigma}}_{u(j+1)})$ and where $K_{\Gamma,j}$ and $K_{u,j}$ are given by Eq. (41)

3. Update the actual model estimation as

$$\begin{cases} \hat{\Gamma}_{(j+1)} = \hat{\Gamma}_{(j)} + \Delta\Gamma_{(j)}, \\ \hat{\bar{\Sigma}}_{u(j+1)} = \hat{\bar{\Sigma}}_{u(j)} + \Delta\hat{\Sigma}_{u(j)}. \end{cases} \tag{43}$$

4. If $\|vec(\Delta\Gamma_{(j)})\|_2^2 \geq \delta$ (with $\delta > 0$ a small quantity), set $j = j+1$ and go back to step 2; else, define the final adjusted model elements as

$$\begin{cases} \hat{\Gamma} \equiv \hat{\Gamma}_{(j+1)}, \\ \hat{\bar{\Sigma}}_u \equiv \hat{\bar{\Sigma}}_{u(j+1)}, \end{cases} \tag{44}$$

and end the algorithm.

Note that by using the Schur complement and the linearization step (see Appendix B), we have reduced the order of the polynomial to a linear function and we have included a new constraint of the type Linear Matrix Inequality in (42b) that can be faced by any standard solver for convex optimization.

4.7. Methodology summary and limitations

The Algorithm 1 summarizes the proposed algorithm for model adjustment. Fig. 3 overviews the proposed methodology with the input data required and the output data related to model adjustment. In summary, the proposed methodology can be read as follows. Once an initial model guess, process data and engineering knowledge have been established, obtain matrices $\hat{\Gamma}$ and $\hat{\Sigma}_u$ that minimize some metric that measures the error between Σ_y and its estimation $\hat{\Sigma}_y$ through the model (13), fulfilling the derived engineering-based constraints and using S_y as an approximation for Σ_y, i.e.,

$$[\hat{\Gamma},\ \hat{\Sigma}_u] = f\left(S_y,\ \Gamma_0\right). \tag{45}$$

One of the limitations of the proposed methodology is that the estimation problem presents multiple feasible solutions by model parameters escalations (i.e., an identifiability issue). For instance, one can achieve the same predicted output covariance matrix with different model parameters, i.e.,

$$\hat{\Sigma}_y = \Gamma \cdot \Sigma_u \cdot \Gamma^T + \Sigma_v = \Gamma' \cdot \Sigma_u' \cdot (\Gamma')^T + \Sigma_v, \tag{46a}$$

with any scalar scaling factor $\alpha \in \mathbf{R}$ ($\alpha \neq 0$) such that the following is fulfilled

$$\Gamma' = \alpha\,\Gamma, \qquad \Sigma_u' = \frac{1}{\alpha^2}\Sigma_u. \tag{46b}$$

In that sense, our method is devoted to search for the model that is closer to the initial estimated one, and it is of great importance to state properly the bounds on the search (the constraints in the optimization problem) to avoid that the optimizer iteration move around equivalent (scaled) solutions.

A modification of the algorithm to limit large changes of the initial parameters that may result in similar results can be made by including a regularization process where the distance of the searched model parameters with respect to its initial value is penalized. For that, one can

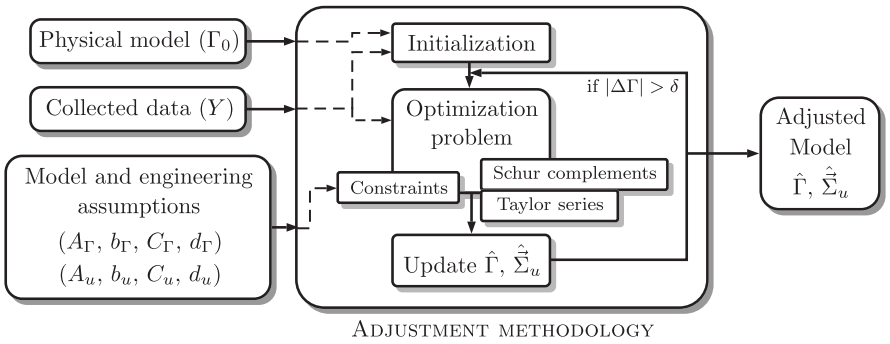


Fig. 3. Summary diagram of the adjustment methodology.

Algorithm 1 Proposed model adjustment algorithm

1. Apply physical modeling to obtain Γ_0 (20).
2. Construct matrices A_Γ, b_Γ, C_Γ, d_Γ (25) to set known model uncertainty ranges and other engineering-based assumptions.
3. Acquire process data Y and obtain sample covariance S_y (27).
4. Obtain initial estimation of variation source covariance matrix $\Sigma_{u,0}$ (30).
5. Construct A_u, b_u, C_u and d_u to set known variation covariance uncertainty ranges (34).
6. Calibrate the model (i.e., obtain $\hat{\Gamma}$ and $\hat{\Sigma}_u$) through optimization problem (36) using Γ_0 and $\Sigma_{u,0}$ as an initial guess. To solve the optimization problem through standard semidefinite programming, follow the steps for iterative procedure through convex optimization problems as:
   (a) Initialize with $j = 0$, $\hat{\Gamma}_{(j)} = \Gamma_0$, and $\hat{\vec{\Sigma}}_{u(j)} = \Sigma_{u,0}$.
   (b) Solve optimization (42).
   (c) Update model estimation $\hat{\Gamma}_{(j+1)}$ and $\hat{\vec{\Sigma}}_{u(j+1)}$ with (43)
   (d) Check exit conditions:
      i. If fulfilled: set $\hat{\Gamma}$ and $\hat{\Sigma}_u$ with (44) and exit.
      ii. Else: set $j = j + 1$ ad return to (b).

change the objective function (36a) in the optimization problem (36) to

$$(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \arg\min_{\hat{\Gamma}, \hat{\vec{\Sigma}}_u} \|e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)\|_2^2 + \beta_\Gamma \|vec(\hat{\Gamma} - \Gamma_0)\|_2^2 + \beta_u \|\hat{\vec{\Sigma}}_u - \vec{\Sigma}_{u,0}\|_2^2 \tag{47a}$$

$$\text{s.t. (36d)–(36e)} \tag{47b}$$

where $\beta_\Gamma > 0$ and $\beta_u > 0$ are scalar weighting factors chosen by the user for the penalization. The application of this regularization in the iterative approach through convex optimizations must refer to the incremental decision variables ($\Delta\Gamma_{(j)}$ and $\Delta\hat{\Sigma}_{u(j)}$), that directly measure the distance from an initial solution to a new one, and, then, constraint (42b) (the one obtained through linearization and Schur complements) translates to

$$\begin{bmatrix} t_{(j)} & \varepsilon_{(j+1)}^\top & vec(\Delta\Gamma_{(j)})^\top & \Delta\hat{\Sigma}_{u(j)}^\top \\ \varepsilon_{(j+1)} & I_{n_{Sy}} & 0 & 0 \\ vec(\Delta\Gamma_{(j)}) & 0 & \frac{1}{\beta_\Gamma} I_{n_y \cdot n_u} & 0 \\ \Delta\hat{\Sigma}_{u(j)} & 0 & 0 & \frac{1}{\beta_u} I_{n_u} \end{bmatrix} \succeq 0. \tag{48}$$

This approach can also be extended to penalize with a different weighting factor each of the decision variables. In the case of the nonlinear optimization proposal, this implies substituting the term $\beta_\Gamma \|vec(\hat{\Gamma} - \Gamma_0)\|_2^2 + \beta_u \|\hat{\vec{\Sigma}}_u - \vec{\Sigma}_{u,0}\|_2^2$ in the objective function by

$$vec(\hat{\Gamma} - \Gamma_0)^\top \cdot \beta_\Gamma \cdot vec(\hat{\Gamma} - \Gamma_0) + (\hat{\vec{\Sigma}}_u - \vec{\Sigma}_{u,0})^\top \cdot \beta_u \cdot (\hat{\vec{\Sigma}}_u - \vec{\Sigma}_{u,0})$$

where $\beta_\Gamma \succ 0$ and $\beta_u \succ 0$ are now positive-definite diagonal matrices

$$\beta_\Gamma = diag([\beta_\Gamma[1] \cdots \beta_\Gamma[n_y \cdot n_u]]), \qquad \beta_u = diag([\beta_u[1] \cdots \beta_u[n_u]]).$$

The Schur complement in the convexified approximation leads in this case to

$$\begin{bmatrix} t_{(j)} & \varepsilon_{(j+1)}^\top & vec(\Delta\Gamma_{(j)})^\top & \Delta\hat{\Sigma}_{u(j)}^\top \\ \varepsilon_{(j+1)} & I_{n_{Sy}} & 0 & 0 \\ vec(\Delta\Gamma_{(j)}) & 0 & [\beta_\Gamma]^{-1} & 0 \\ \Delta\hat{\Sigma}_{u(j)} & 0 & 0 & [\beta_u]^{-1} \end{bmatrix} \succeq 0. \tag{49}$$

On the other hand, as the data used for the adjustment is obtained in standard operation without controlled excitation for identification purposes, we must understand that the fitted model will be able to catch the variation source propagation, i.e., its usage for accurate fault diagnosis is assured, but its use for feedforward control (where the goal is to correct deviations in average) could be more limited if the initial model is far from the correct values and the bounds are too relaxed.

In the next section we address how to evaluate the goodness of the model with the available data set Y.

5. Indexes for model validation

The proposed adjustment algorithm shown in Eqs. (42)–(44) is based on an iterative optimization procedure, and it uses an estimation of Σ_y by means of S_y, and a metric that estimates the sum of the square of the difference of the elements of Σ_y and $\hat{\Sigma}_y(\hat{\Gamma}, \hat{\Sigma}_u)$. The output of the proposed algorithm is an adjusted model of the process. However, in order to validate the model, we must assess different data sets.

In that sense, we propose to validate the model using a training–testing procedure. From a theoretical validation point of view, we will assume that we have available the theoretical real values of matrices Σ_y, Σ_u, Σ_v and Γ. We will also assume that we have two sets of available data, one of them used in the estimation algorithm (training set, that leads to sample covariance S_{yTr}), and another one used to evaluate the goodness of the adjustment (test set, with sample covariance S_{yTs}). Thus, adjusting the model with the training set will yield the values of $\hat{\Gamma}$ and $\hat{\vec{\Sigma}}_u$:

$$[\hat{\Gamma}, \ \hat{\vec{\Sigma}}_u] = f\left(S_{yTr}, \ \Gamma_0\right). \tag{50}$$

The model is validated using performance indexes, which compare the sample covariance values and the theoretical values with the estimated covariance of the KPCd measurements ($\hat{\Sigma}_y$), obtained using Eq. (35). The performance indexes are vectored using symmetrical vectorization in order to avoid the duplication of the effect of the elements located outside the diagonal. The indexes are defined as follows:

- Index I_1 evaluates the performance of the algorithm on estimating the theoretical covariance matrix of the KPCd measurements. Given that the covariance of the measurement noise Σ_v is known, as per Assumption 3.10, index I_1 presents this form:

$$I_1 = \|e(\Sigma_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)\|_2^2. \tag{51}$$

with $e(\Sigma_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u) = svec(\Sigma_y - \hat{\Sigma}_y(\hat{\Gamma}, \hat{\vec{\Sigma}}_u))$, i.e., error function from Eq. (38) evaluated with Σ_y instead of S_y.

- Index I_2 expresses the proportion of the fourth root of index I_1 and the l_2^2-norm of the symmetrical vectored elements of $\Sigma_y - \Sigma_v$, in percentage:

$$I_2 = \left(\frac{I_1}{\|svec\left(\Sigma_y - \Sigma_v\right)\|_2^2} \right)^{\frac{1}{4}} \cdot 100. \tag{52}$$

Roughly speaking, this index evaluates the error on estimating Σ_y with respect to the explained variance in Σ_y when there are no measurement noises.

- Index I_3 evaluates the performance of the algorithm on estimating the theoretical standard deviations of the KPCd measurements (i.e. the terms of the diagonal) by comparing the maximum error of the estimation.

$$I_3 = \max\left(\left| \sqrt{\vec{\Sigma}_y} - \sqrt{\hat{\vec{\Sigma}}_y} \right| \right). \tag{53}$$

- Index I_4 expresses the maximum ratio of the estimation error of the standard deviations with respect to the standard deviation of $\vec{\Sigma}_y$:

$$I_4 = \max\left(\frac{\left| \sqrt{\vec{\Sigma}_y} - \sqrt{\hat{\vec{\Sigma}}_y} \right|}{\sqrt{\vec{\Sigma}_y}} \right) \cdot 100. \tag{54}$$

Note that the previous quotient refers to an element-wise division of the vector components.

It should be noted that previous indexes are applied to evaluate the performance of the proposed adjustment methodology in a theoretical way, since it requires to know the real values of Σ_y, Σ_u, Σ_v and Γ. For practitioners, the methodology can be validated at the shop floor level using the following index:

- Index I_P evaluates the performance of the algorithm on estimating the values of the covariance matrix of the KPCd measurements obtained from the testing set.

$$I_P = \|e(S_{yTs}, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)\|_2^2. \tag{55}$$

Index I_P is the only index that can be assessed in practical cases, but the other indexes are the ones that would provide real information of the goodness of the fit.

In the next section a case study is presented to illustrate the performance of our adjustment methodology using these indexes. We will show through numerical examples how I_P can be related with other indexes and how it can help to decide if a proper adjusted model has been reached.

6. Case study

6.1. Multistage manufacturing process

In order to validate the performance of the proposed adjustment methodology, we have conducted a case study of a multistage manufacturing process. In this case study, we consider that matrix Γ_0, which is derived using the SoV methodology, presents notable deviations with respect to the real values of Γ as a result of simplifications and unreliable modelizations of some parts of the process. In this case, for the sake of dimensional simplicity but without loss of generality, we have used as Γ an adapted version of the matrix presented in a case study from [45]:

$$\Gamma = \begin{bmatrix} 0.093 & 0.577 & -0.120 \\ -0.093 & 0 & 0.843 \\ 0.093 & 0.577 & -0.120 \\ 0.647 & 0 & -0.120 \\ -0.370 & 0.577 & 0.482 \\ 0.647 & 0 & -0.120 \end{bmatrix}. \tag{56}$$

For this MMP, we assume that u presents a Gaussian distribution with zero mean and a variance $\sigma_u^2 = 1.111 \cdot 10^3$ μm^2. Thus, considering that all variation sources behave with the same variance, $\Sigma_u = \sigma_u^2 \mathbf{I}$. We also consider that the following constraints are known from the process, and, therefore, they are used in the optimization procedure to adjust the model:

- The values of Γ that we know that are null have been forced to zero in $\hat{\Gamma}$, and several values that we know that are positive have been forced to be greater or equal to zero. Additionally, we have constrained the values of $\hat{\Gamma}$ to be within a 30% range with respect to the initialization matrix Γ_0.
- The values of $\hat{\vec{\Sigma}}_u$ have been forced to be positive, and within the 500−1500 μm^2 range, which would be obtained using backup data from other similar process.

With this considerations, we have constructed matrices A_Γ, b_Γ, C_Γ, d_Γ, A_u, b_u, C_u and d_u.

6.2. Simulated shop floor data and model initialization

The proposed model adjustment uses data from the MMP. In this case study, the simulated values of the KPCd measurements for each part at the last stage ($y_M(i)$) are generated using the real value of Γ detailed in Eq. (56) and the values of $u(i)$ and $v(i)$ are randomly generated to include the effect of the variation sources, as well as measurement noises. With this, we have generated our data set $\{y_M(i)\}$. We consider that all KPCd are measured with the same instrument, so the variances of all measurement noises are identical ($\Sigma_v = \sigma_v^2 \mathbf{I}$).

To analyze the impact of different levels of measurement noise in the proposed model calibration methodology, five experimental scenarios, named A, B, C, D, E, are studied. The measurement noises introduced at each experimental scenario are shown in Table 1. To compare the level of measurement noise with the KPCd at each scenario, the table also includes the proportion in percentage between the standard deviations of the measurement noise and the standard deviations of the KPCd measurements, defined as:

$$Prop_v\ (\%) = \left(\frac{svec\left(\Sigma_v\right)^\top svec\left(\Sigma_v\right)}{svec\left(\Sigma_y\right)^\top svec\left(\Sigma_y\right)} \right)^{\frac{1}{4}} \cdot 100.$$

Two data sets are simulated in the analysis. The training data set refers to the data that is obtained from the shopfloor and it is used to run the adjustment algorithm. The testing data refers to the data that is used after model calibration, and it is used to calculate index I_P. The training and testing data sets applied in the simulations are shown in Table 2.

The simulated data is obtained by applying Monte Carlo simulations. For each experimental scenario and data sets, 200 Monte Carlo

Table 1
Experiment scenarios under analysis in the case study.

Exp.	A	B	C	D	E
σ_v^2 (μm^2)	1	4	16	64	256
$Prop_v$ (%)	3.72	7.42	14.75	28.77	52.36

Table 2
Training and testing data sets used in the simulation.

Number of Monte Carlo iterations	200
Number of parts in the training set (N_{tr})	{1, 5, 15, …, 95, 100, 200, …, 2100}
Number of parts in the testing set (N_{ts})	{350, 500, …, 1100, 1250}

simulations are run. Since we assume that a initial value of the model, Γ_0, is given from the SoV derivation, Γ_0 is randomly generated within the 30% range of the real value Γ for each Monte Carlo simulation, thus fulfilling one of the previous mentioned constraints.

6.3. Results of the adjusted model: performance indexes

The proposed model adjustment is applied for each experimental scenario and data sets. For each Monte Carlo simulation and, in order to adjust the model by solving the proposed optimization problem, we use the YALMIP parser [46] and the optimization software *mosek*. Our two-core i7 laptop takes around 8 s to solve the proposed optimization problem, which in turn performs 8 iterations in order to converge the results. This indicates that solving (42) takes around 1 s, and, in average, 8 iterations (from $j = 0$ to $j = 7$ in (42)) were required at each Monte Carlo simulation.

The performance indexes calculated during the experiments are shown as follows:

- Index I_1 is presented in Fig. 4(a) for the different experimental scenarios presented in Table 1. As it can be observed, index I_1 is reduced when the number of processed parts used in the training set increases. A detailed version of the first stretch of the values of Index I_1 with respect to training data size is presented in Fig. 4(b), where a sudden peak and a swift descent can be observed for all experiment scenarios. In both Figs. 4(a) and 4(b) it can be observed that the performance of the proposed methodology is lower when the standard deviation of the measurement noise is higher.
- Index I_2 is presented in Fig. 5 for the different experimental cases. Its general behavior is very similar to index I_1, as we have higher values for this performance index when the considered measurement noise is higher.
- Indexes I_3 and I_4 are presented in Figs. 6 and 7, respectively, for the proposed measurement noises. They show a similar behavior to the other indexes, but they lack the initial peak, as the initial point is higher because maximum points are searched here instead. These indexes are less affected by the variance of the measurement noise.
- Index I_P is presented in Fig. 8(a) for measurement noises of $\sigma_v^2 = 4$ μm^2 (Exp. B) and in Fig. 8(b) for measurement noises of $\sigma_v^2 = 256$ μm^2 (Exp. C), using several amounts of processed parts in the testing set. The values of Index I_P for different training test size is shown together with Index I_1, which presents a clearly lower value for all sizes N_{ts}. This difference increases as the number of parts used in the training set increases. In fact, we see that I_P shows a negligible sensitivity with respect to N_{ts} for values above 1000.

Finally, in order to show an example of the results of the adjustment algorithm, we present the real parameters, its initial estimation and the adjusted ones with our proposal. The results presented below are from the experimental scenario B with $N_{ts} = 2100$ parts in the training set. The real data are Γ and $\vec{\Sigma}_u$, the initial values from engineering knowledge are Γ_0 and $\hat{\vec{\Sigma}}_{u(0)}$, and the final adjusted parameters are $\hat{\Gamma}_B$ and $\hat{\vec{\Sigma}}_u$.

$$\Gamma_0 = \begin{bmatrix} 0.1102 & 0.7344 & -0.1367 \\ -0.0951 & 0 & 0.9546 \\ 0.1078 & 0.5034 & -0.1492 \\ 0.5892 & 0 & -0.1481 \\ -0.3366 & 0.5576 & 0.4142 \\ 0.8015 & 0 & -0.1503 \end{bmatrix},$$

$$\hat{\Gamma}_B = \begin{bmatrix} 0.0999 & 0.5844 & -0.1433 \\ -0.0957 & 0 & 0.8501 \\ 0.0990 & 0.5919 & -0.1359 \\ 0.6737 & 0 & -0.1491 \\ -0.3805 & 0.5892 & 0.4871 \\ 0.6666 & 0 & -0.1541 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} 0.093 & 0.577 & -0.120 \\ -0.093 & 0 & 0.843 \\ 0.093 & 0.577 & -0.120 \\ 0.647 & 0 & -0.120 \\ -0.370 & 0.577 & 0.482 \\ 0.647 & 0 & -0.120 \end{bmatrix}. \tag{57}$$

$$\hat{\vec{\Sigma}}_{u(0)} = \begin{bmatrix} 898.2 \\ 1043.0 \\ 939.5 \end{bmatrix}, \quad \hat{\vec{\Sigma}}_u = \begin{bmatrix} 1017.5 \\ 1107.4 \\ 1127.8 \end{bmatrix}, \quad \vec{\Sigma}_u = \begin{bmatrix} 1111.1 \\ 1111.1 \\ 1111.1 \end{bmatrix}. \tag{58}$$

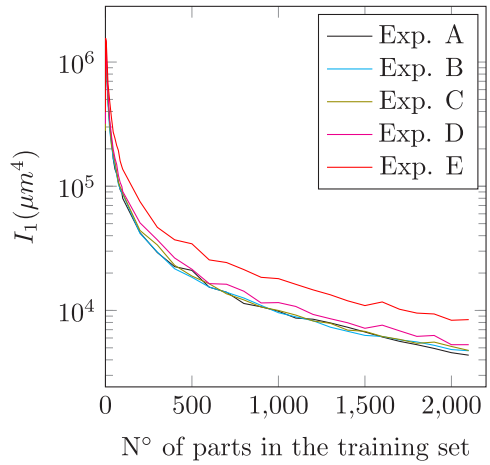
As it can be observed, most of the elements of $\hat{\Gamma}_B$ have been adjusted from Γ_0 towards the direction of Γ. Likewise, the elements of $\hat{\vec{\Sigma}}_u$ resemble the real values of $\vec{\Sigma}_u$.

6.4. Discussion

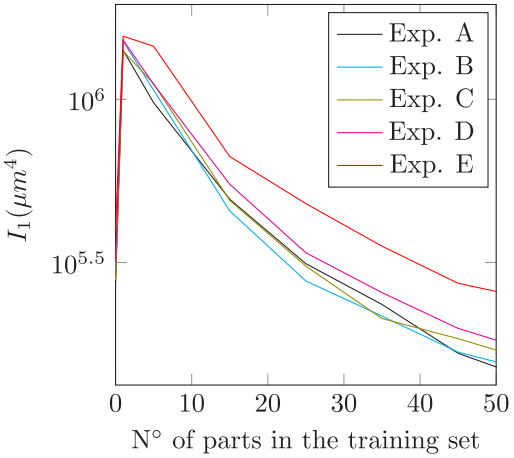
As it can be observed from the results in Figs. 4(a)–7, all indexes show a rapid decrease of the model error as the number of the data used for adjusting the model increases. As data increases, the estimation of the real Σ_y improves and the optimized parameters are closer to the real ones. For instance, given a training data set of 500 parts the adjusted model reduces the I_1 index from $10^{5.5}$ μm^4 to $2 \cdot 10^4$ μm^4, one order magnitude. Note that in these figures, the value of the index when data training is 0 refers to the index where the adjusted model is the one obtained from the SoV model and the engineering data, i.e., Γ_0.

However, index I_1 shows that when using only small data training sets the adjusted model could be even worst that the model without adjustment. In fact, when using up to 20–25 parts in the training data set, the index presents higher values than the initial value when no adjustment is made. This minimum number of parts in the training set required to enhance the results obtained through the initialization equations is quite similar to the number of decision variables that are calculated in this case study, as $\Gamma \in \mathbb{R}^{6\times3}$ and $\hat{\vec{\Sigma}}_u \in \mathbb{R}^{3\times1}$, thus the total amount of decision variables is 21. Therefore, a minimum set of data is needed to tune the model parameters and reduce the value of I_1, mainly because of the large amount of parameters to be adjusted. This behavior is not found in I_3 and I_4 since these indexes refer to the maximum parameter deviation from the real ones and index I_1 refers to an average error from all model parameters. Then, our methodology reduces the maximum parameter deviation of the model even when using small data sets although for tuning all parameters in the correct direction, a minimum data is required that depends on the number of the decision variables of the problem.

Another interesting result can be observed when comparing the model adjustment indexes with different measurement errors (noises). The evolution of the indexes is mainly drawn by the number of training data. However, the effect of the measurement noise, represented through the different experimental scenarios A, .., E, has a minimal impact. For instance, when 500 parts are used as training data, the adjusted model presents a I_1 value of $2 \cdot 10^4$ μm^4 and $3 \cdot 10^4$ μm^4 for a variance measurement error of 4 μm^2, and 256 μm^2, respectively. In both cases, the reduction of model error due to the adjustment is around one order magnitude with respect to no adjustment procedure. In fact, in order to obtain a similar model adjustment under a noisy environment, instead of using a data set of 500 parts, only 100 additional parts is needed, 600. Therefore, the methodology can be successfully applied



(a) Index I_1 w.r.t. number of parts used in the training set.



(b) Detail of the evolution of the index I_1 cases presented in Figure 4a.

Fig. 4. Index I_1 and detail of index I_1.

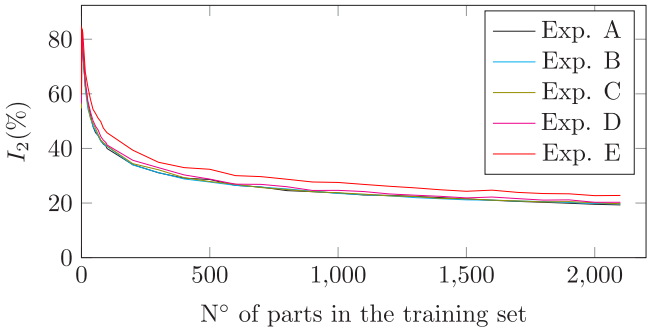


Fig. 5. Index I_2 w.r.t. number of parts used in the training set.

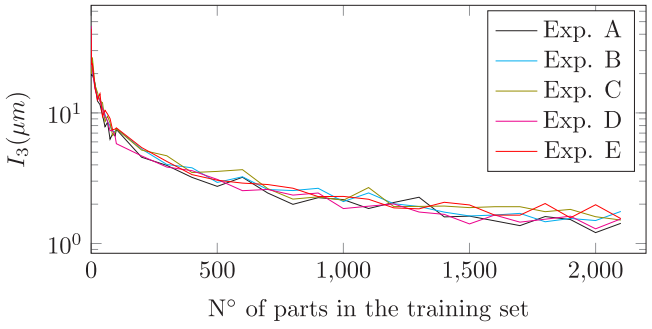


Fig. 6. Index I_3 w.r.t. number of parts used in the training set.

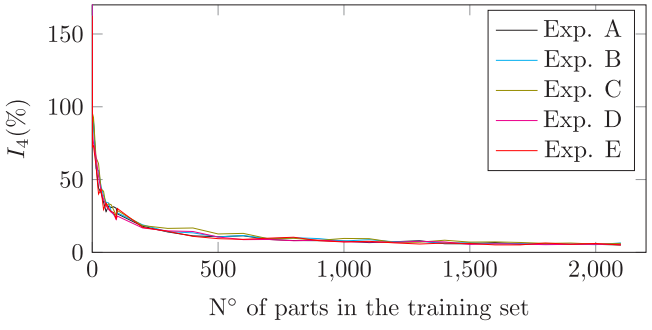


Fig. 7. Index I_4 w.r.t. number of parts used in the training set.

for model adjustment even when measurement data is obtained from noisy environments.

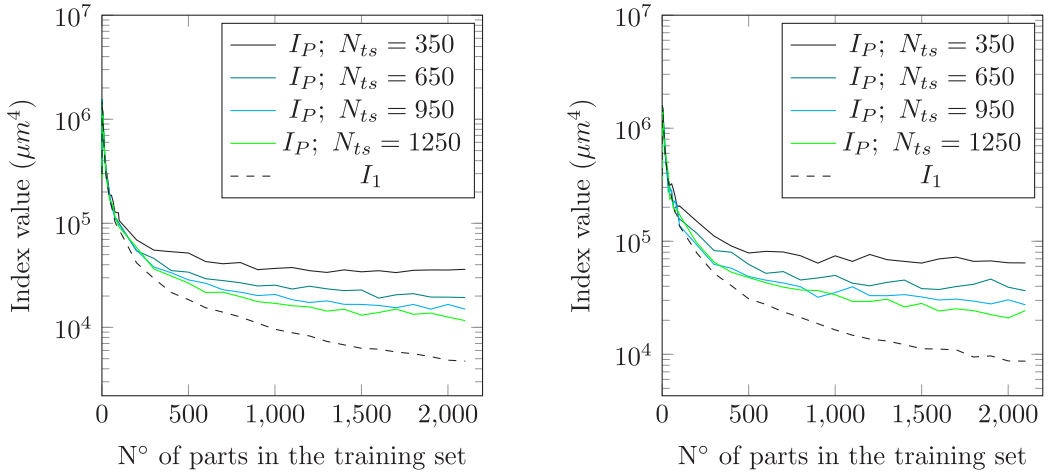
To reinforce this perception, it is presented Fig. 9 that plots the performance index I_1 with respect to the training data set and the measurement noise. We observe that performance index I_1 is inversely proportional to the number of parts used in the training set, similarly to other process estimators [40]. I_1 is also affected by the precision of the measuring instrument, as it is proportional to a polynomial function of the standard deviation of the measurement noise. Thus, the resulting surface can be fitted into the following form:

$$I_1 \sim \frac{b\sigma_v^2 + c\sigma_v + d}{N_{tr}}. \tag{59}$$

As stated above, practitioners may be interested in analyzing the performance of the adjusted model and thus, the I_P index should be applied. Figs. 8(a) and 8(b) show the evolution of the I_P indexes for different values of testing data sets in the experimental scenario B and E, respectively. Note that I_P analyzes the model error with respect to the sampling covariance of the testing data and not with respect to the real covariance of the data.

As it can be observed under both scenarios, it is necessary to use a notably high amount of testing parts in order to use I_P as a reliable substitute of I_1 in a practical case. That means that the index I_P may report a less model improvement that the reality, which can be seen by I_1. As observed in both figures, index I_P seems to be caught in a minimum value even if a higher data training set is available, and its use could lead the practitioner infer that the estimator is giving the problem predicted in (46), i.e., a similar (scaled) model is being obtained and the estimated model cannot be improved. However, index I_1 shows that the model is estimated with higher accuracy in those cases (with a larger data set).

Therefore, practitioners can use index I_P to estimate the error if the training data set is short, but must trust in the use of a larger data set to obtain better model estimation even if index I_P seems not to improve. Furthermore, if only a limited number of parts are available, it is recommendable to mostly use them to train the algorithm than to test it. For instance, we see in Fig. 8(a) that the use of 1000 parts for training leads to an error (in terms of I_1) of 10^4 μm^4, and the use of 2000 parts leads to $4{\cdot}10^3$ μm^4. However, the value that we can compute numerically in an experiment is index I_P that tells us that the error is $4{\cdot}10^4$ μm^4 with 1000 parts and $3.6{\cdot}10^4$ μm^4 (practically the same) with 2000 parts when using 350 parts for testing through I_P. This behavior of I_P is mainly explained by the use of a scarce set of parts for testing



(a) Comparison of I_P for several N_{ts} and I_1 for $\sigma_v^2 = 4\ \mu m^2$ (Exp. B).

(b) Comparison of I_P for several N_{ts} and I_1 for $\sigma_v^2 = 256\ \mu m^2$ (Exp. E).

Fig. 8. Comparison of I_P for different N_{ts} and I_1 for different measurement noises.

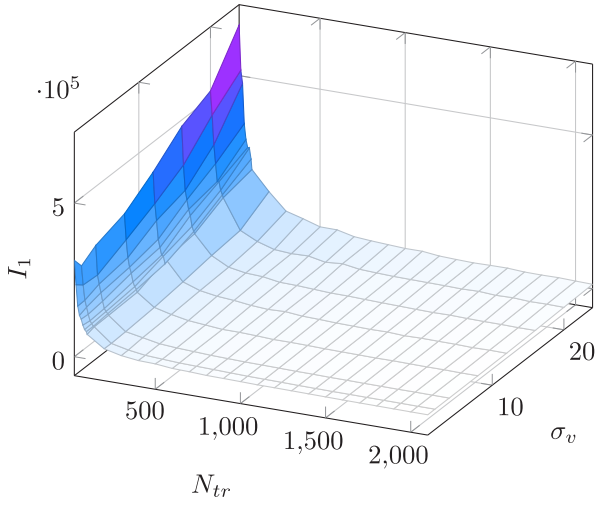


Fig. 9. Evolution of performance index I_1 for different number of processed parts in the training set and different standard deviations of the measurement noise.

that requires the estimation of the output covariance (Σ_y) through the sampled one ($\hat{\Sigma}_y$).

The analysis of the performance of the adjustment methodology with respect to the processed part shows that, for some given data size, we can execute our algorithm to obtain a more accurate estimate of model parameters. In that sense, our approach can be also applied for situations when some changes have occurred in the system due to small changes in the manufacturing process plan, fixture redesigns or other manufacturing readjustments. In those situations, one can run the optimization procedure when enough new data is gathered from the KPCs measurements of a set of processed parts, and then, the model is adapted to the new situation. With this, we can state that our algorithm allows to have some flexibility in adapting the model to new situations.

7. Conclusions

This paper presents a methodology to improve the accuracy of dimensional variation propagation models in MMPs by adjusting the model parameters using shop floor data. Variation propagation models based on engineering knowledge such as SoV models present limited accuracy due to linearization, approximation and modeling errors. This problem is aggravated as the number of stages increases due to error accumulation.

The initial mathematical development of the proposal requires solving a complex nonlinear polynomial problem that minimizes the difference between the KPCd measurements from a batch of processed parts and its estimation, which is a function of the estimated variation propagation matrix Γ and the estimated covariance of the variation sources. This optimization problem is initialized using physical models such as the SoV model and it is bounded using prior engineering knowledge and backup data. This problem can be hardly numerically faced with the available existing solvers, so we have used convexification techniques to obtain a tractable optimization problem by iterating through convex optimization problems that are solved in finite time and only require a few iterations.

After applying the proposed methodology to a simulated case study, we validated the adjusted model using several performance indexes. We conclude that the algorithm can be trained using a low amount of processed parts (at least the amount of decision variables of the optimization problem), and with any higher amount the estimation error of the covariance of the KPCd measurements is reduced proportionally to the amount of processed parts used during the training process. However, if only a sample covariance matrix of the KPCd measurements is available from a batch of processed parts (as it is common in practical cases), the amount of parts in the batch that are required to test the performance of the algorithm does increase notably.

The proposed methodology presents some limitations. First, it assumes that the theoretical covariance of the measurement noise remains identical as its respective sample covariance. This assumption may not hold but, since it is always assumed that their values will be notably lower than those of the KPCd measurements covariance, this assumption may not notably affect the accuracy of the model adjustment. Second, we consider that the magnitude of the errors in the model caused by approximations and other unmodeled linear errors that have arisen when obtaining the linear input–output variation propagation model of an MMP using physical models will be considerably higher than the linearizations performed in order to convexify the optimization problem and ensure that the optimization problem is solved within a finite time. Future research may include a greater refinement of the linearization methods for convexifying the optimization problem and analyzing the impact of the adjusting methodology in processes with higher dimensionality.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the grant ACIF/2018/245 from Generalitat Valenciana (Spain) and Fondo Social Europeo, and project UJI-B2020-33 from Universitat Jaume I (Spain).

Appendix A. Obtaining $\vec{\Sigma}_y$

Let us detail the behavior of the $diag(\cdot)$ operator as

$$diag\left(\begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}\right) = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix}. \tag{60}$$

Let B and C be two square matrices of the same size, the distributive property of the $diag(\cdot)$ operator allows

$$diag\,(B + C) = diag\,(B) + diag\,(C)\,. \tag{61}$$

According to [47], let $A \in \mathbb{R}^{n_a \times n}$ and $X \in \mathbb{R}^{n \times n}$ be some matrices, and X is a diagonal matrix. Then

$$diag\left(AXA^{\top}\right) = A^{\circ 2} diag(X). \tag{62}$$

Applying Eq. (62) into Eq. (13), and assuming that Σ_u is a diagonal matrix, then

$$\vec{\Sigma}_y = diag(\Sigma_y) = diag(\Gamma \Sigma_u \Gamma^{\top} + \Sigma_v) = diag(\Gamma \Sigma_u \Gamma^{\top}) + diag(\Sigma_v). \tag{63}$$

Substituting Eqs. (15), (17) and (18) into Eq. (63), we obtain Eq. (19).

Appendix B. Order reduction justification

B.1. The Schur complement

The order of the polynomial is reduced replacing the product of the objective function by an equivalent constraint. This conversion is implemented using the Schur complement.[1] We define a scalar variable t that bounds the maximum value of squared $e(\cdot)$. Considering that $e(\cdot)^{\top} \cdot e(\cdot)$ will always be zero or positive, we rewrite the optimization problem in Eq. (36a) into

$$(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \arg \min_{\hat{\Gamma}, \hat{\vec{\Sigma}}_u, t} \; t \tag{64a}$$

$$s.t.\; e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)^{\top} \cdot e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u) \le t. \tag{64b}$$

Constraint (64b) is rewritten into

$$t - e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)^{\top} \cdot I_{n_{S_y}} \cdot e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u) \ge 0, \tag{65}$$

as the identity matrix $I_{n_{S_y}}$ is always positive definite. Using the properties of the Schur Complement, we affirm that Eq. (65) will be positive semi-definite if and only if the following matrix H is also positive semi-definite.

$$H = \begin{bmatrix} t & e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)^{\top} \\ e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u) & I_{n_{S_y}} \end{bmatrix} \succeq 0. \tag{66}$$

Thus, both expressions are exchangeable without losing their inequality properties. Linear Matrix Inequality in Eq. (66) is applied in constraint (42b).

[1] $\begin{bmatrix} A & B \\ B^{\top} & C \end{bmatrix} \succeq 0$ if and only if $C \succ 0$ and $A - B\,C^{-1}\,B^{\top} \succeq 0$.

B.2. Linearization

The error function $e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)$ (38) is a 3rd order polynomial on the decision variables. We reduce its order using a first-order Taylor series approximation around an initial estimation of $\hat{\Gamma}$ and $\hat{\vec{\Sigma}}_u$ (40)–(41).

Assuming $vec(\hat{\Gamma}) = \left[\hat{\Gamma}[1,1]\ \hat{\Gamma}[2,1]\ \ldots\ \hat{\Gamma}[n_y, n_u]\right]^{\top}$ and $\hat{\vec{\Sigma}}_u = \left[\hat{\vec{\Sigma}}_u[1],\ \ldots,\ \hat{\vec{\Sigma}}_u[n_u]\right]^{\top}$, then

$$\frac{\partial e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)}{\partial vec(\hat{\Gamma})} = \left[\frac{\partial e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)}{\partial \hat{\Gamma}[1,1]} \quad \frac{\partial e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)}{\partial \hat{\Gamma}[2,1]} \quad \ldots \quad \frac{\partial e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)}{\partial \hat{\Gamma}[n_y, n_u]}\right], \tag{67}$$

$$\frac{\partial e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)}{\partial \hat{\vec{\Sigma}}_u} = \left[\frac{\partial e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)}{\partial \hat{\vec{\Sigma}}_u[1]} \quad \frac{\partial e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)}{\partial \hat{\vec{\Sigma}}_u[2]} \quad \ldots \quad \frac{\partial e(S_y, \hat{\Gamma}, \hat{\vec{\Sigma}}_u)}{\partial \hat{\vec{\Sigma}}_u[n_u]}\right]. \tag{68}$$

After that, the decision variables are obtained and $\hat{\Gamma}_{(j)}$ and $\hat{\vec{\Sigma}}_{u(j)}$ are updated into $\hat{\Gamma}_{(j+1)}$ and $\hat{\vec{\Sigma}}_{u(j+1)}$ using $\Delta\Gamma_{(j)}$ and $\Delta\hat{\Sigma}_{u(j)}$, as previously presented in Eq. (43).

References

[1] Powell Daryl, Magnanini Maria Chiara, Colledani Marcello, Myklebust Odd. Advancing zero defect manufacturing: A state-of-the-art perspective and future research directions. Comput Ind 2022;136:103596.
[2] Psarommatis Foivos, Sousa João, Mendonça João Pedro, Kiritsis Dimitris. Zero-defect manufacturing the approach for higher manufacturing sustainability in the era of industry 4.0: A position paper. Int J Prod Res 2022;60(1):73–91.
[3] Caiazzo Bianca, Di Nardo Mario, Murino Teresa, Petrillo Alberto, Piccirillo Gianluca, Santini Stefania. Towards Zero Defect Manufacturing paradigm: A review of the state-of-the-art methods and open challenges. Comput Ind 2022;134:103548.
[4] Psarommatis Foivos, May Gökan, Dreyfus Paul-Arthur, Kiritsis Dimitris. Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research. Int J Prod Res 2020;58(1):1–17.
[5] May Gökan, Kiritsis Dimitris. Zero defect manufacturing strategies and platform for smart factories of industry 4.0. In: International conference on the industry 4.0 model for advanced manufacturing. Springer; 2019, p. 142–52.
[6] Shi Jianjun. Stream of variation modeling and analysis for multistage manufacturing processes. CRC Press; 2006.
[7] Zhou Shiyu, Huang Qiang, Shi Jianjun. State space modeling of dimensional variation propagation in multistage machining process using differential motion vectors. IEEE Trans Robot Autom 2003;19(2):296–309.
[8] Kim Hyun-Jin, Kim Kwang-Jae, Kwak Doh-Soon. A case study on modeling and optimizing photolithography stage of semiconductor fabrication process. Qual Reliab Eng Int 2010;26(7):765–74.
[9] Abellán-Nebot José V. Derivation and application of the stream of variation model to the manufacture of ceramic floor tiles. Qual Eng 2018;30(4):713–29.
[10] Jin Jionghua, Shi Jianjun. State space modeling of sheet metal assembly for dimensional control. J Manuf Sci Eng 1999;121(4):756–62.
[11] Camelio Jaime, Hu S Jack, Ceglarek Dariusz. Modeling variation propagation of multi-station assembly systems with compliant parts. J Mech Des 2003;125(4):673–81.
[12] Liu Jian, Jin Jionghua, Shi Jianjun. State space modeling for 3-D variation propagation in rigid-body multistage assembly processes. IEEE Trans Autom Sci Eng 2009;7(2):274–90.
[13] Zhang Tingyu, Shi Jianjun. Stream of variation modeling and analysis for compliant composite part assembly—Part I: Single-station processes. J Manuf Sci Eng 2016;138(12).
[14] Zhang Tingyu, Shi Jianjun. Stream of variation modeling and analysis for compliant composite part assembly—Part II: Multistation processes. J Manuf Sci Eng 2016;138(12).
[15] Abellan-Nebot José V, Liu Jian, Subirón Fernando Romero, Shi Jianjun. State space modeling of variation propagation in multistation machining processes considering machining-induced variations. J Manuf Sci Eng 2012;134(2).
[16] Moliner-Heredia Rubén, Abellán-Nebot José Vicente, Peñarrocha-Alós Ignacio. Extension of the stream-of-variation model for general-purpose workholding devices: Vices and three-jaw chucks. IEEE Trans Autom Sci Eng 2021.
[17] Loose Jean-Philippe, Zhou Shiyu, Ceglarek Dariusz. Kinematic analysis of dimensional variation propagation for multistage machining processes with general fixture layouts. IEEE Trans Autom Sci Eng 2007;4(2):141–52.
[18] Izquierdo L Eduardo, Shi Jianjun, Hu S Jack, Wampler Charles W. Feedforward control of multistage assembly processes using programmable tooling. Trans NAMRI/SME 2007;35:295–302.
[19] Abellan-Nebot José V, Liu Jian, Subirón F Romero. Quality prediction and compensation in multi-station machining processes using sensor-based fixtures. Robot Comput-Integr Manuf 2012;28(2):208–19.

[20] Zhu Cheng, Chang Qing, Arinez Jorge. Data-enabled modeling and analysis of multistage manufacturing systems with quality rework loops. J Manuf Syst 2020;56:573–84.
[21] Abellán-Nebot José V, Penarrocha Ignacio, Sales-Setién Ester, Liu Jian. Optimal inspection/actuator placement for robust dimensional compensation in multistage manufacturing processes. In: Computational methods and production engineering. Elsevier; 2017, p. 31–50.
[22] Ding Yu, Shi Jianjun, Ceglarek Dariusz. Diagnosability analysis of multi-station manufacturing processes. J Dyn Syst, Meas, Control 2002;124(1):1–13.
[23] Jiao Yibo, Djurdjanovic Dragan. Compensability of errors in product quality in multistage manufacturing processes. J Manuf Syst 2011;30(4):204–13.
[24] Bradley William, Kim Jinhyeun, Kilwein Zachary, Blakely Logan, Eydenberg Michael, Jalvin Jordan, Laird Carl, Boukouvala Fani. Perspectives on the integration between first-principles and data-driven modeling. Comput Chem Eng 2022;107898.
[25] Wang Jinjiang, Li Yilin, Gao Robert X, Zhang Fengli. Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. J Manuf Syst 2022;63:381–91.
[26] Sansana Joel, Joswiak Mark N, Castillo Ivan, Wang Zhenyu, Rendall Ricardo, Chiang Leo H, Reis Marco S. Recent trends on hybrid modeling for Industry 4.0. Comput Chem Eng 2021;151:107365.
[27] Von Stosch Moritz, Oliveira Rui, Peres Joana, de Azevedo Sebastião Feyo. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. Comput Chem Eng 2014;60:86–101.
[28] McFarland John, Mahadevan Sankaran, Romero Vicente, Swiler Laura. Calibration and uncertainty analysis for computer simulations with multivariate output. AIAA J 2008;46(5):1253–65.
[29] Kennedy Marc C, O'Hagan Anthony. Bayesian calibration of computer models. J R Stat Soc Ser B Stat Methodol 2001;63(3):425–64.
[30] Joseph V Roshan, Melkote Shreyes N. Statistical adjustments to engineering models. J Qual Technol 2009;41(4):362–75.
[31] Plumlee Matthew. Bayesian calibration of inexact computer models. J Amer Statist Assoc 2017;112(519):1274–85.
[32] Wang Yan, Tuo Rui. Semi-parametric adjustment to computer models. Statistics 2020;54(6):1255–75.
[33] Arendt Paul D, Apley Daniel W, Chen Wei. Quantification of model uncertainty: Calibration, model discrepancy, and identifiability. J Mech Des (1990) 2012;134(10).
[34] Joseph V Roshan, Yan Huan. Engineering-driven statistical adjustment and calibration. Technometrics 2015;57(2):257–67.
[35] Bayarri Maria J, Berger James O, Paulo Rui, Sacks Jerry, Cafeo John A, Cavendish James, Lin Chin-Hsu, Tu Jian. A framework for validation of computer models. Technometrics 2007;49(2):138–54.
[36] Arendt Paul D, Apley Daniel W, Chen Wei, Lamb David, Gorsich David. Improving identifiability in model calibration using multiple responses. J Mech Des (1990) 2012;134(10).
[37] Wang Yan, Yue Xiaowei, Tuo Rui, Hunt Jeffrey H, Shi Jianjun. Effective model calibration via sensible variable identification and adjustment with application to composite fuselage simulation. Ann Appl Stat 2020;14(4):1759–76.
[38] Liu Bingjie, Yue Xubo, Byon Eunshin, Al Kontar Raed. Parameter calibration in wake effect simulation model with stochastic gradient descent and stratified sampling. Ann Appl Stat 2022;16(3):1795–821.
[39] Sun Hao, Zhao Shengqiang, Zhang Teng, Peng Fangyu, Zhou Lin, Yan Rong. Analysis and inference of stream of dimensional errors in multistage machining process based on an improved semiparametric model. In: 2022 IEEE/ASME International conference on advanced intelligent mechatronics (AIM). IEEE; 2022, p. 996–1003.
[40] Ding Yu, Zhou Shiyu, Chen Yong. A comparison of process variation estimators for in-process dimensional measurements and control. J Dyn Syst Meas Control 2005;127:69.
[41] Sales-Setién Ester, Penarrocha-Alós Ignacio, Abellán-Nebot José V. Estimation of nonstationary process variance in multistage manufacturing processes using a model-based observer. IEEE Trans Autom Sci Eng 2018;16(2):741–54.
[42] Lasserre Jean B. Global optimization with polynomials and the problem of moments. SIAM J Optim 2001;11(3):796–817.
[43] Henrion Didier, Lasserre Jean-Bernard, Löfberg Johan. GloptiPoly 3: moments, optimization and semidefinite programming. Optim Methods Softw 2009;24(4–5):761–79.
[44] Lasserre Jean B. A semidefinite programming approach to the generalized problem of moments. Math Program 2008;112(1):65–92.
[45] Apley Daniel W, Shi Jianjun. Diagnosis of multiple fixture faults in panel assembly. J Manuf Sci Eng 1998;120(4):793–801.
[46] Löfberg J. YALMIP : A toolbox for modeling and optimization in MATLAB. In: Proceedings of the CACSD conference. Taipei, Taiwan; 2004.
[47] Searle Shayle R, Khuri Andre I. Matrix algebra useful for statistics. John Wiley & Sons; 2017.

Rubén Moliner-Heredia was born in Vila-real (Castelló), Spain in 1994. He received the M.Sc. degree in Industrial Engineering from the Universitat Jaume I de Castelló in 2018, and he is currently pursuing a Ph.D. degree in the Industrial Technologies and Materials Programme at Universitat Jaume I. His research interests are fault diagnosis and Stream-of-Variation in manufacturing systems.

Ignacio Peñarrocha-Alós was born in Castelló, Spain in 1978. He received the M.Sc. degree in Industrial Engineering from the Universitat Jaume I de Castelló in 2002, and his Ph.D. in Computering and Control Engineering from Universitat Politècnica de València (UPV) in 2006. He has been working since 2004 at the Universitat Jaume I de Castelló. His current position is as an Associate Professor at the Department of Industrial Engineering and Design. He has participated in several local and national research projects. His research interests include identification, estimation, fault diagnosis and control over networks, and fault tolerant control of wind energy systems.

José V. Abellán-Nebot is an Associate Professor in the Department of Industrial Engineering and Design at Universitat Jaume I de Castelló, Spain. He received his M.Sc. in Industrial Engineering and his Ph.D. in Technological Innovation Projects in Product and Process Engineering from the Universitat Jaume I de Castelló in 2003 and 2011, respectively. He was a Visiting Scholar at Monterrey Institute of Technology (2005), University of Michigan (2007), University of Arizona (2009 and 2012) and Georgia Institute of Technology (2014). His research interests focus on product quality due to the Stream-of-Variation in manufacturing systems and intelligent machining systems.