
Doctoral Programme in Industrial Technologies and Materials
Doctoral School of the Universitat Jaume I

# Modeling, fault diagnosis and prognosis under the Zero Defect Manufacturing paradigm

*A dissertation submitted by Rubén Moliner Heredia to obtain the degree of Doctor of Philosophy from the Universitat Jaume I*

AUTHOR
Rubén Moliner Heredia

ADVISORS
Dr. Ignacio Peñarrocha Alós
Dr. José Vicente Abellán Nebot

Castellón de la Plana, January 2023

Programa de Doctorado en Tecnologías Industriales y Materiales
Escuela de Doctorado de la Universitat Jaume I

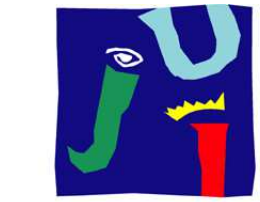
# Modelado, diagnóstico de fallos y prognosis bajo el paradigma de la Fabricación Cero Defectos

*Memoria presentada por Rubén Moliner Heredia para optar al grado de doctor por la Universitat Jaume I*

AUTOR
Rubén Moliner Heredia

DIRECTORES
Dr. Ignacio Peñarrocha Alós
Dr. José Vicente Abellán Nebot

Castellón de la Plana, Enero 2023



Programa de Doctorat en Tecnologies Industrials i Materials
Escola de Doctorat de la Universitat Jaume I

# Modelització, diagnòstic de fallades i prognosi sota el paradigma de la Fabricació Zero Defectes

*Memòria presentada per Rubén Moliner Heredia per a optar al grau de doctor per la Universitat Jaume I*


AUTOR
Rubén Moliner Heredia

DIRECTORS
Dr. Ignacio Peñarrocha Alós
Dr. José Vicente Abellán Nebot


Castelló de la Plana, Gener 2023

# Funding | Financiación


### Contrato predoctoral

- Ayudas predoctorales para la formación de personal investigador, financiadas por la Universitat Jaume I. Referencia PREDOC/2017/56.
- Subvención para la contratación de personal investigador de carácter predoctoral (ACIF2018), financiada por la Generalitat Valenciana y por el Fondo Social Europeo.
  Referencia ACIF/2018/245.

### Proyectos de investigación

- Desarrollo de modelos portables y adaptables en operaciones de mecanizado: uso y optimización bajo condiciones de sostenibilidad. Código: P1-1B2015-53. Entidad financiadora: Universitat Jaume I.
- Modelado e implementación del gemelo digital para sistemas ciber-físicos de producción (CPPS) orientado al aseguramiento de la calidad geométrica de producto. Código: UJI-B2020-33. Entidad financiadora: Universitat Jaume I.
- Adaptación de algoritmos de control PID al contexto de la Industria 4.0. Código: UJI-B2018-39. Entidad financiadora: Universitat Jaume I.
- Estrategias de estimación y control para la minimización del coste energético en procesos con perturbaciones variantes y sujetos a periodos tarifarios. Código: TEC2015-69155-R. Entidad financiadora: Ministerio de Economía y Competitividad.


*To my grandparents*

# Abstract

The Fourth Industrial Revolution is underway thanks to the latest technological advances, such as cyber-physical systems, cloud computing, big data or data mining. Industries have to adapt to the new production paradigms, which require the minimization of errors and the maximization of quality in complex production systems. One of the most promising quality management strategies is Zero Defect Manufacturing (ZDM), which seeks to guarantee that no defective products reach the customer, leading to increased product quality and sustainability and minimizing costs and waste. ZDM is based on four interrelated strategies: defect detection, prediction, reparation and prevention.

This thesis addresses several problems that may arise during the development of these strategies. We have developed variation propagation models and proposed methodologies to adjust these models using process data. We have also developed fault diagnosis algorithms to minimize the measurement cost using process planning-based models, and prognosis adaptive algorithms to indirectly estimate the remaining useful life of a cutting tool.

First, in this thesis we have proposed variation propagation models for vices and self-centering three-jaw chucks, which are fixtures that are traditionally used in industry, using the Stream-of-Variation methodology. These models detail the effect of the deviations of the locating components of each fixture and the deviations of the locating datums of the workpiece on the newly generated features. In this thesis, we also propose a methodology to adjust linear input-output models, given that physical-based models may contain unknown or inaccurate terms. This methodology uses process data and engineering knowledge to perform the adjustment. The methodology consists in the minimization of the prediction errors using the covariance matrices of the key dimensional characteristics of the product.

We have proposed a procedure to detect and isolate faults online in a process with the objective of reducing the necessary amount of required measurements and thus, the total measurement cost. This procedure uses a model obtained using the planning process, and the fault isolation criteria applied in the procedure algorithm are based on the information gain index of each measurable variable.

This thesis also addresses the problem of indirectly monitoring and predicting the condition and the remaining useful life of cutting tools during the machining process. First, we have proposed a method to estimate the surface roughness of the processed parts during the periods where those measurements may be unavailable, using empirical models, power consumption

measurements and an observer with a steady-state Kalman filter. Additionally, we have proposed a prognosis methodology to predict the future values of the power consumption (whose measurements have a large amount of measurement noise) and the surface roughness of the processed parts (which are scarce measurements) using an adaptive recursive least squares algorithm. This algorithm can learn the behavior of the predicted variables with a low amount of measurements, and can adapt to sudden changes in the cutting conditions without being affected by minor disturbances. This algorithm can estimate the remaining useful life of the cutting tool and can be used to avoid early tool replacements or processing parts out of specifications.

# Resumen

La Cuarta Revolución Industrial está en progreso gracias a los últimos avances tecnológicos, como los sistemas ciberfísicos, la computación en la nube, el *big data* o la minería de datos. Las industrias tienen que adaptarse a los nuevos paradigmas de producción, que requieren la minimización de los errores y la maximización de la calidad en sistemas de producción complejos. Una de las estrategias de gestión de la calidad más prometedoras es la Fabricación Cero Defectos (*Zero Defect Manufacturing (ZDM)*, en inglés), que busca garantizar que ningún producto con defectos llegue al cliente, lo que conlleva un aumento de la calidad de los productos producidos y de la sostenibilidad y una minimización del coste y de los desperdicios. ZDM está basado en cuatro estrategias interrelacionadas: detección, predicción, reparación y prevención de los defectos.

Esta tesis aborda varios de los problemas que pueden aparecer durante el desarrollo de estas estrategias. Hemos desarrollado modelos de propagación de la variación y propuesto metodologías para ajustar dichos modelos usando datos del proceso. También hemos desarrollado algoritmos de diagnóstico de fallos para minimizar el coste de tomar medidas utilizando modelos basados en la planificación de procesos, y algoritmos adaptativos de prognosis para estimar indirectamente el tiempo restante de vida útil de una herramienta de corte.

Primero, en esta tesis hemos propuesto modelos de propagación de la variación para mordazas y platos de tres garras autocentrantes, que son utillajes utilizados tradicionalmente en la industria, utilizando la metodología del *Stream-of-Variation*. Estos modelos detallan el efecto de las desviaciones de los componentes localizadores de cada utillaje y de las desviaciones de los *datums* localizadores de la pieza en los nuevos *features* generados. También proponemos en esta tesis una metodología para ajustar modelos lineales de entrada-salida, dado que los modelos físicos pueden contener términos desconocidos o imprecisos. Esta metodología usa datos del proceso y conocimiento ingenieril para realizar el ajuste. La metodología consiste en la minimización de los errores de predicción usando las matrices de covarianza de las características dimensionales clave del producto.

Hemos propuesto un procedimiento para detectar y aislar fallos online en un proceso con el objetivo de reducir la cantidad necesaria de toma de medidas y, por tanto, el coste total de tomar las medidas. Este procedimiento utiliza un modelo basado en la planificación de procesos, y los criterios de aislamiento de fallos utilizados en el algoritmo están basados en el índice de ganancia de información de cada variable que puede medirse.

Esta tesis también aborda el problema de la monitorización indirecta y la predicción del estado y de la vida útil restante de las herramientas de corte durante los procesos de mecanizado. Primero, hemos propuesto un método para estimar la rugosidad superficial de las partes procesadas durante los periodos en los que esas medidas pueden no estar disponibles, usando modelos empíricos, medidas de la potencia consumida y un observador con un filtro de Kalman en estado estacionario. Además, hemos propuesto una metodología de prognosis para predecir los valores futuros de la potencia consumida (cuyas medidas tienen mucho ruido) y de la rugosidad superficial de las partes producidas (cuyas medidas son escasas) usando un algoritmo RLS (*Recursive-Least-Squares*, en inglés) adaptativo. Este algoritmo puede aprender el comportamiento de las variables que se predicen usando una cantidad de medidas reducida, y puede adaptarse a cambios súbitos en las condiciones de corte sin verse afectado por perturbaciones menores. Este algoritmo puede estimar la vida útil restante de la herramienta de corte y puede usarse para evitar el reemplazo prematuro de las herramientas o producir partes que estén ya fuera de especificaciones.

# Resum

La Quarta Revolució Industrial es troba en progrés gràcies als últims avanços tecnològics, com els sistemes ciberfísics, la computació en el núvol, el big data o la mineria de dades. Les indústries han d'adaptar-se als nous paradigmes de producció, que requereixen la minimització d'errors i la maximització de la qualitat en sistemes de producció complexos. Una de les estratègies de gestió de la qualitat més prometedores és la Fabricació Zero Defectes (*Zero Defect Manufacturing (ZDM)*, en anglès), que busca garantir que cap producte amb defectes arribe al client, fet que comporta un augment de la qualitat i la sostenibilitat i una minimització del cost i dels materials rebutjats. ZDM està basat en quatre estratègies interrelacionades: detecció, predicció, reparació i prevenció de defectes.

Aquesta tesi tracta diversos problemes que poden sorgir durant el desenvolupament d'aquestes estratègies. Hem desenvolupat models de propagació de la variació i proposat metodologies per ajustar aquests models utilitzant dades del procés. També hem desenvolupat algorismes de diagnòstic de fallades per minimitzar el cost de prendre mesures utilitzant models basats en la planificació de processos, i algorismes adaptatius de prognosi per estimar indirectament el temps restant de vida útil d'una eina de tall.

Primer, en aquesta tesi hem proposat models de propagació de la variació per a mordasses i plats de tres urpes amb autocentrament, que són utillatges utilitzats tradicionalment en la indústria, utilitzant la metodologia del *Stream of Variation*. Aquests models detallen l'efecte de les desviacions dels components localitzadors de cada utillatge i de les desviacions dels *datums* localitzadors de la peça en els nous *features* generats. També proposem en aquesta tesi una metodologia per ajustar models lineals d'entrada-eixida, donat que els models físics poden contenir termes desconeguts o imprecisos. Aquesta metodologia utilitza dades del procés i coneixements d'enginyeria per realitzar l'ajust. La metodologia consisteix en la minimització dels errors de predicció utilitzant les matrius de covariància de les característiques dimensionals clau del producte.

Hem proposat un procediment per a detectar i aïllar fallades online en un procés amb l'objectiu de reduir la quantitat necessària de mesuraments i, per tant, el cost total de realitzar els mesuraments. Aquest procediment utilitza un model basat en la planificació de processos, i els criteris d'aïllament de fallades utilitzats en l'algorisme estan basats en l'índex de guany d'informació de cada variable que es pot mesurar.

Aquesta tesi també tracta el problema de la monitorització indirecta i la predicció de l'estat i la vida útil restant de les eines de tall durant els processos de mecanització. Primer, hem proposat

un mètode per estimar la rugositat superficial de les parts processades durant els períodes en que aquestes mesures poden no estar disponibles, utilitzant models empírics, mesures de la potència consumida i un observador amb un filtre de Kalman en estat estacionari. A més, hem proposat una metodologia de prognosi per predir els valors futurs de la potència consumida (amb mesures amb molt de soroll) i de la rugositat superficial (amb mesures escasses) utilitzant un algorisme RLS (*Recursive-Least-Squares*, en anglès) adaptatiu. Aquest algorisme pot aprendre el comportament de les variables que es prediuen utilitzant una quantitat reduïda de mesures, i pot adaptar-se a canvis sobtats en les condicions de tall sense veure's afectat per pertorbacions menors. Aquest algorisme pot estimar la vida útil restant de la eina de tall i pot utilitzar-se per evitar el reemplaçament prematur de les eines o produir parts fora d'especificacions.

# Acknowledgements | Agradecimientos

A mis padres, José Miguel y Estrella, y a mi hermano, Daniel, por toda su comprensión y por todo el apoyo incondicional que me han dado, tanto durante el trascurso del doctorado como durante toda mi carrera en la universidad.

A mis directores, Ignacio Peñarrocha y José Vicente Abellán, por toda la ayuda y experiencia que me habéis proporcionado y, sobre todo, la confianza que habéis tenido en mí. Gracias a *Jovi*, por haberme acogido en el área de Ingeniería de Procesos y haberme hecho sentir como en casa. Y a *Nacho*, por haberme mentorizado desde que comencé el grado de Ingeniería en Tecnologías Industriales, y sobre todo, por haberme apoyado siempre, en todas las circunstancias.

A mis compañeros de laboratorio, Carlos, Oscar, Ester y David, con los que he compartido los mejores y los más peculiares momentos durante nuestras investigaciones. Quiero mencionar a todos los compañeros de cantina que también han hecho y siguen haciendo el mismo camino que he hecho yo. Y también a los demás compañeros del departamento y de la universidad, por toda la ayuda que me han prestado.

# Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction

## 1.1 Motivation

### 1.1.1 Industry 4.0 and Zero Defect Manufacturing

The Fourth Industrial Revolution, also known as Industry 4.0, is under way as the result of a production paradigm shift towards Mass Personalization Production (MPP). In this new paradigm, customers are highly involved in the manufacturing process, as production is personalized for each individual customer, to the extent that even basic design and product structure can be adapted to meet each individual's requirements. The development of Industry 4.0 and this new paradigm has been possible thanks to the latest technological advances, such as cyber-physical systems, cloud computing, Internet of things, big data and data mining [1].

However, Mass Personalization Production presents several new challenges to the manufacturing industry, as products are expected to be more customizable over time, produced at faster rates in smaller batches, which leads to shorter life cycles of high complex modern manufacturing processes, and thus leading to an increased probability of the occurrence of production errors. Thus, new quality management strategies are required in order to deal with these new challenges [2].

One of the most promising strategies is Zero Defect Manufacturing (ZDM). Psarommatis et al [3] have proposed the following definition:

*'ZDM is 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.'*

In fact, ZDM assumes that defects are an intrinsic part of any manufacturing process, and aims at its early detection in order to adjust the process, in such a way that its impact on the final quality of the product is minimized. ZDM also aims to minimize the amount of waste

generated during the manufacturing process (*'zero waste'*) and the reduction of the use of resources, thus leading to a more sustainable production [4].

The concept of Zero Defects in manufacturing had already appeared in the 60s [4], as part of the US Army Pershing Missile Program, and in Toyota Motor Company, Japan, as a vision to stimulate quality control activities across the organization. This zero defect vision motivated subsequent methods such as Taguchi methods, Six Sigma or Lean Production [2]. However, the concept of ZDM as it is known present-day began to gain traction in the latest decade [2, 4]. Using the aforementioned latest technologies and Industry 4.0, ZDM aims to fill the gaps from existing quality control methods by learning from defects [5].

Zero Defect Manufacturing has also been encouraged via state-funded projects. In the case of the European Commission, it has funded projects since 2011, such as InterQ, ZDMP, QU4LITY, among many others, for a grand total budge of over 156 million euros since then [3].

### 1.1.2 Implementing Zero Defect Manufacturing

Zero Defect Manufacturing can be implemented with a product-oriented approach or with a process-oriented approach. The product-oriented approach focuses on analyzing and reducing the defects of the processed parts, while the process-oriented approach focuses on studying and eliminating the defects of the manufacturing process. Both approaches are intertwined and part of the same concept [6], as product defects can be inferred from process defects, and vice versa. Figure 1.1 illustrates this concept.

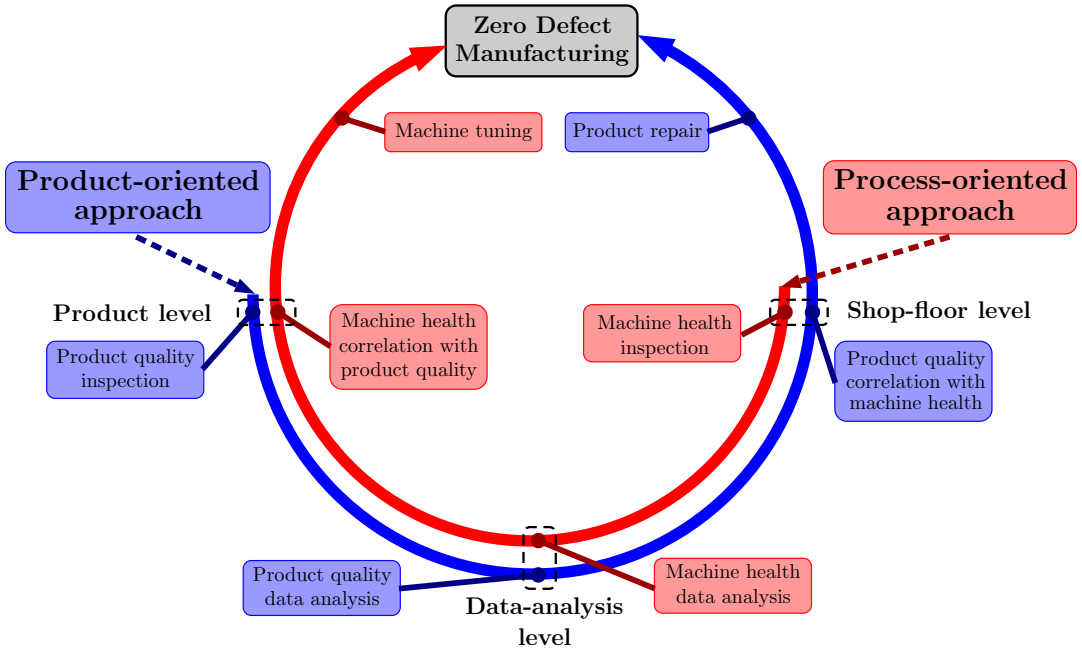



**Figure 1.1.** Relationship between the product-oriented and the process-oriented approaches to ZDM. Adapted from [6].

Zero Defect Manufacturing is comprised by four strategies, known as Detect, Predict, Repair and Prevent, which are interconnected as the following trigger-action pairs: Detect - Repair, Detect - Prevent and Predict - Prevent [7]. These relations are shown in Figure 1.2. These strategies and relations parallel the intelligent fault diagnosis and prognosis systems (IFDAPS) framework for ZDM, which consists of a control loop that collects and processes data, and applies fault diagnosis and prognosis to propose defect corrections and compensations on the process [8].

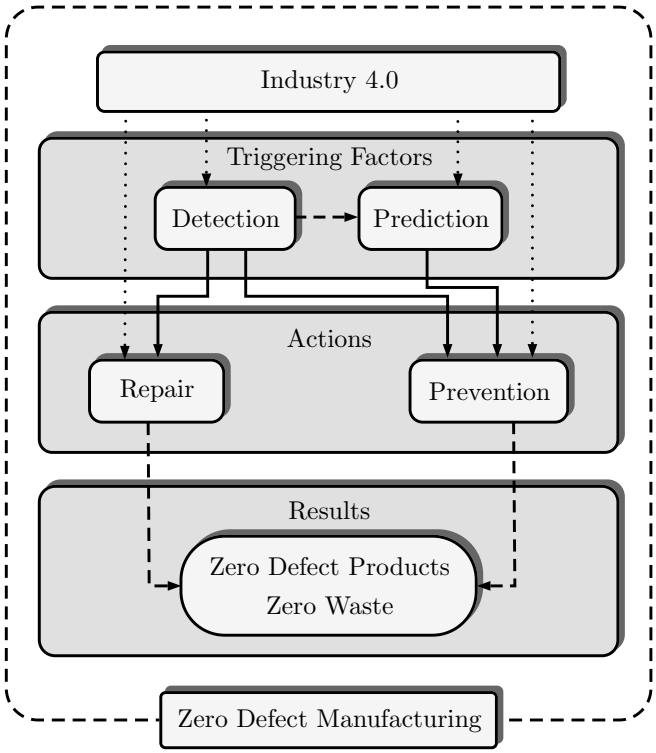



**Figure 1.2.** Implementation of the strategies within Zero Defect Manufacturing. Adapted from [6].

This thesis aims to propose solutions to problems within the Detection - Repair and Prediction - Prevention pairs, thus, we will focus on these from now on.

### Detect - Repair

In order to detect any defects on the manufacturing system, it is important to monitor the state of the processed product during the manufacturing process, via direct or indirect means and using online or offline techniques, depending on the features of each type of measurement or process. Alternatively, the state of the process can also be monitored indirectly using additional outputs from the manufacturing process (such as power consumption or vibrations).

Together with the monitoring process, it is essential to develop fault diagnosis methods, in order to detect, isolate and identify in real-time any faults and failures that may appear during the manufacturing process.

These methods require reliable models of the individual components of the product and the process, as well as models of the general behavior of the process. These models can be obtained through physical, empirical or data mining methods. The models are used to analyze and trace the transmission of the defects from the process to the final product (e.g. [9,10]) and determine the process component deterioration (e.g. [11,12]). The precision and accuracy of the used models will directly affect the performance of the fault diagnosis methods.

Once a defect in the product has been detected, it is necessary to make certain decisions in order to repair the product before it leaves the manufacturing process. These decisions may include modifying the following operations downstream or, in multistage processes, applying rework loops [13] by recirculating the product to additional and/or previous stages.

Reparation methods require accurate knowledge of the process behavior, as it is essential to analyze its diagnosability [14] and compensability [15], in order to decide if a defect can be repaired and where it can be repaired.

The pair Detect - Repair is further related as monitorization requires using sensors to measure the state of the process and the product, which require to be located in order to ensure that defects can be repaired; however, due to the specifications of each manufacturing process, some problems may arise; for example, some measurements may require the destruction of the product, or the measurement entails an unacceptable economical cost. Thus, it is clear that a holistic approach to the pair is required in order to achieve ZDM.

#### Predict - Prevent

The evolution of the state of the components in manufacturing process, its effect in the final quality of the product and the apparition of defects in the processed parts can be predicted. This prediction requires reliable knowledge of the process, with accurate models that can be analyzed accordingly. It also requires adequately-treated data from the process, obtained through the aforementioned monitoring system, in order to update the models and predict the moment of failure.

Prevention actions are then triggered by the prediction of an upcoming defect in the process. These prediction actions may consist of direct physical actions or modifications of the internal algorithm that controls the manufacturing process, in order to ensure that any products leave the process within the quality standards.

A frequent example of the pair Predict - Prevent can be found in remaining useful life (RUL) methods [16], which aim to predict the remaining life of a part of a process (such as machining tools or fixtures) and develop algorithms to replace them preventively, before the product quality is affected, but also reducing cost and waste.

## 1.2 Challenging problems

This thesis deals with several challenges comprised within the Zero Defect Manufacturing strategies, focusing on developing and adjusting product and process models, fault detection and prognosis with low measurement availability.

In the following section, we will briefly describe some of the major challenging problems that we have covered along the thesis.

### 1.2.1 Modeling and adjustment

The core motivation of ZDM is to guarantee that no defective products leave the factory. In order to achieve these specifications, it is necessary to monitor the state of the process and the product, detect any defects and faults that may appear and predict the future state of the system, so actions can be taken active or preventively. Therefore, in order to perform these actions successfully, it is essential to obtain beforehand reliable models of the process, the manufactured product and the relationship between process and product that explains defect propagation in a manufacturing process.

Depending on the characteristics of each process and product, these models can be obtained through theoretical methods (such as physical, geometrical or engineering-based means), empirical methods (by performing experiments focused on relating certain variables), data-based methods (such as pattern detection and neural networks) or by combinations of these methods.

However, we have found that, given the sheer variety of processes and product with their respective characteristics that can be found in the manufacturing industry, there are still many methods and combinations that have not been developed or tested yet. Thus, we define the first problem as follows.

**P1.** *To design, develop and test methods and combinations of methods in order to obtain reliable models of processes, products and the propagation of defects and faults along the manufacturing process.*

Many of the defect propagation models that have already been developed and obtained are those from advanced factories, which can modify their manufacturing stages using customized fixtures that behave similarly to 3-2-1 (physical-based) models. In order to bring closer Zero Defect Manufacturing to more traditional industries, and continuing with problem **P1**, it is interesting to obtain models of classical fixtures, which do not necessarily behave like 3-2-1 models. The second problem is defined as:

**P2.** *To develop, obtain and validate physical-based defect propagation models of fixtures that are used in traditional industries.*

Some physical-based models, such as those developed within the Stream-of-Variation (SoV) approach, assume that defect propagation is a linear function of the previous states of the product and the sources of variation from the process. Thus, these relationships are linearized.

Given that the dimensions of the defects are notably smaller than the dimensions of the products, linearizations and simplifications do not cause noticeable errors when these models are used to estimate the defect propagation in single stages. However, these errors might accumulate

in multistage manufacturing process with a high amount of stages. Additionally, there are cases where some propagation processes are not well known or cannot be generically modeled.

In all the previous cases, the resulting defect propagation model may present inaccuracies that can render them unfit for their use in detection and prediction algorithms. Thus, it would be useful to combine these models with output data from the processed products, thus adjusting the model to fit the real behavior of the manufacturing process. The third problem is then defined below.

**P3.** *To develop a methodology to adjust a linear model by combining physical and data-driven methods, and using additional engineering knowledge.*

### 1.2.2 Monitoring, fault detection and prognosis

ZDM aims to minimize the waste produced by manufacturing processes (*zero-waste*), which also encompasses energy consumption minimization. These guidelines clearly lead to final cost reduction.

To minimize waste, it is then essential to reduce the required amount of processed products that have to measured in order to initialize any ZDM-oriented algorithm, as initial parts may be defective and may require to be disposed of. The fourth problem rephrases this concept.

**P4.** *To propose methodologies that minimize the amount of processed parts required to initialize the quality-ensuring algorithms.*

Continuing with the zero-waste premise, we focus now on tool quality in machining processes. When cutting tool wear surpasses a certain threshold, the processed products present unacceptable surface quality properties. However, replacing cutting tools prematurely leads to misusing resources. In both extremes, unnecessary waste is generated.

Usually, cutting tool wear cannot be directly measured without halting the machining process; therefore, indirect measurements must be taken. Then, the fifth problem is defined as:

**P5.** *To develop autonomous algorithms that adapt models with the objective of indirectly monitoring tool wear and other machining conditions and apply prognosis methods to determine the optimal moment to replace cutting tools.*

Due to the complexity of manufacturing processes, process and product measurements are not always available. Sometimes they can only be obtained occasionally, or they may present disturbances that reduce its reliability. These issues create difficulties for monitoring the process, so the sixth challenging problem we have to overcome is:

**P6.** *To develop process monitoring techniques when measurements are scarce or unreliable.*

Closely related to the previous issues, in some cases, measurements are available, but taking these measurements can be costly. When detecting and isolating faults in the process, there are different combinations of product measurements that can be used, which leads us to the seventh problem.

**P7.** *To develop fault and isolation methods that minimize the total measurement cost.*

## 1.3 Thesis outline and contributions

The main goal of this thesis is to propose methodologies to adjust models and develop fault diagnosis and prognosis algorithms in manufacturing processes, within the principles of defect, cost, time and waste minimization, encompassed under the ZDM strategies.

This thesis is structured as follows. First, in Chapter 2 we briefly introduce the basics of the Stream-of-Variation models, fault diagnosis and tool wear theory, which are required for the following chapters. In Chapter 3 we present geometrical models of two fixtures that are traditionally used in the manufacturing industry. In Chapter 4 we present an algorithm to adjust the geometrical model of an MMP using process data and engineering information. In Chapter 5 we propose a sequential algorithm for fault detection minimizing the amount of necessary measurements. Next, in Chapters 6 and 7 we focus on indirectly monitoring tool wear in CNC machines, and propose several algorithms for predicting the remaining useful life of machining tools, which adapt models in the presence of disturbances, noise and different measurement frequencies. Lastly, in Chapter 8 we present the conclusions of the thesis and we propose future research lines.

Please note that each chapter is self-consistent and can be mostly read independently of the other chapters. Most of the content in this thesis is derived from published or under peer review material; thus, repetitions of background explanations might appear, and notation between chapters might present differences. In order to end the introduction, let us present a summary of each chapter and reference the papers that are based on.

### Chapter 2: Background

In this Chapter, we present a general review of the theory basics to understand the following chapters of this thesis.

First, we define the concepts of multistage processes and its variants. Then, we clarify the difference between defects, deviations, faults and failures, and define the parts of fault diagnosis. We proceed with an explanation of the Stream-of-Variation methodology and several manufacturing process-related terms. After that, we review the basics of tool wear. Finally, we describe several methodologies and algorithms from the systems engineering field.

### Chapter 3: Extension of the Stream-of-Variation Model for General-Purpose Workholding Devices: Vices and Three-Jaw Chucks

This Chapter develops physical variation propagation models (**P1**) of a vice and a self-centering three-jaw chuck, which are fixtures that are frequently used in traditional manufacturing process industries (**P2**).

These models define the defect propagation from the variation sources from the process towards the processed products in a single stage. It also defines the propagation of defects

from previous stages in a multistage manufacturing process, as defects are transmitted through datums.

The models have been obtained through geometrical means within the Stream-of-Variation (SoV) approach using Differential Motion Vectors (DMVs). Part of the models have been obtained using the 3-2-1 model, as detailed in [10]. An alternative method has been proposed to solve the coordinate system problems in the self-centering three-jaw chuck, and no-linearity behavior in vices has been calculated using direct geometrical calculations.

The resulting models have been validated through mathematical simulations, CAD and experiments.

The results of this chapter were mainly addressed in [17]:

- **Moliner-Heredia, R.**, Abellán-Nebot, J. V., & Peñarrocha-Alós, I. (2021). Extension of the Stream-of-Variation Model for General-Purpose Workholding Devices: Vices and Three-Jaw Chucks. IEEE Transactions on Automation Science and Engineering.

### Chapter 4: A methodology for data-driven adjustment of variation propagation models in multistage manufacturing processes.

In Chapter 3, we found the existence of unavoidable non-linearities in traditional fixtures. These non-linearities may not affect the reliability of variation propagation models in single stages, but they can accumulate in MMPs with large amount of stages.

Thus, motivated by these conclusions, this Chapter proposes a methodology to adjust a linear input/output variation propagation model of a process using output data from the process and engineering knowledge, using a previously obtained physical model as a basis (**P3**).

This methodology consists in the recursive optimization of the difference between the covariance matrix of the measurements from the key dimensional characteristics of the product and the respective estimation obtained from the model that is being adjusted.

The adjusted hybrid model will fit the model better than the original model (**P1**), thus increasing its reliability in order to be used in monitoring and fault diagnosis methods.

The proposed methodology has been designed to require a reduced amount of measurements in order to be ready for its implementation (**P4**). This methodology has been validated using a simulated case study, and we have evaluated the evolution of the reliability of the adjusted model with respect to the amount of processed parts required to initialize the implementation.

The results of this chapter were mainly addressed in:

- **Moliner-Heredia, R.**, Peñarrocha-Alós, I. & Abellán-Nebot, J. V.. A methodology for data-driven adjustment of variation propagation models in multistage manufacturing processes. *Currently under peer review.*

### Chapter 5: A Sequential Inspection Procedure for Fault Detection in Multistage Manufacturing Processes

In Chapter 5, we present a procedure to detect and isolate faults online in a process with the objective of reducing the necessary amount of required measurements. This procedure is based on sequential inspections, which is a fault isolation method used in other knowledge areas, such as software testing.

First, we have developed a propagation model of the process-product relationship using knowledge from the planning process (**P1**). This linear input-output model of the process consists of a binary propagation matrix, where the propagated information is the presence of faults in the process.

The sequential inspection procedure consists of a detection stage and a fault isolation stage. In the first stage we select the minimum amount of required sensors that are necessary to take measurements and potentially detect the presence of a fault. In the second stage, depending on which sensors have detected this fault, we calculate an Information Gain with the objective of minimizing the amount of sensors required to isolate the detected fault, thus selecting the sensors that isolate the fault, at minimum total cost (**P7**).

We have validated the proposed procedure using different simulated case studies, analyzing the relationship between both the amount of required sensors and the total cost and the structure of the binary propagation matrix.

The results of this chapter were mainly addressed in [18]:

- **Moliner-Heredia, R.**, Bruscas-Bellido, G. M., Abellán-Nebot, J. V., & Peñarrocha-Alós, I. (2021). A Sequential Inspection Procedure for Fault Detection in Multistage Manufacturing Processes. *Sensors, 21(22), 7524.*

### Chapter 6: Model-based observer proposal for surface roughness monitoring

In this Chapter, we propose an observer to indirectly monitor the surface roughness of the processed parts in a CNC machine (**P5**), as we assume that surface roughness cannot be measured constantly (**P6**).

The proposed observer is based on a steady-state Kalman filter. Using additional measurements of the power consumed by the CNC machine and a model that is based on experimental curves that relate this power consumption and the surface roughness of the processed parts (which is affected by tool flank wear), during the monitoring process the observer is updated depending on the received measurements from the machining process.

This observer has been validated and compared with simpler proposals using a simulated case study.

The results of this chapter were mainly addressed in [19]:

- **Moliner-Heredia, R.**, Abellán-Nebot, J. V., & Peñarrocha-Alós, I. (2019). Model-based observer proposal for surface roughness monitoring. *Procedia Manufacturing, 41, 618-625.*

### Chapter 7: Model-based tool condition prognosis using power consumption and scarce surface roughness measurements

Continuing with the general concepts of Chapter 6, in this Chapter we propose a methodology to indirectly monitor the effects of tool flank wear in a CNC machining process and predict the optimal moment to replace the machining tool (i.e. predicting the remaining useful life of the tool) (**P5**).

This methodology assumes that measurements of the surface roughness of the processed parts are scarce (**P6**), and power consumption measurements are constantly available but contain notable measurement noise.

The models that this methodology uses to perform the prognosis are based on polynomial models of the relationship between power consumption and surface roughness during the machining tool life, which in turn have been adjusted from empirical models of these behaviors. These polynomial models are subsequently adjusted online by an adaptive recursive least squares algorithm, using measurements from the machining process (**P1**). The last part of the methodology determines the tool replacement moment using the adjusted models. We propose two alternatives to determinate the replacement moment.

We have validated the proposed methodology using a simulated case study where several tools are consumed and replaced. We have checked that the algorithm requires a small amount of data (**P4**) to successfully update the model parameters when cutting conditions change. Finally, we have compared the performance of our methodology with other alternatives of remaining useful life prognosis.

The results of this chapter were mainly addressed in [20]:

- **Moliner-Heredia, R.**, Peñarrocha-Alós, I., & Abellán-Nebot, J. V. (2021). Model-based tool condition prognosis using power consumption and scarce surface roughness measurements. *Journal of Manufacturing Systems, 61, 311-325.*

### Chapter 8: Conclusions and future research

In the last chapter, we present the conclusions of this thesis and propose several research topics for its potential development in the future.

### 1.3.1 List of contributions

During the period of development of this thesis, the author has contributed to the following papers, which are currently published or under peer review:

#### Journal papers

- **Moliner-Heredia, R.**, Abellán-Nebot, J. V., & Peñarrocha-Alós, I. (2021). Extension of the Stream-of-Variation Model for General-Purpose Workholding Devices: Vices and Three-Jaw Chucks. *IEEE Transactions on Automation Science and Engineering.* [17]
- **Moliner-Heredia, R.**, Peñarrocha-Alós, I., & Abellán-Nebot, J. V. (2021). Model-based tool condition prognosis using power consumption and scarce surface roughness measurements. *Journal of Manufacturing Systems, 61, 311-325.* [20]
- **Moliner-Heredia, R.**, Bruscas-Bellido, G. M., Abellán-Nebot, J. V., & Peñarrocha-Alós, I. (2021). A Sequential Inspection Procedure for Fault Detection in Multistage Manufacturing Processes. *Sensors, 21(22), 7524.* [18]
- **Moliner-Heredia, R.**, Peñarrocha-Alós, I. & Abellán-Nebot, J. V.. A methodology for data-driven adjustment of variation propagation models in multistage manufacturing processes. *Currently under peer review.*
- **Moliner-Heredia, R.** & Abellán-Nebot, J. V. Evaluation of the impact of gamification on students' performance and engagement in manufacturing engineering courses. *Currently under peer review.*

#### Conference papers

- **Moliner-Heredia, R.**, Abellán-Nebot, J. V., & Peñarrocha-Alós, I. (2019). Model-based observer proposal for surface roughness monitoring. *Procedia Manufacturing, 41, 618-625.* [19]
- Abellán-Nebot, J. V., **Moliner-Heredia, R.**, Bruscas, G. M., & Serrano, J. (2019). Variation propagation of bench vises in multi-stage machining processes. *Procedia Manufacturing, 41, 906-913.* [21]
- **Moliner-Heredia, R.**, Peñarrocha-Alós, I., & Sanchis-Llopis, R. (2019, July). Economic model predictive control of wastewater treatment plants based on BSM1 using linear prediction models. *In 2019 IEEE 15th International Conference on Control and Automation (ICCA) (pp. 73-78). IEEE.* [22]
- Tena, D., Peñarrocha-Alós, I., Sanchis, R., & **Moliner-Heredia, R.** (2020). Ammonium Sensor Fault Detection in Wastewater Treatment Plants. *In ICINCO (pp. 681-688).* [23]

Chapter 2

# Background

This thesis applies systems engineering knowledge to manufacturing process engineering. In order to clarify any doubts that readers from each field of knowledge may have, in this Chapter we explain some basic concepts that will be useful to understand this thesis.

First, we present the definition of multistage processes and its variations. Then, we present the differences between defects, faults and other related terms, as well as the different parts of fault diagnosis. After that, we define several concepts related with the Stream of Variation approach to model multistage processes. We continue with some definitions of key concepts in the field of tool wear. Finally, we define additional concepts related to systems engineering.

## 2.1 Multistage processes

**Multistage manufacturing process (MMP).** A multistage manufacturing process is a complex manufacturing system involving multiple stations (or stages) to produce a product. The quality of the product can be qualitatively characterized by several features or attributes, and the product quality variations are contributed by the errors generated at the current station and the accumulated errors from previous stations [9].

**Multistage assembly process (MAP).** A multistage assembly process is a subtype of MMP where assembly operations are performed. Typical examples include automotive assemblies and aircraft fuselage assemblies [9].

The ideal multistage process consists of a direct single line of stations where only a single type of product is processed. In this thesis, we have taken this assumption. However, MMPs are usually more complicated, and may include branched lines, different product types sharing some stations or rework loops.

## 2.2 Defects, faults and fault diagnosis

### 2.2.1 Defects and deviations

**Defect.** A defect is the non-fulfillment of a requirement, related to an intended or specified use. This requirement refers to a need or expectation that is stated, generally implied or obligatory [24]. Then, in the context of this thesis, the non-fulfillment of a requirement refers to failing to manufacture a product that offers a certain functionality.

**Deviation.** A deviation of a feature consists of the deviation of the actual coordinate system of that feature from their nominal coordinate system [10]. Then, this deviation consists of a distance and orientation displacement of the feature. A deviation does not imply a *fault* in the product.

**Variation.** In many cases, used as a generic equivalent to *deviation*, without specifying feature displacements. The *variation sources*, or sources of variation, are the parts of the process (e.g. fixtures, cutting tools, etc) that may cause variation in the processed workpieces. The variation sources that are important due to its effect on the final product are often called key control characteristics (KCC) in the literature [9].

**Key product characteristic (KPC).** Key product characteristics are critical *features* of a product [9]. Proper manufacturing of these features is essential for the correct functionality of the product. KPCs include the dimensional and geometrical characteristics of the product, together with their admissible deviations. Thus, in this thesis, we sometimes use the term KPC to refer to these deviations.

### 2.2.2 Faults and failures

**Fault.** A fault is a non permitted deviation of a characteristic of the system, which leads to the inability to fulfill its intended purpose. This malfunction may be tolerable in a given stage, but it is important to diagnose any faults as early as possible in order to prevent any consequences [25, 26].

**Failure.** A failure is a complete breakdown of the component or function of the system [26].

In this thesis, we have proposed methods for diagnosing faults and predicting future faults before they result in failures.

### 2.2.3 Fault diagnosis

**Fault diagnosis.** Fault diagnosis is a procedure to detect, locate and estimate the significance of faults in a system, using the information of a monitoring system as a basis [26, 27]. It consists of three tasks:

- **Fault detection.** The objective of fault detection is to determine the occurrence of faults in the system, i.e., whether a fault has occurred or not.
- **Fault isolation.** The objective of fault isolation is to locate the fault, i.e., which part of the system is faulty.
- **Fault identification.** The objective of fault identification is to identify the type, magnitude and behavior of the fault.

In this thesis, we have focused on the parts of fault detection and isolation in manufacturing systems.

## 2.3 Stream of Variation - Basics

**State space model.** Also written as state-space model, it is a mathematical model of a dynamical system. In this thesis, we have focused on discrete space state models.

**Discrete time linear state space models (Discrete time linear SSM).** In discrete time linear SSMs [28], the output of the system depends on the current input and the internal states of the system. The values of the internal states of the system depend on the input and the previous state of those internal state variables. The discrete time linear SSM presents the following form:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) + w(k), \\ y(k) = Cx(k) + Du(k) + v(k), \end{cases} \tag{2.1}$$

where $k$ indicates the current discrete instant of time; $x(k)$ contains the internal state variables, $u(k)$ the system input, $y(k)$ the system output, $w(k)$ are disturbances in the form of model errors, and $v(k)$ the measurement noise, all of the above expressed in instant $k$. Matrices $A$, $B$, $C$ and $D$ represent the dynamics of the system. Here, we assume that they do not change with time.

**Stream of Variation (SoV).** The Stream of Variation (also written as Stream-of-Variation) methodology focuses on developing a linear mathematical model that links the KPCs with the KCCs using a state space model [9], i.e., a model that represents how variation "flows" through the production stream until the end of the process. The SoV model presents the following form [10]:

$$\begin{cases} x_k = A_{k-1}x_{k-1} + B_k u_k + w_k, \\ y_k = C_k x_k + v_k. \end{cases} \tag{2.2}$$

Here, $k$ represents the current stage where the part is being processed, the internal states $x_k$ are the deviations of certain features of the workpiece when it is processed in station $k$, $A$ is used to transmit datum errors from previous stations, $u_k$ are the errors produced by the variation sources in stage $k$, $B$ quantifies and locates the effects on the variation sources on the corresponding feature deviations, $w_k$ is considered unmodeled system noise, $y_k$ is the final product measurements of the KPCs, $C_k$ is a measurement selection matrix and $v_k$ is the measurement noise.

The Stream of Variation methodology, including the development of matrices $A$ and $B$, is thoroughly developed in Chapter 3.2.

**Linear input-output model.** The SoV model from (2.2) can be transformed into a linear input-output model [9]:

$$y = \Gamma\ u + v, \tag{2.3}$$

where $y$ is a vector that contains all the measurements of the KPC deviations after leaving the MMP, $u$ is a vector that contains the representation of all the variation sources of the process, $v$ includes the measurement noise and unmodeled system noise, and $\Gamma$ is a matrix that represents the model of the relation between KPCs and variation sources of the process.

Matrix $\Gamma$ is a function of matrices $A$, $B$ and $C$ from the equations in (2.2) of each station of the MMP. Thus, any approximations and errors committed during the development of these matrices are propagated to $\Gamma$. Therefore, in this thesis we wonder if we could develop a methodology to adjust or update $\Gamma$ with process data in order to increase its reliability.

**Workpiece.** Physical element whose shape is modified during the manufacturing process. In the case of machining processes, the material of the workpiece is usually metallic. We denominate *raw* workpieces to those that enter the manufacturing process.

In this thesis, we use the term *part* interchangeably, and *processed part* or *product* if it has left the manufacturing process.

**Locators.** Locators are devices used in manufacturing processes whose function is locating the workpiece with respect to a machine reference system.

**Fixture.** A fixture is a device used in manufacturing processes that fixes the workpiece during the machining process. Its main functions are countering the cutting forces that act on the part during the process, and locating and orienting the workpiece with respect to a machine reference system.

**Fixture and locator errors.** These errors are caused by an incorrect positioning of the locators or parts of the fixture, which end locating the workpiece in a different position and orientation than its nominal.

**Cutting tools.** Cutting tools are devices used in machining process to tear small fragments of the workpiece (named chips) in order to give it the desired shape. Some cutting tools are made entirely with the same material, while others are composed by an insert (which cuts the workpiece) wielded or clamped to a toolholder.

Cutting tools deteriorate during its use, leading to lower cutting quality performance. After a given time, they must be replaced.

**Machining errors.** These errors can be caused by several different factors, such as thermal deformation of the machine, cutting tool deflection caused by excessive applied forces or tool wear, for example.

**Geometrical feature.** A geometrical feature is a point, line, surface, volume or set of these items [29]. A geometrical feature can be an ideal (parametrical-based) or a non-ideal (real-based) feature.

**Datum.** A datum is a feature associated to a real (non-ideal feature) selected to define a location or orientation of other ideal features. The non-ideal feature used to establish a datum is called *datum feature*.

If a datum is not influenced by constraints from other datums, it is called *primary datum*. If a datum is influenced by orientation constraints of a primary datum, it is a *secondary datum*. If a datum is then influenced by a primary and a secondary datum, it is a *tertiary datum* [30].

**Datum errors.** These errors are caused when a non-ideal feature with location or orientation errors is used as a locating datum for generating a new feature in the workpiece. Thus, these errors are propagated within the workpiece.

**Clamping errors.** These errors consist of the deviation of the location of the workpiece due to deformation caused by applied clamping forces. In this thesis, we have assumed that the complete system is rigid, so these errors are not considered.

**Process planning.** Process planning consists in determining the methods to manufacture a product under economic and competitive premises [31]. It selects and specifies the processes, machine tools, fixtures, and the order of the operations to shape raw workpieces into finished and assembled products.

## 2.4 Tool wear

Cutting tools deteriorate during the cutting process. This deterioration can be classified in two different classes [32]:

**Chipping.** Chipping is the occurrence of cracks in the cutting part of the tool, which leads to loss of small fragments due to brittle fracture.

**Tool wear.** Tool wear is the change in shape of the cutting part of the tool, due to progressive loss of material during the cutting process.

Tool wear may be produced on the flanks of the tool (which are in contact with the newly created real feature in the workpiece) or on the tool face (which is in contact with the generated chips). In this thesis, we focus on the former, called *tool flank wear*, as it directly affects the quality of the finished products and thus, it is a variation source.

**Remaining useful life (RUL).** The remaining useful life of a part of a process is the total time that can be used before it ceases functioning properly, i.e. providing results out of expected specifications.

Focusing on the case of cutting tools, in practical workshop situations, the end of the useful life of a tool is usually determined by the moment in which produces workpieces with size or surface quality out of specifications. Strictly speaking, tool life is determined by tool deterioration in the form of tool wear [32].

The objective of RUL methods is estimating the remaining useful life of a cutting tool using physical-, data- or model-based methods [16].

**Tool condition monitoring (TCM).** Tool condition monitoring methods focus on estimate the current state of the cutting tool wear, specially tool flank wear, as it affects the quality of the output products and the useful tool life. There are two different types of tool condition methods:

- **Direct methods.** Direct methods consist in stopping the machining process, extracting the cutting tool and observing it through a microscope.
- **Indirect methods.** Indirect methods consist in estimating the state of the tool wear using other outputs of the process, such as cutting forces, power consumption of the machine, vibrations, surface quality of the processed parts or changes in temperature.

Due to the fact that indirect methods can be applied during the machining process, the latest research has been oriented towards developing these methods [33]. In this thesis, we wonder if we could relate some of the output signals of the process, in order to obtain an enhanced prognosis algorithm to model the state of the cutting tool wear, specially in cases where some of the output signals are scarcely obtained.

### 2.4.1 Tool life equations

Taylor's equation for tool life [34] was developed after performing several experimental tests to obtain the relation between the useful life of a tool and the cutting speed and other cutting parameters. The equation presents this form:

$$V_c \cdot T^n = C, \tag{2.4}$$

where $V_c$ is the cutting speed, $T$ is the useful life of the tool, $n$ depends on the cutting tool material, and $C$ is a constant term. The useful life of the tool here is related to the moment where tool flank wear surpasses a certain threshold.

This equation was updated afterwards to include the effects of the feed rate and the depth of the cut:

$$T = \frac{C}{V_c^x \cdot s^y \cdot t^z}, \tag{2.5}$$

where $s$ is the feed rate, $t$ the depth of cut, and $x$, $y$ and $z$ are constants that can be obtained experimentally.

These equations are useful as initial approximations, but cannot be used to readjust and recalculate the remaining useful life of the tool during the machining process. Therefore, other

correlations between the remaining useful tool life, tool wear and output signals that can be obtained during the machining process have been researched.

For example, [35] states that tool flank wear increases non-linearly with respect to the total cutting time of the tool, and obtains an experimental equation of this relation. Other examples include [36], where the proportional relationship between tool flank wear and cutting forces is defined, and thus, it relates them to the power consumed during the cutting process; or in [37], where an empirical equation relating tool wear and the surface roughness of the processed parts is presented.

The best relationship between these variables to develop models to calculate the RUL should be chosen depending on the available signals and its reliability.

## 2.5 Systems engineering terminology

**Prognosis.** Prognosis consists in forecasting the most probable result of a situation. It consists of two steps; first, obtaining an analytical model of the situation; then, update the model with any new information and predict the future development of the situation [16].

**Observer.** Given the equation system

$$\begin{cases} x_{k+1} = Ax_k + Bu_k, \\ y_k = Cx_k, \end{cases} \tag{2.6}$$

consider that we know $A$, $B$ and $C$, and do not have access to $x$, We want to estimate the state of $x$ in the next step. The equation is:

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + L(y_k - C\hat{x}_k), \tag{2.7}$$

where $A\hat{x}_k + Bu_k$ is the open loop estimation, and $L(y_k - C\hat{x}_k)$ the correction term, which uses the output of the system to correct the initial estimation, as well as the gain matrix $L$ [38]. This matrix can be obtained using different methods that focus on obtaining an equilibrium between minimizing the estimation error as soon as possible and being robust upon disturbances. Some examples include pole assignment or the Kalman filter. Depending on the characteristics of the system, there are different variants of the Kalman filter that can be developed.

**Steady-state Kalman filter (SSKF).** A Kalman filter that is used when the variances of the model disturbance and measurement noise are static in time. Given the system:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k + Gw_k, \\ y_k = Cx_k + v_k, \end{cases} \tag{2.8}$$

where for disturbance $w_k$, $E\{w_k\} = 0$ and $E\{w_k w_k^\intercal\} = W$; and for measurement noise $v_k$, $E\{v_k\} = 0$ and $E\{v_k v_k^\intercal\} = V$.

The goal of the SSKF is reducing the average estimation error of $x$ to 0 and minimize its quadratic error. Depending on whether we consider that our model ($A$, $B$ and $C$) is more reliable than our sensor measurements, we decrease or increase $W$ and $V$ and calculate the gain matrix $L$.

**Recursive Least Squares.** A recursive algorithm that is used online to filter and obtain the coefficients of a polynomial that minimizes an error function. If the algorithm contains a parameter that is decision-based, it is considered an *Adaptive Recursive Least Squares (ARLS)*. In this thesis, we have implemented an ARLS algorithm in Chapter 7.6.

**Optimization problem.** An optimization problem presents the following form [39]:

$$\min_x \ f_0(x) \tag{2.9a}$$

$$\text{s.t.} \ \ f_i(x) \leq b_i, \ i = 1, ..., m \tag{2.9b}$$

where vector $x = (x_1, ..., x_n)$ is the optimization variable, $f_0 : \mathbb{R}^n \to \mathbb{R}$ is the objective function, and $f_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, ..., m$ are constraint functions. Solving the optimization problem consists in obtaining the optimal vector $x^*$, which yields the minimum objective value of the function while satisfying the constraint functions.

**Convex optimization problem.** An optimization problem is convex if the objective and constraint functions are convex, i.e., they satisfy the following conditions [39]:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y), \ i = 0, 1, ..., m \tag{2.10}$$

for all $x, y \in \mathbb{R}^n$ and all $\alpha, \beta \in \mathbb{R}$ with $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$.

Convex optimization problems can be solved efficiently, but in can be difficult to recognize a convex function or rewrite nonlinear optimization problems into convex optimization problems. There are many methods in the literature to rewrite those problems.

### 2.5.1 Schur complement

The Schur complement is a technique to reduce the size of linear systems [40]. Given a linear system

$$Mz = 0, \tag{2.11}$$

where matrix $M$ is partitioned as

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{2.12}$$

and where $A$ is nonsingular, and $z$ is partitioned as

$$z = \begin{bmatrix} x \\ y \end{bmatrix}, \tag{2.13}$$

linear system (2.11) is equivalent to

$$Ax + By = 0, \tag{2.14}$$

$$Cx + Dy = 0. \tag{2.15}$$

Multiplying (2.14) by $-CA^{-1}$ and adding it to (2.15), we eliminate $x$ and obtain a linear system of smaller size:

$$\left(D - CA^{-1}B\right) y = 0. \tag{2.16}$$

Thus, we define

$$M/A = D - CA^{-1}B \tag{2.17}$$

as the Schur complement of $A$ in $M$.
Operating likewise, we define the Schur complement of $D$ in $M$ as

$$M/D = A - BD^{-1}C. \tag{2.18}$$

The properties of the Schur complement can be used to convert nonlinear inequality constraints in convex Linear Matrix Inequalities (LMI) [41]. A LMI with the form

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^{\intercal} & R(x) \end{bmatrix} \succ 0, \tag{2.19}$$

where $Q(x)$ and $R(x)$ are symmetrical, and both of them and $S(x)$ depend affinely on $x$, is then equivalent to

$$R(x) \succ 0,\ Q(x) - S(x)R(x)^{-1}S(x)^{\intercal} \succ 0. \tag{2.20}$$

These equivalences can also be generalized to nonstrict inequalities. In this case,

$$\begin{bmatrix} Q(x) & S(x) \\ S(x)^{\intercal} & R(x) \end{bmatrix} \succeq 0 \tag{2.21}$$

is equivalent to

$$R(x) \succeq 0,\ Q(x) - S(x)R(x)^{\dagger}S(x)^{\intercal} \succeq 0,\ S\left(I - RR^{\dagger}\right) = 0, \tag{2.22}$$

where $R^{\dagger}$ is the Moore-Penrose inverse of R.

Chapter 3

# Extension of the Stream of Variation Model for General Purpose Workholding Devices: Vices and 3-jaw Chucks


## Abstract

Nowadays, advanced manufacturing models such as the Stream-of-Variation (SoV) model have been successfully applied to derive the complex relationships between fixturing, manufacturing and datum errors throughout a multi-stage machining process. However, the current development of the SoV model is still based on 3-2-1 fixturing schemes and, although some improvements have been done, e.g. N-2-1 fixtures, the effect of general workholding systems such as bench vices or 3-jaw chucks has not yet been included into the model.

This chapter presents the extension of the SoV model to include fixture and datum errors considering both bench vices and 3-jaw chucks as a fixturing devices in multi-stage machining processes. The model includes different workholding configurations and it is shown how to include the workholding accuracy to estimate part quality. The extended SoV model is validated in a 3-stage machining process by both machining experimentation and CAD simulations.

Note to Practitioners: Part quality estimation in multi-stage machining systems is a challenging issue. The Stream of Variation (SoV) model is a straightforward model that can be used for this purpose. However, current model is limited to fixture based on punctual locators and common shop-floor devices are not considered yet. To overcome this limitation, this chapter extends the current SoV model to include vices and 3-jaw chucks as workholding devices. The proposed methodology let practitioners to estimate the manufacturing capability of a process considering the technical specifications of these devices (e.g., parallelism and perpendicularity of vice surfaces, total indicator runout of chucks) or it can be used for diagnosing workholding issues. The model assumes that the workpiece acts as a rigid part and errors due to deformation during clamping are assumed to be negligible in comparison with fixture- and datum-induced errors.

## Nomenclature

| | |
|---|---|
| ${}^0$FCS, FCS | Nominal and actual fixture coordinate system |
| ${}^0\mathbf{H}_F^R$, $\mathbf{H}_F^R$ | Nominal and actual Homogeneous Transformation Matrix (HTM) between RCS and FCS |
| $\mathbf{\Delta}_F^R$, $\delta\mathbf{H}_F^R$ | Differential and deviation transformation matrix between RCS and FCS |
| $\mathbf{x}_F^R$ | DMV representing the deviation of FCS in RCS. |
| $\mathbf{d}_F^R$, $\boldsymbol{\theta}_F^R$ | Position and orientation deviation of FCS with respect to (w.r.t.) RCS |
| $\mathbf{t}_F^R$, $\boldsymbol{\omega}_F^R$ | Position and orientation vector of FCS w.r.t. RCS |
| $\hat{\boldsymbol{\theta}}_F^R$ | Skew symmetric matrix from $\boldsymbol{\theta}_F^R$ |
| $\mathbf{x}_k$ | Vector with the DMV of all features stacked up at stage $k$ |
| $\mathbf{u}_k^f$ | Fixture errors at stage $k$ |
| $\mathbf{A}_k^3$ | Fixture-induced variation matrix in SoV model |
| $\mathbf{A}_k^2$ | Datum-induced variation matrix in SoV model |

## 3.1 Introduction

Manufacturing processes have to be environmental friendly and safe and deliver high quality products rapidly adapted to customer requirements at a minimum cost. One of the most important challenges in modern industry is the implementation of manufacturing systems capable of generating products with zero defects. A recent roadmap promoted by the European Comission in the research area of zero-defect manufacturing processes has presented the state of the art, the gap to be overcome and the research priorities and future trends in this field [42]. According to the roadmap, a research priority for the development of zero-defect manufacturing processes is related to the "integration of machine, fixture, tool and workpiece models for quality and resource deterioration prediction" in multi-stage manufacturing processes (MMPs).

MMPs are manufacturing processes that consist of a sequence of stages where manufacturing operations such as assembly or machining operations are sequentially conducted to manufacture a part or product. Typical examples of MMPs are automobile body assembly processes and multi-job machining processes where a part moves from one stage to another until a semi-finished or finished product is obtained. Due to the sequential nature of these manufacturing operations, the error generated at the first stages may be propagated downstream to other stages which produces additional manufacturing errors. These complex error interactions make difficult to control product quality and tasks such as predictive maintenance, process control, quality assurance and fault diagnosis are challenging.

In order to illustrate the error propagation in a MMP, consider a MMP composed of 3 stages in a machining line as shown in Figure 3.1. As it can be observed, stage 1 presents a deviation of the cutting-tool trajectory, producing a machining-induced error. The resulting part moves to stage 2, where there are no additional errors. However, since the previous machined surface is used as a datum surface, the square shoulder machining operation is deviated with respect to the top surface, producing a datum-induced error. Finally, the part is moved to stage 3,

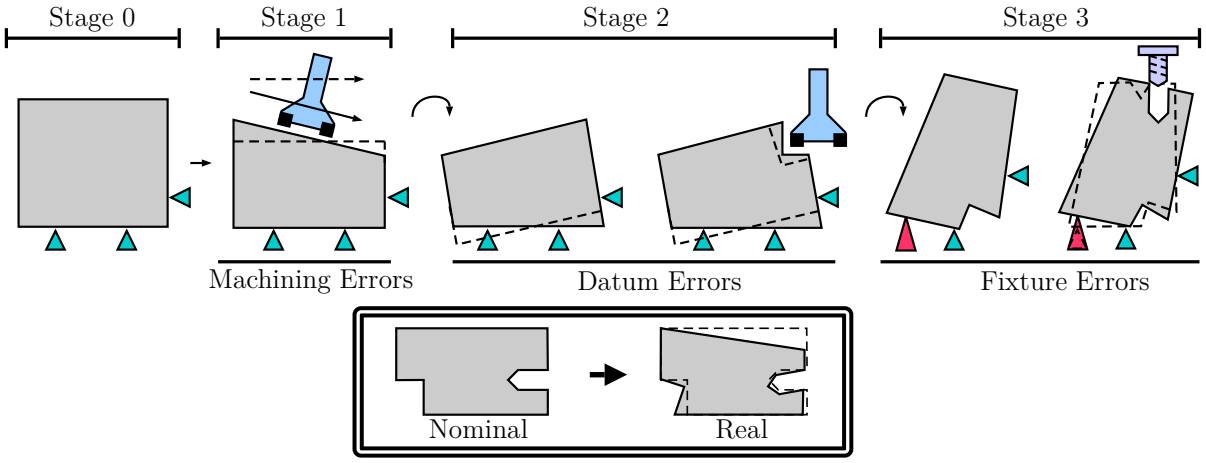



**Figure 3.1.** Error transmission in Multi-stage Manufacturing Processes (MMP).

where a locator has been deviated from its nominal position, which produces a deviation of part location and thus, the drill is misplaced producing a fixture-induced error. As it can be seen, in MMPs where machining operations are conducted, three main sources of errors arise: *Machining-induced errors*, *Datum-induced errors* and *Fixture-induced errors*. Note that a similar reasoning can be conducted in assembly lines where welding operations are performed instead of machining operations.

Despite being very common manufacturing systems in industry, the MMPs are usually too complex to be mathematically modeled and the development of tools and strategies for effective quality assurance and fault diagnosis is currently a challenging task that hinders the deployment of zero-defect manufacturing processes. In the literature, some approaches have been proposed to model the error propagation within these types of manufacturing systems. One of these approaches is the so-called Stream-of-Variation (SoV) approach, which was successfully developed in the late 90s for multi-stage assembly processes by Jin and Shi [43]. The SoV model is based on the State-Space Model from control theory to define mathematically the relationships between fixture and machining errors on machined surfaces, and the datum errors are introduced to link the errors between the stages. The SoV model was expanded to include multi-stage machining processes in [44] and later, the model was highly improved by Zhou et al. [10] with the introduction of Differential Motion Vectors (DMVs) to model the small displacements of each geometrical feature as it is used in the field of robotics [45]. This model can be considered as the SoV reference model within multi-stage machining processes, where the methodology to derive the model is explained in detail under the limitation of fixture devices based on 3-2-1 punctual locators. The model was expanded by Abellán-Nebot et al. [46] to include specific machining errors such as tool wear errors, deflection errors, kinematic errors from tool axis, and so on. In regards to assembly processes, the SoV model was firstly developed for rigid sheet metal parts in [43] but it was later extended to deal with compliant sheet metal parts by Camelio et al. [47]. In [48], the SoV model was expanded to deal with compliant parts using N-2-1 locating schemes based on punctual locators and later, the mathematical derivation to consider 3D rigid assemblies instead of sheet metal parts was presented in [49]. More recently the model was also

extended to deal with composites in multi-stage assembly processes for the aeronautic industry considering compliant parts with anisotropic properties [50].

The application of the SoV model has been widely studied in the last two decades and promising results have been presented in different fields such as fault diagnosis and quality control [51–56], process planning [57, 58], manufacturing tolerance allocation and predictive maintenance [59,60] and so on. However, despite the efforts made by many researchers, the SoV model still presents some drawbacks for its application in MMPs. One of the major criticisms refers to fixture error modeling, which is focused on punctual locators based on 3-2-1 schemes or N-2-1 schemes when compliant parts are considered, but positioning cases with plane/plane contact or cylinder/cylinder floating contact are not considered [61]. Under 3-2-1 schemes, the touching points between the locating surface and the fixture device are known, and the mathematical model that relates the error of each locator and the deviation of workpiece location can be determined. However, other common fixture devices such as vices or chucks do not follow this behavior, and the touching points of the locating surface and the fixture device may depend on previous errors. In this situation, the mathematical model between fixture errors and workpiece location errors cannot be determined in advance, and it will depend on the workpiece errors at the moment of clamping. This problem was tackled by Abellán-Nebot et al. [62], where a generic procedure for modeling fixtures based on surfaces instead of punctual locators was presented. Although the methodology deals with different configurations which depend on previous errors, the research does not deal with specific fixtures such as vices and omits other types of fixtures such as chucks. More recently, the inclusion of the bench vice errors into the SoV model has been introduced in [63]. However, the proposed methodology only showed the result for a specific vice without deriving a generic approach based on differential and homogeneous transformation matrices among fixture/workpiece features. Therefore, more general vices or alternative ones (i.e., rotatory or universal vices) cannot be modeled. Furthermore, the applicability of the methodology is limited since there is no clear use of common workholding specifications into the modeling approach, which prevents their use in industry.

This chapter shows a methodology to model the effect of fixturing errors from two common fixtures in MMPs: 3-jaw chucks and bench vices. The mathematical development of these models follow the structure of the SoV model proposed in aforementioned studies, providing compatibility with the general SoV approach. The model includes different workholding configurations and it is also shown how to include the workholding accuracy to estimate part quality. The mathematical derivation of the models is validated through both CAD simulations and machining experimentation proving the high accuracy of the model despite linearization errors. Please, note that despite the low accuracy of these workholding devices in comparison with dedicated fixtures, their level of clamping and locating accuracy can be enough for low volume production systems where manufacturing tolerances of tenths of a millimeter are allowed [64] and, thus, the inclusion of these devices into the SoV model may be of interest.

The chapter is organized as follows. Section 3.2 provides the general methodology of the SoV model in order to identify the parts of the model that have to be extended. Section 3.3 shows the mathematical derivation of the fixture- and datum-induced errors for bench vices, whereas Section 3.4 presents the mathematical derivation for 3-jaw chucks. Section 3.5 shows a case

study where a MMP with both types of fixtures is applied and the model is validated through CAD simulations and machining experiments. Finally, Section 3.6 shows the conclusions of the chapter.

## 3.2 Methodology Overview - The Stream of Variation model

The Stream-of-Variation (SoV) model uses the DMVs to define dimensional deviations of part features from nominal positions. As each feature is determined by a Local Coordinate System (LCS), DMVs define the displacement of each LCS from its nominal position ($^0$LCS). This displacement is composed of a position deviation vector $\mathbf{d}_L^{0L} = [d_{Lx}^{0L}, d_{Ly}^{0L}, d_{Lz}^{0L}]^T$ and an orientation deviation vector $\boldsymbol{\theta}_L^{0L} = [\theta_{Lx}^{0L}, \theta_{Ly}^{0L}, \theta_{Lz}^{0L}]^T$, so a DMV is defined as $\mathbf{x}_L^{0L} = [(\mathbf{d}_L^{0L})^T, (\boldsymbol{\theta}_L^{0L})^T]^T$. In regards of nominal values, each $^0$LCS is referred to the reference coordinate system (RCS) using a locating vector that defines its position $\mathbf{t}_{0L}^R = [t_{0Lx}^R, t_{0Ly}^R, t_{0Lz}^R]^T$ and its orientation $\boldsymbol{\omega}_{0L}^R = [\omega_{0Lx}^R, \omega_{0Ly}^R, \omega_{0Lz}^R]^T$. In this chapter, terms $\omega_{0Lx}^R, \omega_{0Ly}^R$ and $\omega_{0Lz}^R$ are expressed as proper Euler angles between RCS and $^0$LCS in a Z-Y'-Z" order (this means a rotation of RCS around its Z axis, followed by a rotation around the new Y axis, and lastly, a rotation around the new Z axis). Fig. 3.2 shows an example of a locating vector of a machined feature and its corresponding DMV to model the deviation of the feature from nominal values.

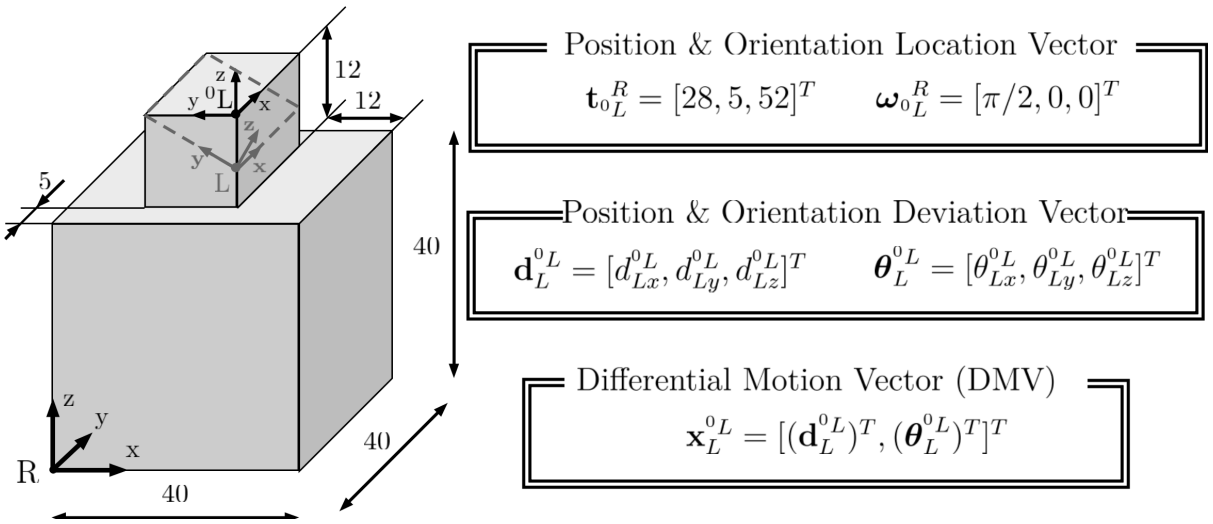



**Figure 3.2.** Example of a DMV in a machining process.

In the SoV model, the deviations of all features are stacked up in a vector, denoted as $\mathbf{x}_k = [(\mathbf{x}_k^1)^T, (\mathbf{x}_k^2)^T, ..., (\mathbf{x}_k^M)^T]^T$, where $k = 1, ..., N$ refers to the number of the stage and $\mathbf{x}_k^1, ..., \mathbf{x}_k^M$ are the DMV of features $1, ..., M$. As it was pointed out above, the error propagation throughout the MMP is conducted by the adoption of the State-Space Model from control theory. Under this framework, the SoV model in a MMP of N-stages can be defined as [10]

$$\mathbf{x}_k = \mathbf{A}_{k-1} \cdot \mathbf{x}_{k-1} + \mathbf{B}_k^f \cdot \mathbf{u}_k^f + \mathbf{B}_k^m \cdot \mathbf{u}_k^m + \mathbf{w}_k, \tag{3.1}$$

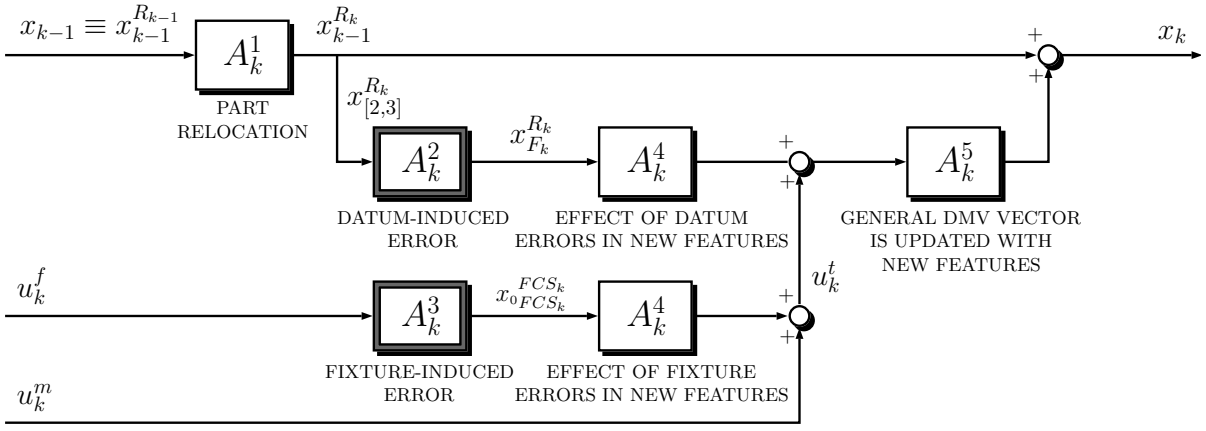



**Figure 3.3.** Steps for modeling the Stream of Variation (SoV) of a MMP.

where $\mathbf{A}_{k-1}\cdot\mathbf{x}_{k-1}$ represents the variations transmitted by datum features generated at upstream stages, $\mathbf{B}_k^f \cdot \mathbf{u}_k^f$ represents the fixture-induced variations within stage $k$, where $\mathbf{u}_k^f$ denotes the fixture errors; $\mathbf{B}_k^m \cdot \mathbf{u}_k^m$ represents the machining-induced variations within stage $k$, where the cutting-tool path deviation is denoted as $\mathbf{u}_k^m$; and $\mathbf{w}_k$ is the un-modeled system noise and linearization errors. The derivation of this model is detailed in [10], where it is presented the procedure to obtain matrices $\mathbf{A}_{k-1}$, $\mathbf{B}_k^f$ and $\mathbf{B}_k^m$ at each stage, according to given product and process information (part geometry and fixture layouts). Fig. 3.3 shows the auxiliary matrices to build the SoV main matrices according to the methodology detailed in Zhou's et al. research work [10]. Following their methodology, the matrices $\mathbf{A}_{k-1}$, $\mathbf{B}_k^f$ and $\mathbf{B}_k^m$ are defined as

$$\mathbf{A}_{k-1} = [\mathbf{A}_k^1 + \mathbf{A}_k^5 \cdot \mathbf{A}_k^4 \cdot \mathbf{A}_k^2 \cdot \mathbf{A}_k^1], \tag{3.2}$$

$$\mathbf{B}_k^f = [\mathbf{A}_k^5 \cdot \mathbf{A}_k^4 \cdot \mathbf{A}_k^3], \tag{3.3}$$

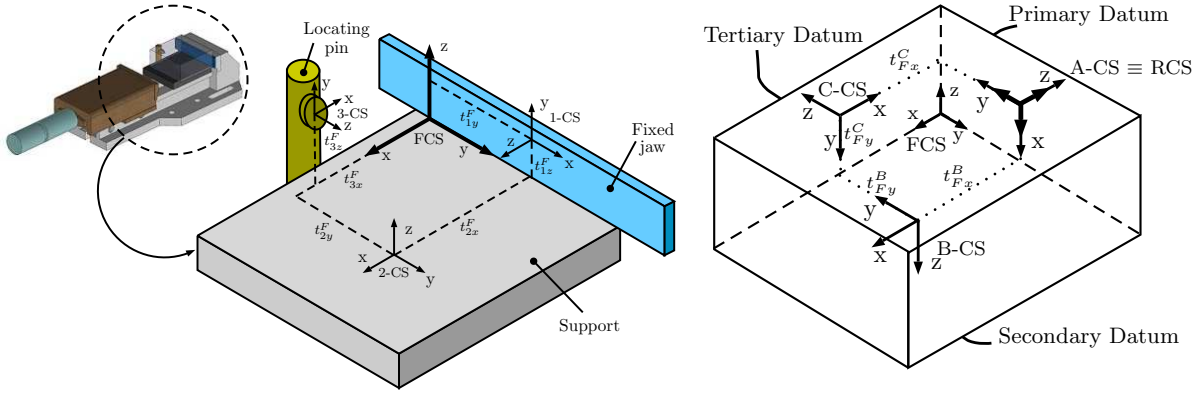
$$\mathbf{B}_k^m = [\mathbf{A}_k^5], \tag{3.4}$$

where $\mathbf{A}_k^1$ is the relocating matrix, $\mathbf{A}_k^2$ is the datum-induced variation matrix, $\mathbf{A}_k^3$ is the fixture-induced variation matrix, $\mathbf{A}_k^4$ is the feature generation matrix, and $\mathbf{A}_k^5$ is the selector matrix. Matrices $\mathbf{A}_k^2$ and $\mathbf{A}_k^3$ are currently derived for 3-2-1 punctual schemes [10], the N-2-1 extension [47], general punctual fixture configurations [48] and surface based fixtures [62].

The next sections presents the mathematical derivation of the corresponding matrices $\mathbf{A}_k^2$ and $\mathbf{A}_k^3$ when the workholding device is a bench vice or a 3-jaw chuck. Please note that in this chapter it is assumed that the workpiece acts as a rigid part and errors due to deformation during clamping are assumed to be negligible in comparison with fixture- and datum-induced errors. For the sake of simplicity form errors are assumed to be negligible, but their inclusion can be straightforward when using small jaws or locators by treating form tolerances as an independent fixture error on each locator/jaw, as it is explained in [65]. If surfaces are used for locating, e.g. vice jaws, it can be considered that form errors have little or no effect on the result, as a feature's form tolerance is always smaller that its location/orientation tolerance. Form errors will be only significant if the surfaces in contact present an specific shape and the high points of both surfaces are aligned and touch each other, which is very remote [66, Chapter 20].

## 3.3 Bench vices

Bench vices are common devices for holding workpieces on a milling machine table. Among bench vices, plain vices are probably the most widely used fixture device in shop-floor. A plain vice has two jaws, one fixed and one movable, and the workpiece is held by the force exerted from the movable jaw to the fixed one with a pull-down action. Although the jaws are usually plain, special jaws with irregular shape are sometimes used to hold non-prismatic parts. Figure 3.4a shows a typical bench vice for milling where the components of the vice (jaws, supports and pins) and the fixture coordinate system (FCS) are identified; Figure 3.4b shows the workpiece held in the vice and the datums CS: A-CS, B-CS and C-CS.




**(a)** Typical bench vice and definition of the FCS.

**(b)** Workpiece held in the vice and datum coordinate systems.

**Figure 3.4.** Fixture and datum coordinate systems in a bench vice.

Following the methodology proposed in [10], the relationship between the errors of vice surfaces (fixture-induced errors) and the deviation of the FCS is defined by the matrix $\mathbf{A}_k^3$, and the relationship between the errors of datum surfaces and the position of the FCS (datum-induced errors) is defined by matrix $\mathbf{A}_k^2$. However, it should be noted that the final assembly fixture-workpiece depends on the position and orientation deviation of the fixed jaw, the support surface and primary and secondary datums and thus, the superposition of fixture and datum errors as presented in previous researches cannot be straightforward applied. In other words, the values of both matrices $\mathbf{A}_k^2$ and $\mathbf{A}_k^3$ should be expressed as a function of the interaction between current position and orientation of datum and locating surfaces.

### 3.3.1 Fixture-induced errors

To understand the FCS deviation due to fixture errors, let us explain the clamping process in a plain vice. First, the workpiece is placed over the support which makes the support surface and primary datum to be coplanar. Then, the workpiece is moved over the support to touch first the primary datum with the fixed jaw and then, to touch the tertiary datum with the pin of the vice. Finally, the movable jaw moves until it clamps the part, exerting a force towards the fixed jaw and the support due to its pull-down action. Due to this clamping force, the primary datum

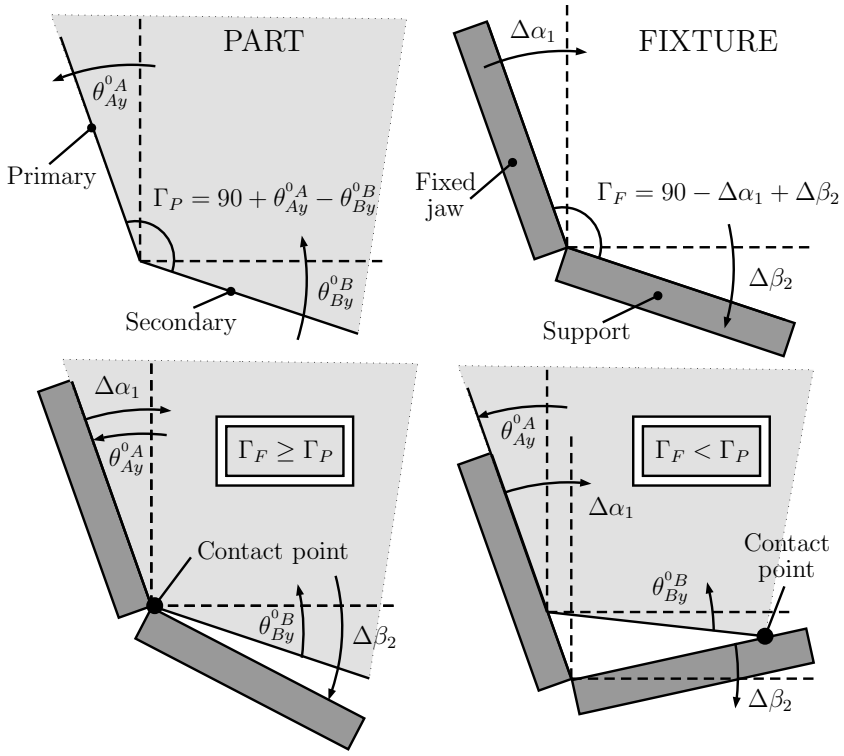



**Figure 3.5.** Different part location due to different fixture and datum errors. The arrows indicate the positive direction of each angle.

will move to be coplanar with the fixed jaw which may cause that the secondary datum lifts from the support. Therefore, the fixed jaw blocks three degrees of freedom (DOF), two rotations and one translation; the support blocks other two DOF, one rotation and one translation, and the pin blocks the remaining DOF.

The fixturing errors in vice fixtures can be defined as

$$\mathbf{u}_k^F = [\Delta z_1, \Delta\alpha_1, \Delta\beta_1, \Delta z_2, \Delta\alpha_2, \Delta\beta_2, \Delta z_3]^T, \tag{3.5}$$

where $\Delta z_1, \Delta\alpha_1, \Delta\beta_1$ refer to the Z-axis deviation and orientation deviations around X and Y axis of the fixed jaw (1-CS), respectively; $\Delta z_2, \Delta\alpha_2, \Delta\beta_2$ refer to similar deviations but from the support (2-CS); and $\Delta z_3$ refers to the deviation of the locating pin of the vice fixture (3-CS). Therefore, the resulting deviation of the FCS due to fixture errors is defined as

$$\mathbf{x}_{0_F}^F = \mathbf{A}_k^3 \cdot \mathbf{u}_k^F, \tag{3.6}$$

where matrix $\mathbf{A}_k^3$ can be estimated as follows.

As stated above, in a vice workholding system there is a plane to plane contact between the fixed jaw and the primary datum. Thus, 3 DOF are blocked by the fixed jaw and the errors $\Delta z_1$, $\Delta\alpha_1$, and $\Delta\beta_1$ are directly propagated to the FCS in these DOF. Therefore

$$\delta\mathbf{H}_F^{0F} =^0 \mathbf{H}_1^F \cdot \mathbf{H}_1^{01} \cdot \mathbf{H}_F^1, \tag{3.7}$$

where

$$\delta\mathbf{H}_F^{{}^0F} = \begin{bmatrix} 1 & -\theta_{Fz}^F & \theta_{Fy}^F & d_{Fx}^F \\ \theta_{Fz}^F & 1 & \theta_{Fx}^F & d_{Fy}^F \\ -\theta_{Fy}^F & \theta_{Fx}^F & 1 & d_{Fz}^F \\ 0 & 0 & 0 & 1 \end{bmatrix} = \mathbf{I}_{4\times4} + \boldsymbol{\Delta}_F^{{}^0F}, \tag{3.8}$$

and

$$\boldsymbol{\Delta}_F^{{}^0F} = \begin{bmatrix} \hat{\boldsymbol{\theta}}_F^{{}^0F} & \boldsymbol{d}_F^{{}^0F} \\ \mathbf{0} & 0 \end{bmatrix}. \tag{3.9}$$

In Eq. (3.7), the HTM $\mathbf{H}_F^1$ is equal to ${}^0\mathbf{H}_F^1$ since there is a plane to plane contact and the movements along $X$ axis and rotations around $Y$ and $Z$ axis of FCS are blocked. Then, considering $\delta\mathbf{H}_{{}^0F}^F = (\delta\mathbf{H}_F^{{}^0F})^{-1} = \mathbf{I}_{4\times4} - \boldsymbol{\Delta}_F^{{}^0F}$ and $\mathbf{x}_{{}^0F}^F = [d_{Fx}^F, d_{Fy}^F, d_{Fz}^F, \theta_{Fx}^F, \theta_{Fy}^F, \theta_{Fz}^F]^T$, Eq. (3.7) is solved to obtain the values of $\mathbf{x}_{{}^0F}^F(1)$, $\mathbf{x}_{{}^0F}^F(5)$ and $\mathbf{x}_{{}^0F}^F(6)$ as a function of $\Delta z_1$, $\Delta\alpha_1$, and $\Delta\beta_1$.

The secondary datum blocks other two DOF, the movement of the part along the $Z$ axis of FCS and the rotation of the part around the $X$ axis of FCS. Furthermore, since the plane to plane contact is given at the primary datum, the secondary datum and the support touch each other at least in two points. Denoting the two contact points as $\mathbf{p}_D$ and $\mathbf{p}_E$, where $\widetilde{\mathbf{p}} = [\mathbf{p}^T, 1]^T$, we have

$$\begin{aligned} \widetilde{\mathbf{p}}_D^F &= \mathbf{H}_{{}^01}^F \cdot \mathbf{H}_2^{{}^01} \cdot \widetilde{\mathbf{p}}_D^2 \\ &= \delta\mathbf{H}_{{}^0F}^F \cdot \mathbf{H}_{{}^01}^{{}^0F} \cdot \mathbf{H}_{{}^02}^{{}^01} \cdot \mathbf{H}_2^{{}^02} \cdot \widetilde{\mathbf{p}}_D^2, \end{aligned} \tag{3.10}$$

and the same expression for $\widetilde{\mathbf{p}}_E^F$ holds. From Eq. (3.10) we know that $\widetilde{\mathbf{p}}_D^F(3) = \widetilde{\mathbf{p}}_E^F(3) = 0$ since the contact points define the location of the part in $Z$ direction of the FCS, and the orientation deviation along $X$ axis is the same as the orientation deviation of the support which blocks this DOF. Thus, Eq. (3.10) can be solved to relate $\mathbf{x}_{{}^0F}^F(3)$ and $\mathbf{x}_{{}^0F}^F(4)$ with the fixed jaw errors together with the support errors.

Finally, the locating pin blocks the movement of the part along the Y axis, thus, following the same procedure we have

$$\begin{aligned} \widetilde{\mathbf{p}}_G^F &= \mathbf{H}_{{}^01}^F \cdot \mathbf{H}_3^{{}^01} \cdot \widetilde{\mathbf{p}}_G^3 \\ &= \delta\mathbf{H}_{{}^0F}^F \cdot \mathbf{H}_{{}^01}^{{}^0F} \cdot \mathbf{H}_{{}^03}^{{}^01} \cdot \mathbf{H}_3^{{}^03} \cdot \widetilde{\mathbf{p}}_G^3, \end{aligned} \tag{3.11}$$

where $\widetilde{\mathbf{p}}_G$ is the contact point defined by the locating pin of the fixture and $\widetilde{\mathbf{p}}_G^F(2) = 0$. Eq. (3.11) can be solved to relate $\mathbf{x}_{{}^0F}^F(2)$ with the locating pin errors together with support and fixed jaw errors.

Following the steps shown above, the DMV $\mathbf{x}_{{}^0F}^F$ can be expressed as a function of fixture errors through matrix $\mathbf{A}_k^3$. For the workpiece and vice shown in Figure 3.4 the numerical solution of this matrix is:

$$\mathbf{A}_k^3 = \begin{bmatrix} -1 & t_{1z}^F & -t_{1y}^F & 0 & 0 & 0 & 0 \\ 0 & 0 & t_{3x}^F & 0 & -t_{3z}^F & 0 & -1 \\ 0 & -a & 0 & -1 & t_{2y}^F & a - t_{2x}^F & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{3.12}$$

where $t_{1y}^F$ and $t_{1z}^F$ refer to the location of the fixed jaw CS w.r.t. the FCS, $t_{2x}^F$ and $t_{2y}^F$ refers to the location of the support CS w.r.t. the FCS, and $t_{3x}^F$, $t_{3z}^F$ refer to the location of the locating pin CS w.r.t. the FCS. Parameter $a$ depends on the fixture and datum assembly and resulting contact points $\mathbf{p}_D$ and $\mathbf{p}_E$. As it is shown in Fig. 3.5, the contact points between part and fixture depend on orientation deviations of support and fixed jaw and orientation deviation of primary and secondary datums. For the example given in Fig 3.4, $a$ has the following values:

$$a = \begin{cases} 0, & \text{if } \Gamma_F \geq \Gamma_P, \\ L_s, & \text{otherwise}, \end{cases} \tag{3.13}$$

where $L_s$ is the length of the contact between support and workpiece, $\Gamma_F = 90^\circ + \Delta\beta_1 - \Delta\alpha_2$ and $\Gamma_P = 90^\circ - \theta_{Ay}^{0A} - \theta_{Bx}^{0B}$.

For practical purposes, it is of interest to relate the fixture errors with the technical specifications of the vice. From common technical specifications, we may remark accuracy in clamping repeatability and parallelism and perpendicularity specification of vice surfaces. These geometrical specifications can be translated to DMV limits in the $\mathbf{u}_k^F$ parameters as shown in [65], and thus, the estimation of part quality variability for a given bench vice can be conducted. Therefore, considering $L_v$ and $H_v$ as the length and height of the fixed jaw, respectively; $W_s$ as the contact width between support and part; and $\epsilon_c$, $\epsilon_{pa}$ and $\epsilon_{pe}$ as clamping accuracy and parallelism and perpendicularity of vice surfaces, we have: $|\Delta\alpha_2| \leq \epsilon_{pa}/W_s$; $|\Delta\beta_2| \leq \epsilon_{pa}/L_s$; $|\Delta z_1| \leq \epsilon_c/2$; $|\Delta\alpha_1| \leq \epsilon_{pe}/H_v$; $|\Delta\beta_1| \leq \epsilon_{pe}/L_v$; and their relationships are defined as:

$$L_v \cdot |\Delta\beta_1| + H_v \cdot |\Delta\alpha_1| \leq \epsilon_{pe}. \tag{3.14}$$

$$L_s \cdot |\Delta\beta_2| + W_s \cdot |\Delta\alpha_2| \leq \epsilon_{pa}, \tag{3.15}$$

Additionally, we may add the alignment error of the vice in the machine-tool as $\epsilon_{alig}$ and the position error of the vice on the machine-tool table during the setup process (e.g., touch probe inaccuracy) as $\epsilon_{stp}$. Therefore, we have $|\Delta z_2| \leq \epsilon_{stp}$, $|\Delta z_1| \leq \epsilon_c/2 + \epsilon_{stp}$ and $\Delta\beta_1$ previously defined will add the alignment error $\epsilon_{alig}$.

Another common vice configuration is presented when the location of the workpiece in the parallel direction of the jaws is undefined so the pin locator is removed from the workholding device. Under this configuration, the possible deviation of machined features along this direction is undetermined, and $t_{3x}^F$ and $t_{3z}^F$ from matrices $\mathbf{A}_k^2$ and $\mathbf{A}_k^3$ are replaced by $U$ which refers to an undetermined component. To operate with $U$, the following properties apply:

$$\forall b \in \mathbb{R}, b + U = U; \forall b \in \mathbb{R}, b \cdot U = U. \tag{3.16}$$

### 3.3.2 Datum-induced errors

Considering the primary datum as the reference coordinate system (RCS) of the workpiece, the deviation of the FCS w.r.t. RCS is modeled by the DMV $\mathbf{x}_F^R$ and it can be defined as [10]

$$\mathbf{x}_F^R = \mathbf{T}_1 \cdot \mathbf{x}_2^R + \mathbf{T}_2 \cdot \mathbf{x}_3^R = \mathbf{A}_k^2 \cdot \left[\mathbf{x}_2^R \ \mathbf{x}_3^R\right]^T, \tag{3.17}$$

where $\mathbf{x}_2^R$ and $\mathbf{x}_3^R$ are the DMV that define the deviations of the secondary and tertiary datums of the workpiece which correspond with the workpiece surfaces that touches the support and the locating pin, and $\mathbf{A}_k^2 = [\mathbf{T}_1 \ \mathbf{T}_2]$. Following the procedure presented in [10,63], the matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ for the vice and workpiece shown in Figure 3.4 are defined as

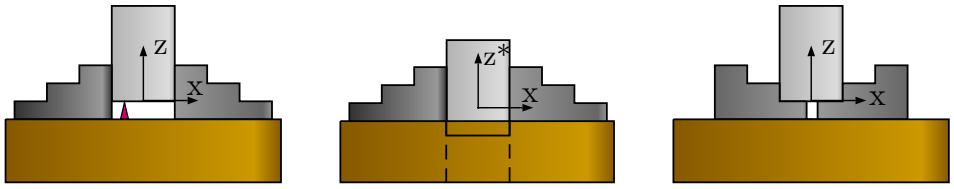
$$\mathbf{T}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -t_{3z}^F & 0 & 0 \\ 0 & 0 & -1 & -t_{Fy}^B & (a - t_{Fx}^B) & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ & & & \mathbf{0}_{2\times 6} & & \end{bmatrix}, \tag{3.18}$$

$$\mathbf{T}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & (t_{3z}^F - t_{Fy}^C) & (t_{Fx}^C - t_{3x}^F) & 0 \\ & & & \mathbf{0}_{4\times 6} & & \end{bmatrix}, \tag{3.19}$$

where $t_{Fx}^B$ and $t_{Fy}^B$ are the X and Y coordinate of FCS w.r.t. B-CS, respectively, $t_{Fx}^C$ and $t_{Fy}^C$ are the same but w.r.t. the C-CS, $t_{3x}^F$ and $t_{3z}^F$ refers to the position of the locating pin of the vice and the parameter $a$ depends on fixture and workpiece assembly (Fig. 3.5) and presents the values shown in Eq. (3.13). Please, refer to Appendix A.1 for the derivation details of $\mathbf{T}_1$ and $\mathbf{T}_2$.

## 3.4 3-jaw self-centering chucks

A 3-jaw self-centering chuck is a workholding device used in turning and milling processes to hold regular-shaped parts such as cylinders. This type of chuck consists of a cylindrical base with three slots carved from the center to the exterior, separated 120° from each other. There is a jaw in each slot, and all three jaws slide simultaneously by the same amount if one of the three pinions is rotated. A 3-jaw chuck can present different configurations depending on the main locating surfaces used. Figure 3.6 shows three configurations analyzed in this chapter. In the first configuration, the main locating surface is the outer diameter of the workpiece and thus, the jaws block 4 DOF whereas a pin locator blocks the Z movement of the part. The second configuration is similar to the previous one but no locating pin is used, so the position of the workpiece in Z direction is undetermined. The third configuration uses the end flat surface of the workpiece as the main locating surface which blocks 3 DOF (the Z movement and two rotations) due to the contact with the jaws. In this configuration, the clamping process locates the part in X and Y direction. In all cases, rotation around the Z-axis is limited by the friction of the workpiece and the jaws.




Configuration 1
Locating surface:
cylindrical face

Configuration 2
Locating surface:
cylindrical face
[z* direction not constrained]

Configuration 3
Locating surface:
end face

**Figure 3.6.** Typical configurations of a 3-jaw self-centering chuck and definition of the FCS.

### 3.4.1 Fixture-induced errors

Some researches have studied the errors of 3-jaw chucks and the methods to improve chuck accuracy [64, 67, 68]. The main identified errors in 3-jaw chucks are: radial displacement error of individual jaws due to internal wear or backlash; taper in jaw alignment; non-symmetric deformation of jaw-workpiece and kinematic redundancy. In this chapter, we consider that the results of those fixture errors are reflected in the deviation of the jaws from their nominal position and orientation. Therefore, we consider as fixture errors the position deviation of each jaw in the radial direction, the position error of the locating pin or the jaw to place the end face of the workpiece, and the orientation error due to jaw alignment.

For any of the chuck configurations defined above, the position of the workpiece in X and Y axis is defined by the position deviation of each jaw in the radial direction. Considering that the jaws are placed perpendicular to the slots and the chuck base, the top-down view of a chuck holding a perfect cylinder (XY view) can be defined as in Figure 3.7. Points P, Q and R (the contact point of the jaws) are separated a distance G from the center plus a jaw error $\delta$, expressed outwards the center. Thus, the deviation of the jaws from nominal positions are denoted as $\delta_P$, $\delta_Q$ and $\delta_R$. The deviation of the center of the workpiece clamped with respect to the center of the chuck is estimated to be 2/3 of the jaw deviation along the direction of jaw deviation, as it is shown in the Appendix A.2.

Furthermore, the workpiece may be deviated from the Z axis if the chuck constrains this direction as in configuration 1 and 3. In configuration 1, the main locating surface is the cylindrical surface of the workpiece and an orientation deviation of the jaws will produce a Z deviation of the part when the end face and the locating pin of the chuck touch each other. In configuration 3, the main locating surface is the end face of the workpiece, and the Z deviation will depend on the position and orientation deviation of the plane defined by the 3 jaws that contact with the end face. In any case, we represent the Z deviation as $\delta_z$, and the orientation deviation of the chuck as $\delta_\alpha$ and $\delta_\beta$, which reproduces the same inclination of the FCS from its nominal position.

Therefore, the fixture errors in a 3-jaw chuck can be defined as

$$\mathbf{u}_k^F = [\delta_P \ \delta_Q \ \delta_R \ \delta_z \ \delta_\alpha \ \delta_\beta]^T , \tag{3.20}$$

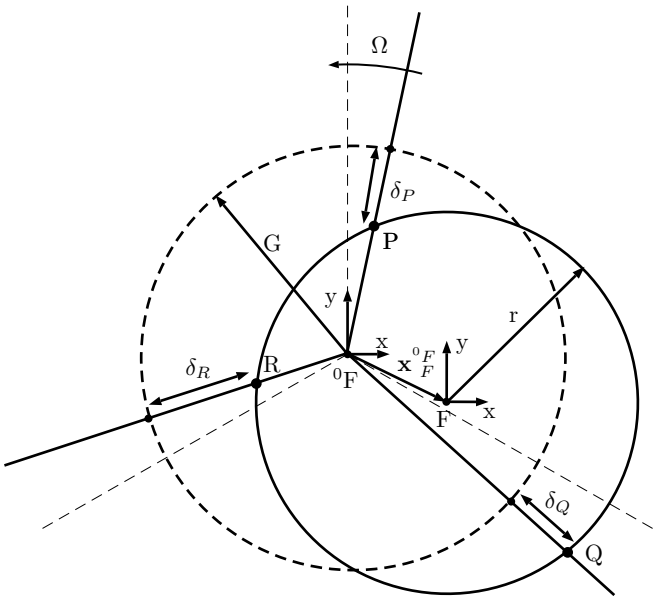



**Figure 3.7.** Deviation of the FCS due to self-centering errors. Errors are exaggerated for illustrative purposes.

and the corresponding matrix $\mathbf{A}^3_k$ can be defined as $[\mathbf{A}^3_{k(1)}\ \mathbf{A}^3_{k(2)}]^T$, where

$$\mathbf{A}^3_{k(1)} = \begin{bmatrix} \frac{2}{3}\sin\Omega & -\frac{2}{3}\cos\Omega \\ -\frac{\sqrt{3}}{3}\cos\Omega - \frac{1}{3}\sin\Omega & -\frac{\sqrt{3}}{3}\sin\Omega + \frac{1}{3}\cos\Omega \\ \frac{\sqrt{3}}{3}\cos\Omega - \frac{1}{3}\sin\Omega & \frac{\sqrt{3}}{3}\sin\Omega + \frac{1}{3}\cos\Omega \\ \mathbf{0}_{3\times 2} \end{bmatrix}, \tag{3.21}$$

$$\mathbf{A}^3_{k(2)} = \begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1} \\ -\mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 1} \end{bmatrix}. \tag{3.22}$$

Please, note that if the configuration 2 applies, there is no control about the Z position of the workpiece and thus, $\Delta z_L$ is replaced by $U$, an undetermined component. From previous equations, the angle $\Omega$ has been included to take into account that the position of the jaws may be rotated from the FCS on the machine-tool table so jaw P may be not in the $+Y$ direction. Furthermore, note that if $\delta_P = \delta_Q = \delta_R$ then the center of the part is the same as the center of the chuck and thus, $\mathbf{x}^{^0F}_F(1) = \mathbf{x}^{^0F}_F(2) = \mathbf{0}$.

As it was presented in the vice, it is of interest to obtain the relation between the technical specifications about accuracy of the 3-jaw chuck and the identified fixture errors. Common technical specifications in chucks refer to maximum TIR (total indicator runout) values in radial and axial direction, as it is shown in Figure 3.8. As it has been shown, the deviation of the jaws will define the centering error which in turn produces a constant radial run-out defect when rotating a cylindrical part. For the configuration 3 (Fig. 3.8a), the radial TIR alongside the jaws, denoted as $TIR^r$, can be defined as two times the centering offset and thus, this accuracy term of the chuck can be represented as the deviation of the jaws, $\delta_P$, $\delta_Q$ and $\delta_R$, in a

range of $[0, \frac{3}{4} \cdot TIR^r]$. On the other hand, the axial TIR is related to the orientation deviation of the chuck defined by $\delta_\alpha$ and $\delta_\beta$ and the diameter of the tested part. Denoting $TIR^a$ as the axial TIR and $D_t$ the diameter of the part tested for the axial TIR, we have $|\delta_\alpha| \leq TIR^a/D_t$ and $|\delta_\beta| \leq TIR^a/D_t$, and the following relationship holds:

$$D_t \cdot \sqrt{\delta_\alpha^2 + \delta_\beta^2} \leq TIR^a. \tag{3.23}$$

For the configuration 1 (Fig. 3.8b), the radial TIR is measured at the length $L_t$ of the tested part. Similar to the configuration 3, the deviation of jaws are limited to a range of $[0, \frac{3}{4} \cdot TIR^r]$ but now, due to the effect of orientation deviations $\delta_\alpha$ and $\delta_\beta$ at the $L_t$ position of the dial indicator, we have $|\delta_\alpha| \leq TIR^r/(2L_t)$ and $|\delta_\beta| \leq TIR^r/(2L_t)$, and additionally, the following relationship holds:

$$4/3 \cdot \sqrt{(\delta_P - 0.5(\delta_Q + \delta_R))^2 + (c^* \cdot (\delta_Q - \delta_R))^2} + L_t \cdot \sqrt{\delta_\alpha^2 + \delta_\beta^2} \leq TIR^r, \tag{3.24}$$

where $c^* = cos(30°)$.

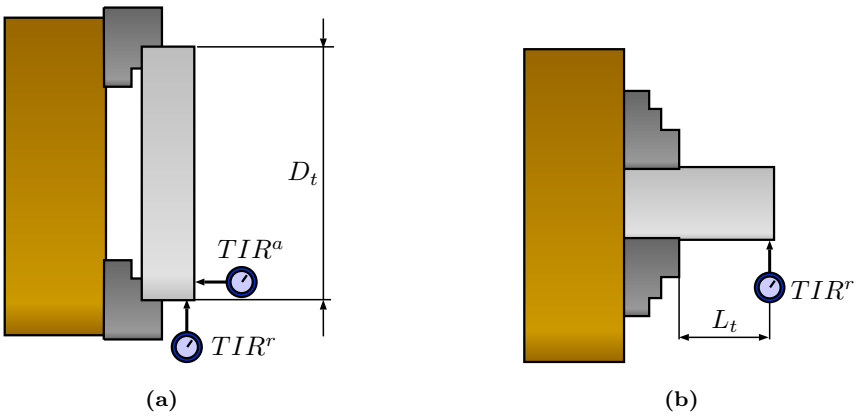



**Figure 3.8.** Common radial and end face (axial) runout used for test certifications in 3-jaw chucks with a) configuration 3, b) configuration 1 and 2.

### 3.4.2 Datum-induced errors

Datum-induced errors in 3-jaw chuck mainly depend on the chuck configuration. A chuck with configuration 2 presents only a primary datum, the cylindrical feature, and there is no secondary datum since there is no constraint over the Z position of the workpiece. Therefore, no datum-induced errors apply. Similarly, a chuck with configuration 3 presents the end face of the workpiece as the primary datum and the cylindrical feature is defined as the secondary datum. Since the chuck is a self-centering chuck and form errors are not considered, the center of the workpiece would be only defined by the jaw errors even though the cylindrical feature would present a orientation deviation, so no datum-induced errors apply. However, when the chuck presents the configuration 1, where a locating pin block the Z movement of the part and the primary datum is the cylindrical feature, a datum-induced error may arise as shown in Fig. 3.9.

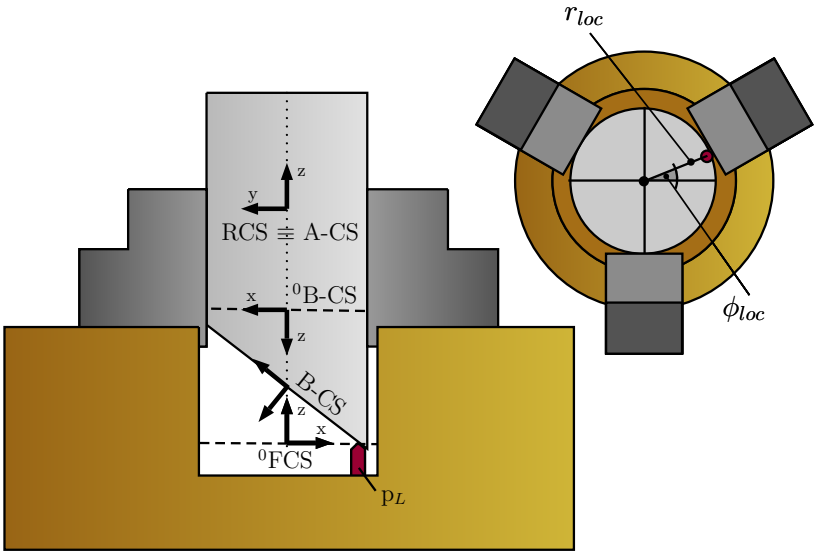



**Figure 3.9.** Effect of datum errors on part location.

As it can be seen, only the position along nominal Z-axis is modified due to datum errors since the cylindrical surface is oriented according to the 3-jaw orientation which is considered perfect when only datum-induced errors are analyzed. Then, following the methodology presented in [10], the deviation of the nominal FCS w.r.t. RCS, $\mathbf{x}_F^R$, can be obtained as

$$\mathbf{x}_F^R = \mathbf{T}_1 \cdot \mathbf{x}_2^R = \mathbf{A}_2 \cdot \mathbf{x}_2^R, \tag{3.25}$$

where $\mathbf{x}_2^R$ is the deviation of the secondary datum w.r.t. the part reference CS. The solving steps are detailed in Appendix A.3. Once solved, the deviation of the FCS w.r.t. RCS in Z axis direction is defined as

$$\mathbf{x}_F^R(3) = -d_{Bz}^R - p_{Ly}^F \cdot \theta_{Bx}^R - p_{Lx}^F \cdot \theta_{By}^R, \tag{3.26}$$

In matrix form, the final matrix $\mathbf{A}_k^2$ from Zhou's methodology can be expressed as

$$\mathbf{A}_k^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -p_{Ly}^F & -p_{Lx}^F & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{3.27}$$

Note that $\mathbf{p}_L^F$ is the position of the locating point w.r.t. FCS and it depends on the distance from the center of the chuck, $r_{loc}$, and the angle w.r.t. the X axis of FCS, $\phi_{loc}$, as it is shown in Fig. 3.9. Then, the X and Y coordinate w.r.t. FCS are

$$p_{Lx}^F = r_{loc} \cdot \cos(\phi_{loc}), \tag{3.28}$$

$$p_{Ly}^F = r_{loc} \cdot \sin(\phi_{loc}). \tag{3.29}$$

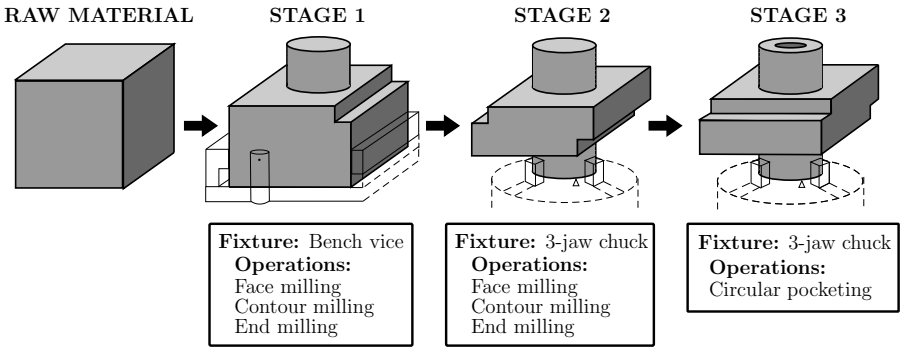



**Figure 3.10.** 3-stage machining process evaluated.

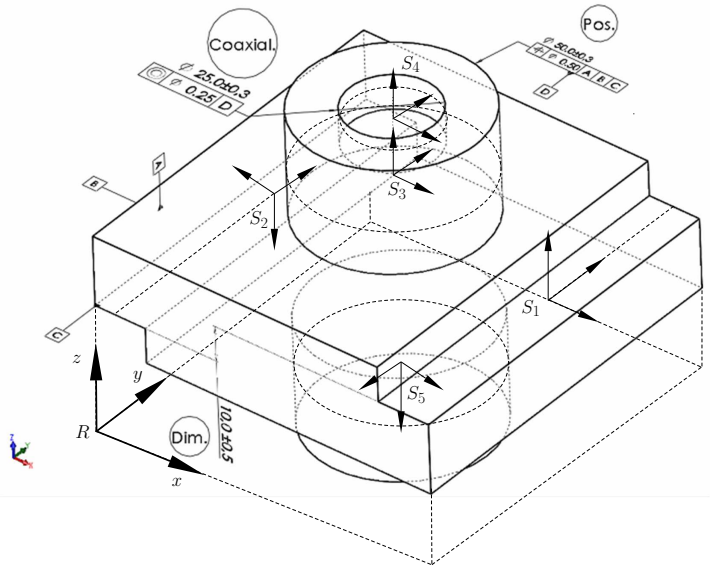



**Figure 3.11.** Part specifications to be inspected. Other dimensions are omitted for simplicity.

## 3.5 Case Study

In order to validate the extension of the SoV model, a 3-stage machining process is analyzed where both bench vices and 3-jaw chucks are used. As it is shown in Fig. 3.10, the manufacturing process consists of a face milling and end-milling operation at the first stage holding the part on a bench vice, a second stage where similar machining operations are conducted but using as fixturing device a 3-jaw chuck with a centered locator, and a third stage where the part is held on the same 3-jaw chuck in order to conduct a circular pocketing operation. As shown in Fig. 3.11, the inspected part specifications are: distance between both square-shoulder features, position of cylinder with respect to datums A, B, C, and the coaxiality of the circular pocketing with respect to its datum D. The position and orientation vectors for the main features are shown in Table 3.1. The raw material is an aluminum block with dimensions 100 x 100 x 100 mm whose surfaces have been premachined so flatness and square errors between surfaces can be assumed negligible.

The validation is conducted in two ways: 1) by using the SoV model to predict the deviations

**Table 3.1.** Position and orientation vectors of main feature CS.

| Feature | $\mathbf{t}^R$ | $\boldsymbol{\omega}^R$ |
|---|---|---|
| $S_1$ | $[92.5, 50, 55]^T$ | $[0, 0, 0]^T$ |
| $S_2$ | $[7.5, 50, 45]^T$ | $[0, \pi, 0]^T$ |
| $S_3$ | $[50, 50, 80]^T$ | $[0, 0, 0]^T$ |
| $S_4$ | $[50, 50, 90]^T$ | $[0, 0, 0]^T$ |
| $S_5$ | $[50, 50, 20]^T$ | $[\pi, \pi, 0]^T$ |

of the 3 geometrical specifications of the part and comparing these results with the resulting deviations obtained using a CAD software and; 2) by machining the part and comparing the results with those expected by the SoV and CAD model. The CAD software used is SolidWorks, and we basically model the fixture-workpiece assembly with the surface errors introduced in the tested cases to check the final part deviation. This is a tedious and time-consuming procedure that can be used for checking the effect of few errors at the same time assuming the rest negligible, and it cannot be used for checking the manufacturing process capability. For the machining experimentation, the machining center used for the experimentation is a Deckel Maho DMC 70V machining center, and the features are inspected in a Brown & Sharpe Mistral 775 coordinate measuring machine. The vice used is a Fresmak Arnold Twin with 0.02 mm of parallelism and perpendicularity and 0.01 mm clamping accuracy, and the 3-jaw chuck is a Optimum K11-125 chuck model mounted according to configuration 1 with an inspected $TIR^r$ of 0.11 mm for a length of $L_t = 50$ mm. The setup process of the vice and 3-jaw chuck in the machine-tool table is conducted with a Renishaw touch probe and the alignment and positioning error is assumed to be $\epsilon_{alig} = 0.020$ mm/100 mm and $\epsilon_{stp} = \pm 0.015$ mm. A first part is machined to calibrate the process (e.g., tool dimensions, offsets due to clamping deformation, etc.). Figure 3.12 illustrates the experimental setup.

Three different situations are tested:

- i) no errors
- ii) relative small fixture errors
- iii) severe fixture errors

The fixture errors that were intentionally added, cases ii) and iii), are shown in Table 3.2. These errors were physically introduced by adding a feeler gauge between the workholding device and the workpiece or modifying the zero part coordinate system in the CNC maching-tool to get the same effect. Furthermore, the SoV model is applied in two ways. The first one considers the errors added into the process and assumes negligible any other errors. Conversely, the second one, named as "SoV + Monte Carlo", considers the technical specifications related to the accuracy of the workholding systems and simulates additional errors according to these specifications in order to calculate a range of values for each inspected specification. For this purpose, 5,000

**Figure 3.12.** Machining center, workholding devices and machined part from case study.

**Table 3.2.** Errors added in the multi-stage machining process. $\Omega$ is 0 in stages 2 and 3.

| Case | Stage 1 (Vice errors) | Stage 2 (Chuck errors) | Stage 3 (Chuck errors) |
|---|---|---|---|
| i | no errors added | no errors added | no errors added |
| ii | $\Delta z_1, \Delta z_2 = 0.1\ mm$<br>$\Delta z_3 = 0.1\ mm$ | $\delta_P = 0.2\ mm$<br>$\delta_z = 0.1\ mm$ | $\delta_P = 0.2\ mm$<br>$\delta_z = 0.1\ mm$ |
| iii | $\Delta\alpha_1 = -0.01\ rad$<br>$\Delta z_2, \Delta z_3 = 0.3\ mm$<br>$\Delta z_1 = 0.35\ mm$ | $\delta_P = 0.5\ mm$<br>$\delta_z = 0.3\ mm$ | $\delta_P = 0.5\ mm$<br>$\delta_z = 0.3\ mm$ |

Monte Carlo simulations were run and the range that comprises the 99.7% of the values was recorded.

The results are shown in Table 3.3. Firstly, the case i) shows the estimation of the range of values for the analyzed part specifications considering the accuracy technical specifications of the vice and the 3-jaw chuck used. These ranges are in fact the manufacturing process capability according to the workholding specifications and assuming no machining-induced errors exist. For this case study, the high TIR of the 3-jaw chuck used is reflected on the high expected coaxility error which is indeed confirmed in the experimentation. However, the specification related to the position error of the cylinder can be kept tight despite the bench vice inaccuracies, ensuring a position error less than 0.029 mm under Monte Carlo simulations and experimentally validated with a measured error of 0.035 mm. Secondly, for both small and severe errors added into the process, the proposed model shows a maximum error of 1% in comparison with the CAD results. The first specification (dimensional deviation between square-shoulder features), gives the same

**Table 3.3.** Validation results. Comparison between SoV model, CAD simulations and machined parts. Units in mm.

| | CAD | | | SoV | | |
|---|---|---|---|---|---|---|
| # | Dim. | Pos. | Coaxial. | Dim. | Pos. | Coaxial. |
| i | 10.000 | 0.000 | 0.000 | 10.000 | 0.000 | 0.000 |
| ii | 9.900 | 0.141 | 0.188 | 9.900 | 0.141 | 0.188 |
| iii | 9.700 | 0.392 | 0.471 | 9.700 | 0.390 | 0.471 |
| | SoV + Monte Carlo | | | Experimental (Machined parts) | | |
| # | Dim. | Pos. | Coaxial. | Dim. | Pos. | Coaxial. |
| i | $[9.974, 10.026]$ | $[0, 0.029]$ | $[0, 0.225]$ | 9.986 | 0.035 | 0.109 |
| ii | $[9.875, 9.927]$ | $[0.121, 0.164]$ | $[0.040, 0.404]$ | 9.868 | 0.154 | 0.358 |
| iii | $[9.675, 9.726]$ | $[0.370, 0.410]$ | $[0.312, 0.687]$ | 9.644 | 0.451 | 0.362 |

results between CAD and SoV model because, given the errors in Table 3.2, only the deviation in Z direction of the 3-jaw chuck at stage 2 has an impact and then, there is no error due to linearizations. However, the position specification shows the effect of linearization errors in the vice due to orientation deviations when comparing with respect to CAD results. According to the results, this error is around 1%.

Finally, the results from the SoV model considering the errors added and the accuracy specifications of the workholding systems are compared with the results obtained after machining and inspecting the parts. As it is shown in Table 3.3, the inspected specifications are in good agreement with the estimated range of values. The position specifications present a slightly higher values than the ones estimated through Monte Carlo which may be explained by machining-induced errors or deformation variations in the bench vice during clamping.

## 3.6 Conclusions

This chapter has shown how to extend the current SoV model in order to include general purpose workholding devices such as bench vices and 3-jaw chucks, not considered yet in the literature. In bench vices, the errors included in the model are position and orientation errors of plain jaws, supports and pins. In 3-jaw chucks, the errors included are the position error of jaws in the chuck, the position error of the locating pin to block the $Z$ direction of the workpiece and the orientation errors of the jaws. In all cases, the model assumes that the workpiece acts as a rigid part and errors due to deformation during clamping are assumed to be negligible in comparison with fixture- and datum-induced errors. The model has been validated on a 3-stage machining process through both CAD simulations and machining experimentation. The model performance with respect to CAD simulations showed an error of less than 1% due to linearization and the machining results validated the capability of the model to estimate 99.7% confidence intervals for different product specifications considering the accuracy of the workholding systems. Unlike previous extensions of the model, the proposed extension let practitioners apply zero-defect strategies in multi-stage machining processes where bench vices or chucks are used and it can also be used for estimating manufacturing process capability under specific workholding devices.

Chapter 4

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


## Abstract

In the current paradigm of zero-defect manufacturing, it is essential to obtain mathematical models that express the propagation of manufacturing deviations along the stages of Multistage Manufacturing Processes (MMPs). Linear physical-based models are commonly used, but its accuracy is reduced when applied to MMPs with a large amount of stages, due to the fact that small approximations made when modeling each individual stage are subsequently accumulated, and when applied to MMPs with complex components that cannot be reliably modeled.

In this chapter we propose a methodology to adjust a propagation model using measurements from the deviations of the Key Product Characteristics (KPCs) of the processed parts at the end of the MMP, as well as prior engineering-based knowledge, under the premise that the state of the variation sources of the process are unknown and must be estimated as part of the model adjustment. The proposed methodology consists of a recursive algorithm that minimizes the difference between the sample covariance of the measured 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.

## 4.1 Introduction

Multistage Manufacturing Processes (MMPs) are processes that require several stages to manufacture a product. Examples of MMPs include 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 [9, 10, 69, 70]. For instance, in the case of automobile assemblies, a typical body-in-white is composed of 100–150 sheet metal parts, which are assembled at between 80 and 120 stations where more than 1500 fixture locator are used to place the parts and more than 4000 welding points are executed [9]. In MMPs, workpiece dimensional errors are caused by sources of variation of the process, such as faults in fixtures or tool deterioration. These dimensional errors (also expressed as variations or deviations) are propagated along the following stages, affecting the output quality of the manufactured product.

In the current paradigm of zero-defect manufacturing, variation propagation reduction is crucial to improve the output quality of the manufactured products. However, the complexity of MMPs, due to the amount of stages, variation sources and the complexity of the interactions at each stage, makes quality assurance a challenging task. Current trends on zero-defect manufacturing promoted by the European Factories of the Future Research Association (EFFRA) and other institutions are encouraging engineers to develop strategies for modeling, monitoring and controlling the output quality of this type of processes [71].

MMP control is used to reduce variation in the output quality by active deviation compensation, which consists of modifying the behavior of downstream stages to correct deviations (or faults, depending on the magnitude and duration) that have been detected in upstream stages [72], or using quality rework loops (whose impact on the system is evaluated in [13]). These techniques require flexible manufacturing, optimal measurement sensor location [73] and appropriate models of the manufacturing system, which must be analyzed to verify its diagnosability [14] and compensability [15]. Thus, in order to be able to compensate deviations downstream, it is important to monitor and estimate the process variance [56], and detect, isolate and identify any faults caused by the process variation sources, as well as evaluate and monitor the deterioration of the process components [11,12]. In the literature [74], fault diagnosis methods are classified in estimation-based methods, pattern matching methods and artificial intelligence methods. Additionally, knowledge-based methods are being developed in the latest years [75].

Estimation-based methods use a defined linear model and collected measured variables to estimate the variance of the state of the variation sources and, using statistical methods, isolate and identify the faulted variation source. These methods have been applied to identify faults in single-stage assembly processes using geometrical models [76], using least squares on ill-conditioned multistation assemblies with compliant parts [77], and in general manufacturing processes using linear mixed models [78].

Pattern matching methods do not require precise linear models, although they may rely on general linear models or engineering knowledge. These methods focus on comparing expected

fault patterns, obtained from those models or knowledge, with the measurement data to identify the faulty variation source. These methods have been applied in assemblies, using CAD and Principal Component Analysis (PCA) [79], and in multistage assembly processes using PCA and the variation propagation model as a basis, in order to detect single faults [80]. These methods can be extended for multiple fault identification through fault space diagnosis [81].

Artificial intelligence methods focus on developing networks to identify fault patterns. These methods do not require models, although they may use engineering knowledge; however, they always need large amounts of process and inspection data. Several examples of these methods include bayesian networks in assembly processes [82] and neural networks in die-casting [83].

Estimation-based methods and some of the pattern matching methods use linear models for detecting faults in both single and multistage processes. Moreover, in the case of the estimation-based methods, their efficiency depends on the accuracy of the used model. During the last decades, many studies have developed several models to define the effect of the process variation sources on the output quality of the processed products. Depending on the methodology, the proposed models can be classified into data-driven models and physical (or analytical) models [84, 85].

Data-driven models do not require detailed a priori knowledge of the process, as these methods focus on identifying patterns using collected data to determine the internal interactions of the process. Some examples include obtaining models using blind source separation methods [86], which do not use any knowledge from the process; or using general knowledge of the process, such as integrating graphical models and statistical techniques [87]. A mixed approach is proposed in [88], where engineering knowledge (derived from analytical models) is converted into a qualitative representation matrix in order to identify spatial patterns. Data-driven methods are used as a basis for fault diagnosis, but they do not grant a detailed explanation of the internal process behavior.

Physical models are developed on the basis of the physical and geometrical principles that define the process. During the last two decades, the Stream-of-Variation (SoV) methodology has been developed and expanded to model and reduce variation in MMPs. The SoV methodology defines the variation propagation model as 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 and machining errors) and on the deviations of certain features manufactured in previous stages (named datum errors). These relationships are expressed in matrix form, and can be obtained using different methods, such as differential motion vectors (DMVs), equivalent fixture error (EFE) and kinematic analysis (KA) [89]. Besides, the state space equation can be rearranged into a linear input-output model that directly relates the output quality with the variation sources, which can be easily used in fault diagnosis techniques.

This methodology was first developed to model the behavior of rigid sheet metal assembly processes, defining the main types of errors [90], which was later extended to compliant parts [91], and extended afterwards to 3D using DMVs [49]. The behavior of compliant composite parts for single and multistage assembly processes has also been modeled [92, 93]. Multistage machining processes have also been modeled. Originally defined using DMVs in [10], subsequent research

has modeled the effects of machining-induced variations [94] and general-purpose workholding devices [17]. The effects of general fixture layouts using kynematic analysis have also been researched [95]. The SoV methodology has also been applied in MMP design to reduce dimensional variability [57] and thus, potential defects.

Physical models offer a detailed explanation of the behavior of the process components, with the immediate drawback that rigorous research of the mechanics, dynamics and geometries of the process is required to develop these models. There is an additional drawback of physical models. Usually, developing these models requires linearizations and approximations that can frequently be ignored due to their low relative magnitude. However, in MMPs with a large amount of stages, these linearization-induced errors accumulate when calculating the input-output model, thus lowering its accuracy. In other cases, linearizations are only valid in certain dimensional ranges, thus leading to model approximations in certain configurations. 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 w.r.t. the real behavior of the process. To overcome this limitation, the physical model can be adjusted using data from the process and thus, a more accurate model can be obtained [96], which, in turn, improves product quality [97].

In this chapter, we present a methodology to reduce the aforementioned divergences by slightly adjusting a physical variation propagation model of an MMP (in the linear input-output form) 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 non-linear 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 chapter 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 the convex optimization solver can provide a solution within a finite time.

This chapter is structured as follows. Section 4.2 presents the propagation model and the process data collection. Section 4.3 defines the error function, the objective function and the constraints. Section 4.4 presents the proposed adjustment algorithm. Section 4.5, proposes several indexes to validate the performance of the algorithm. Section 4.6 proposes a case study, and Section 4.7 presents the conclusions of this chapter.

### 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 $i$th processed part, respectively. $a_{k,n}$ refers to the state of $A$ for the $n$th locator or KPC deviation at stage $k$. Letter $\Sigma$ represents a covariance matrix, and $\sigma^2$ an element of that matrix. Also, letter $S$ represents a sample covariance matrix. Letters $\Sigma$, $\sigma^2$ and $S$ can be accompanied by a subscript (e.g. $\Sigma_z$), which refers to a given assigned term $z$. Thus, $\sigma_a^2[q]$ refers to the variance of the $q$th 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 $\frac{\partial b}{\partial a}$.

## 4.2 Problem statement

The objective of this chapter is to present a methodology to adjust a physical linear input-output model of an MMP 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.

We assume that data from the variation sources are not available, although approximated ranges of their covariance are available. We also assume that the aforementioned model divergences are modeled as disturbances expressed as linear functions of the variation sources. Lastly, we assume that the data is collected from a faultless process; thus, no other disturbances are considered in this methodology.

The output of the proposed adjustment methodology will be an adjusted variation propagation matrix and the estimation of the covariance of the effect of the deviation of the variation sources. The variation propagation model and the proposed assumptions are explained in this section.

### 4.2.1 The variation propagation model

The Stream-of-Variation methodology defines the variation propagation model of an M-stage MMP as a state-space model [9]. 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

$$x_k(i) = A_{k-1} \cdot x_{k-1}(i) + B_k \cdot u_k(i) + w_k(i), \tag{4.1a}$$

$$y_k(i) = C_k \cdot x_k(i) + v_k(i), \tag{4.1b}$$

where $k = \{1,\ 2,\ ...,\ M\}$ refers to the stage index. The feature deviations of part $i = \{1,\ ...,\ \infty\}$ after stage $k$ are expressed by $x_k(i)$. The states of the variation sources are represented by $u_k(i)$, and $w_k(i)$ represents unmodeled errors of the process. Key Product Characteristic deviation (KPCd) measurements 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 characteristics.

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

$$y_M(i) = \Gamma \cdot u(i) + v_M(i) + \omega(i), \tag{4.2}$$

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 $\Gamma$ ($\Gamma \in \mathbb{R}^{n_y \times n_u}$) is the variation propagation matrix, which relates the impact of all variation sources on the KPC deviations using a linear relationship. Matrix $\Gamma$ is the result of the organized products of matrices $A$, $B$ and $C$ of each stage. This matrix defines the general behavior of the MMP.

*Note:* We have assumed that 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 $\Gamma$.

Vectors $y_M(i)$, $v_M(i)$ and $u(i)$ are then defined as

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

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

In the bibliography [98], the linear input-output model (4.2) is considered a stationary process. Matrix $\Gamma$ is assumed to remain constant during the natural working time of the MMP. Thus, equation (4.2) is generalized into

$$y = \Gamma \cdot u + v, \tag{4.3}$$

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

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

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

In Figure 4.1 we present a diagram that summarizes the different concepts we have exposed until now.

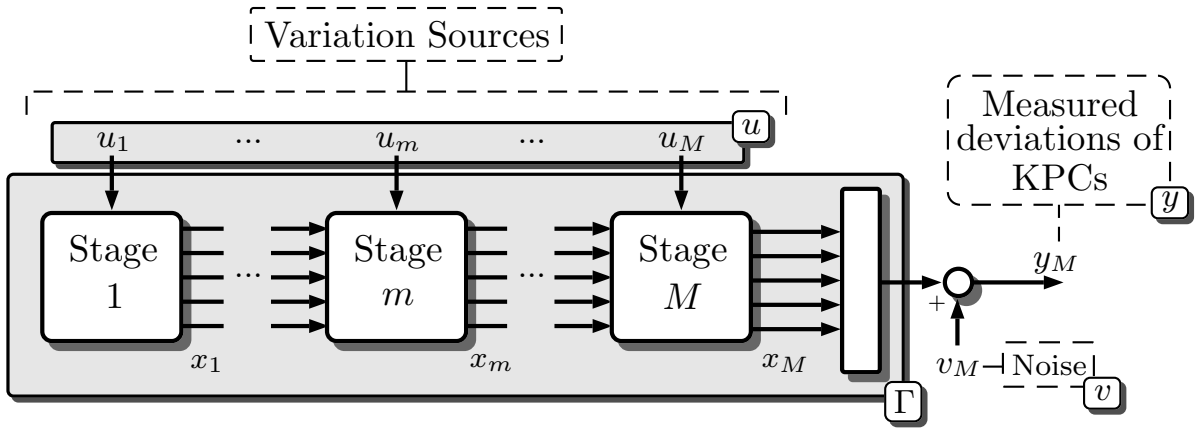



**Figure 4.1.** Diagram of the variation source propagation in a multistage process.

We assume that the expected value of the states of the variation sources $u$ and the measurement noise $v$ is zero. We also assume that $u$ and $v$ are independent variables:

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

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

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 $\Sigma_y$, $\Sigma_u$ and $\Sigma_v$ as

$$\begin{aligned} \Sigma_y &= \boldsymbol{E}\{y \; y^{\intercal}\}, \\ \Sigma_u &= \boldsymbol{E}\{u \; u^{\intercal}\}, \\ \Sigma_v &= \boldsymbol{E}\{v \; v^{\intercal}\}, \end{aligned}$$

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

$$\Sigma_y = \Gamma \; \Sigma_u \; \Gamma^{\intercal} + \Sigma_v. \tag{4.5}$$

The stationary process assumption we considered before implies that the aforementioned variances will be constant during the normal operation of the multistage process. Thus, we assume that there are no faults in the process when conducting the proposed adjustment methodology.

We have also made the following assumptions:

- The variation sources are assumed to be independent, so covariance matrix $\Sigma_u$ is diagonal:

$$\Sigma_u = \begin{bmatrix} \sigma_u^2[1] & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_u^2[n_u] \end{bmatrix}, \tag{4.6}$$

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

$$\vec{\Sigma}_u = \left[\sigma_u^2[1],\ ...,\ \sigma_u^2[n_u]\right]^\intercal \equiv diag\left(\Sigma_u\right). \tag{4.7}$$

- Measurement noises are assumed to be independent. Thus, covariance matrix $\Sigma_v$ is diagonal:

$$\Sigma_v = \begin{bmatrix} \sigma_v^2[1] & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_v^2[n_y] \end{bmatrix}, \tag{4.8}$$

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

$$\vec{\Sigma}_v = \left[\sigma_v^2[1],\ ...,\ \sigma_v^2[n_y]\right]^\intercal \equiv diag\left(\Sigma_v\right). \tag{4.9}$$

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

$$\vec{\Sigma}_y \equiv diag\left(\Sigma_y\right). \tag{4.10}$$

  Attending to Appendix B.1, we can state the following relationship for the diagonal elements of the covariance of the KPCd measurements.

$$\vec{\Sigma}_y = \Gamma^{\circ 2}\vec{\Sigma}_u + \vec{\Sigma}_v. \tag{4.11}$$

### 4.2.2 Process data collection

Collected data from the MMP 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 $Y \in \mathbb{R}^{n_y \times N}$:

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

The sample covariance of $Y$ is calculated using the covariance formula:

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

We call $\vec{S}_y$ to the variable that contains the same terms of the diagonal of $S_y$.

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

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

### 4.2.3 Engineering-based assumptions

Physical models in the Stream-of-Variation methodology focus on detailing the behavior of the variation propagation in MMPs by defining matrices $A_k$, $B_k$ and $C_k$ from (4.1). However, due to the physical complexity of the components of an MMP, it is common to apply approximations and linearizations during the development of these models. Additionally, physical models only take into account the behavior of ideal components of the MMPs; real components may present slight differences.

These errors are frequently small in magnitude, and are often ignored. However, as matrix $\Gamma$ represents a reorganized product of those matrices, in MMPs with a large amount of stages some of these errors get accumulated. Thus, the accuracy of the linear input-output model obtained exclusively using physical models is reduced when the number of stages of the MMP increases. In other cases, due to the complexity of some of the process components, linearizations are only valid in certain dimensional ranges, so the performed approximations are more noteworthy, thus also lowering the accuracy of the obtained input-output model. We call $\Gamma_0$ to this initial estimation of $\Gamma$, which has been obtained through physical models.

The internal elements of $\Gamma_0$ ($\Gamma_0 \in \mathbb{R}^{n_y \times n_u}$) present the following form:

$$\Gamma_0 = \begin{bmatrix} \Gamma_0[1,1] & \dots & \Gamma_0[1,n_u] \\ \vdots & \ddots & \vdots \\ \Gamma_0[n_y,1] & \dots & \Gamma_0[n_y,n_u] \end{bmatrix}. \tag{4.12}$$

In this chapter we assume that, due to approximations and differences between idealized and real components, $\Gamma_0$ will notably differ from $\Gamma$.

Additionally, we assume that a certain range of the covariance of the variation sources $\Sigma_u$ is available, obtained using backup data or from vendor's specifications (e.g. accuracy of fixture locators). We also assume that we know the theoretical covariance of the measurement noise $\Sigma_v$, which is directly related to the precision of the measuring instrument and thus, obtainable from the instruments' specifications. The order of magnitude of $\Sigma_v$ is assumed to be notably lower than the expected values of $\Sigma_y$, as required to perform faithful measurements.

Thus, we can obtain a proposal for the elements of the variance of the variation sources, expressed as $\vec{\Sigma}_u$ (4.7). Using equation

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

which is based on equation (4.11), we can then obtain

$$\vec{\Sigma}_u = \left(\Gamma_0^{\circ 2^\intercal}\Gamma_0^{\circ 2}\right)^{-1}\Gamma_0^{\circ 2^\intercal}\left(\vec{S}_y - \vec{\Sigma}_v\right) \tag{4.13}$$

using Least Squares.

### 4.2.4 Problem formulation

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

- The MMP behaves as a linear input-output model (4.3).
- Independent random variables lead to a VVPM model (4.5).
- An initial estimation for $\Gamma$, called $\Gamma_0$ (4.12), is available and obtained using physical models.
- The variance of the variation sources ($\Sigma_u$) for model (4.5) are independent (4.6).
- The variances of the measurement noises ($\Sigma_v$) are available and independent (4.8).
- We have some engineering knowledge about the system that allows us to state some relationships between model parameters (such as backlog data and geometrical premises from $\Gamma_0$).
- A set of output data measurements $Y$ is available.

We want to obtain both an estimate for $\Gamma$ and $\Sigma_u$ for model (4.5). This model can be used latter, for instance, to reconstruct an estimation of the KPCd measurement covariance matrix $\Sigma_y$ for its use in fault diagnosis. We call those estimations $\hat{\Gamma}$ and $\hat{\Sigma}_u$, which we obtain using the sample covariance $S_y$ from data set $Y$ and $\Gamma_0$:

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

Both $\hat{\Gamma}$ and $\hat{\Sigma}_u$ are used to obtain the KPCd measurement covariance matrix estimate $\hat{\Sigma}_y$.

In Figure 4.2, we present a summary diagram of the problem formulation.

The problem can be stated as an optimization problem that reads as follows. Obtain $\hat{\Gamma}$ and $\hat{\Sigma}_u$ that minimizes some metric that measures the error between $\Sigma_y$ and its estimation $\hat{\Sigma}_y$ through the model as

$$\hat{\Sigma}_y(\hat{\Gamma}, \hat{\Sigma}_u) = \hat{\Gamma}\hat{\Sigma}_u\hat{\Gamma}^\intercal + \Sigma_v,$$

subject to some given relationships between the elements (obtained through engineering knowledge) and using $S_y$ as an approximation for $\Sigma_y$.

In the following sections we address how to express a measure of the error between the output covariance $\Sigma_y$ and its reconstruction through $\hat{\Gamma}$ and $\hat{\Sigma}_u$, how to formulate its search through a tractable optimization problem, and how to evaluate the goodness of the model with the available data $Y$.

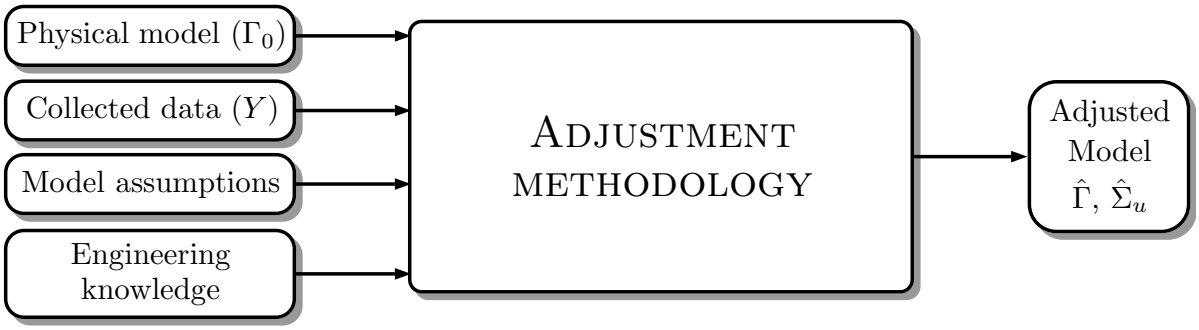



**Figure 4.2.** Diagram of the proposed problem formulation.

## 4.3 Formulation of the optimization problem

### 4.3.1 Evaluation of the estimation error

As stated before, matrix $\Sigma_u$ is assumed to be diagonal. In that sense, for the optimization problem we will consider as decision variables only the diagonal elements within it. We call $\vec{\Sigma}_u$ the vector that contains those diagonal elements, and $\hat{\vec{\Sigma}}_u$ its estimate (one of our decision variables). The internal elements of $\hat{\vec{\Sigma}}_u$ ($\hat{\vec{\Sigma}}_u \in \mathbb{R}^{n_u}$) are defined as

$$\hat{\vec{\Sigma}}_u = \left[\hat{\vec{\Sigma}}_u[1],\ ...,\ \hat{\vec{\Sigma}}_u[n_u]\right]^\intercal . \tag{4.14}$$

The estimation of matrix $\Gamma$, called $\hat{\Gamma}$, is our other decision variable. The internal elements of $\hat{\Gamma}$ ($\hat{\Gamma} \in \mathbb{R}^{n_y \times n_u}$) present the following form:

$$\hat{\Gamma} = \begin{bmatrix} \hat{\Gamma}[1,1] & \dots & \hat{\Gamma}[1,n_u] \\ \vdots & \ddots & \vdots \\ \hat{\Gamma}[n_y,1] & \dots & \hat{\Gamma}[n_y,n_u] \end{bmatrix}. \tag{4.15}$$

With this, the estimate of $\hat{\Sigma}_y$ ($\hat{\Sigma}_y \in \mathbb{R}^{n_y \times n_y}$) can be expressed in terms of the decision variables as

$$\hat{\Sigma}_y(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \hat{\Gamma} diag^{-1}(\hat{\vec{\Sigma}}_u)\hat{\Gamma}^\intercal + \Sigma_v, \tag{4.16}$$

where $\Sigma_v$ is assumed to be known.

The following matrix contains the difference between the KPCd measurement covariances and its estimate, which is expressed as $E$ ($E \in \mathbb{R}^{n_y \times n_y}$):

$$E(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \Sigma_y - \hat{\Sigma}_y(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \Sigma_y - \hat{\Gamma} diag^{-1}(\hat{\vec{\Sigma}}_u)\hat{\Gamma}^\intercal - \Sigma_v.$$

As we do not know $\Sigma_y$ the previous matrix error can be numerically approximated through

$$\hat{E}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = S_y - \hat{\Sigma}_y(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = S_y - \hat{\Gamma} diag^{-1}(\hat{\vec{\Sigma}}_u)\hat{\Gamma}^\intercal - \Sigma_v,$$

by using the sample covariance $S_y$. Here, $\hat{E} \in \mathbb{R}^{n_y \times n_y}$ and $S_y \in \mathbb{R}^{n_y \times n_y}$. As matrix $\hat{E}$ is symmetric, we first obtain a vector that gathers the difference of all unique matrix elements by using the *svec* operator, leading to

$$\hat{e}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = svec\left(\hat{E}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u)\right) = svec(S_y - \hat{\Gamma} diag^{-1}(\hat{\vec{\Sigma}}_u)\hat{\Gamma}^\intercal - \Sigma_v). \tag{4.17}$$

As $\hat{e}$ is the result of a symmetrical vectorization, $\hat{e} \in \mathbb{R}^{n_{Sy}}$ is a column vector, where $n_{Sy} = n_y(n_y+1)/2$ is the number of elements on and under the diagonal of $\hat{E}$.

Then, we propose to define a metric of the error by summing all the squared elements of the previous vector as

$$\|\hat{e}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u)\|_2^2 = \hat{e}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u)^\intercal \hat{e}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u).$$

Then, we can state initially the estimation problem as the solution to the following optimization problem

$$(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \arg \min_{\hat{\Gamma}, \hat{\vec{\Sigma}}_u} \|\hat{e}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u)\|_2^2, \tag{4.18}$$

with $\hat{e}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u)$ defined in (4.17). The function to be minimized in the previous optimization problem is a 6th order polynomial in the decision variables.

### 4.3.2 Definition of the engineering- and data-driven constraints

The minimization of the metric index in (4.18) must be subjected to appropriate constraints in order to guarantee a correct adjustment of the model. Here, we propose several constraints to be included, depending on the available engineering data.

The following constraints bound the calculations to obtain $\hat{\Gamma}$. They are detailed for any elements located on any $a$, $b$, $c$ and $d$ position in $\hat{\Gamma}$ (4.15):

1. **Sign of the elements of $\Gamma$.** 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.

$$C_1 : \hat{\Gamma}[a,b] \geq 0 \;||\; \hat{\Gamma}[a,b] \leq 0. \tag{4.19}$$

2. **Bounds of $\Gamma_0$.** The upper and lower limits for $\hat{\Gamma}$ can be defined as a variation of the approximated physical model $\Gamma_0$ (4.12) within some given deviations.

$$C_2 : \Gamma_0[a,b] + \mu_1 \leq \hat{\Gamma}[a,b] \leq \Gamma_0[a,b] + \mu_2. \tag{4.20}$$

   In practice, this means we assume that the physical model $\Gamma_0$ is, to some extent, close to $\Gamma$.

3. **Null values.** Elements of $\hat{\Gamma}$ can be constrained to zero if it is clear that some variation sources cannot affect the corresponding KPC deviations.

$$C_3 : \hat{\Gamma}[a,b] := 0. \tag{4.21}$$

4. **Related terms.** If supported by geometrical assumptions, the elements in $\hat{\Gamma}$ can be constrained to be proportional or related.

$$C_4 : \lambda_1 \hat{\Gamma}[c,d] \leq \hat{\Gamma}[a,b] \leq \lambda_2 \hat{\Gamma}[c,d], \tag{4.22}$$

where $\lambda_1$ and $\lambda_2$ are scalar values close to the expected relation between the aforementioned elements of $\hat{\Gamma}$.

Likewise, the following constraints bound the values of $\hat{\vec{\Sigma}}_u$.

5. **Positive variance.** The values of $\hat{\vec{\Sigma}}_u$ must be forced to be positive, as it represents the variance of the variation sources.

$$C_5 : \hat{\vec{\Sigma}}_u \geq \mathbf{0}. \tag{4.23}$$

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

$$C_6 : \vec{\Sigma}_{uBU_1} \leq \hat{\vec{\Sigma}}_u \leq \vec{\Sigma}_{uBU_2}, \tag{4.24}$$

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.

With this, we can reformulate the optimization problem for model estimation as

$$(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \arg \min_{\hat{\Gamma}, \hat{\vec{\Sigma}}_u} \|\hat{e}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u)\|_2^2 \tag{4.25a}$$

$$\text{s.t. } (4.17), (4.19) - (4.24). \tag{4.25b}$$

**Note:** From now on, $\hat{e}$ must be understood as $\hat{e}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u)$.

## 4.4 Numerical approach

As stated before, the proposed estimation algorithm requires solving a polynomial optimization of high degree with lots of decision variables. There are several methods and tools to solve optimization problems through semidefinite programming relaxations that may converge to the optimal value (see [99–101]). In an attempt of using those methods and available tools for our problem, we have run into numerical problems derived of the high required computational burden.

Then, in this work, we propose an alternative approach to solve our 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 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.

First, by using the Schur complement, we reduce the order of the polynomial from a $6^{\text{th}}$ order to a $3^{\text{rd}}$ order by rewriting the optimization into

$$\begin{aligned}(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \arg \min_{\hat{\Gamma}, \hat{\vec{\Sigma}}_u, t} \quad & t \\ s.t. \quad & \begin{bmatrix} t & \hat{e}^{\intercal} \\ \hat{e} & I_{n_{Sy}} \end{bmatrix} \succeq 0, \\ & Constraints\ (4.19) - (4.24),\end{aligned}$$

where scalar $t$ is a decision variable ($t \in \mathbb{R}^1$) and the new constraint is a Matrix Inequality. However, the problem is still nonlinear as $\hat{e}$ (4.17) depends on the decision variables as a $3^{\text{rd}}$ order polynomial. With this, we have reduced the polynomial order but we still require some modifications in order to address the problem through standard semidefinite programming tools.

This optimization is guaranteed to be solved within a finite time span if the vectored error function $\hat{e}$ is linear in the decision variables. In order to solve this, we propose an iterative algorithm through the linearization of $\hat{e}$.

### 4.4.1 Adjustment algorithm

The proposed iterative adjustment algorithm requires a change of the terminology and the decision variables.

First, in order to indicate the current iteration, we add a subscript within parenthesis (e.g. for the $j^{th}$ iteration of $\hat{\Gamma}$, we write $\hat{\Gamma}_{(j)}$). The next step is the definition of the new decision variables. Our iterative algorithm uses an initial approximation of $\hat{\Gamma}_{(j)}$ and $\hat{\vec{\Sigma}}_{u(j)}$ to obtain the new decision variables, which are used to update $\hat{\Gamma}_{(j)}$ and $\hat{\vec{\Sigma}}_{u(j)}$, respectively, for the next iteration.

Now, the new decision variables are:

- Matrix $\Delta\Gamma_{(j)}$, which represents the changes in $\hat{\Gamma}$ in each iteration of the algorithm ($\Delta\Gamma_{(j)} \in \mathbb{R}^{n_y \times n_u}$). Its internal structure presents the following form:

$$\Delta\Gamma_{(j)} = \begin{bmatrix} \Delta\Gamma_{(j)}[1,1] & \dots & \Delta\Gamma_{(j)}[1,n_u] \\ \vdots & \ddots & \vdots \\ \Delta\Gamma_{(j)}[n_y,1] & \dots & \Delta\Gamma_{(j)}[n_y,n_u] \end{bmatrix}.$$

- Column vector $\Delta\hat{\Sigma}_{u(j)}$ represents the changes in $\hat{\vec{\Sigma}}_u$ in each iteration of the algorithm ($\Delta\hat{\Sigma}_{u(j)} \in \mathbb{R}^{n_u}$).

The steps of the adjustment algorithm are shown as follows:

1. Initialization of the algorithm. Set the counter variable ($j = 0$). Using the result of the physical models $\Gamma_0$ (4.12) and a proposed initial approximation based on (4.13), calculate

$$\hat{\Gamma}_{(0)} = \Gamma_0, \tag{4.27a}$$

$$\hat{\vec{\Sigma}}_{u(0)} = \left(\Gamma_{(0)}^{\circ 2\intercal}\Gamma_{(0)}^{\circ 2}\right)^{-1}\Gamma_{(0)}^{\circ 2\intercal}\left(\vec{S}_y - \vec{\Sigma}_v\right). \tag{4.27b}$$

2. Solve the optimization problem:

$$(\Delta\Gamma_{(j)}, \Delta\hat{\Sigma}_{u(j)}) = \arg\min_{\Delta\Gamma, \Delta\hat{\Sigma}_u, t} t_{(j)} \tag{4.28a}$$

$$s.t. \begin{bmatrix} t_{(j)} & \hat{\varepsilon}_{(j)}^{\intercal} \\ \hat{\varepsilon}_{(j)} & I_{n_{Sy}} \end{bmatrix} \succeq 0, \tag{4.28b}$$

$$\hat{\varepsilon}_{(j)} = \hat{e}\Big|_{\hat{\Gamma}_{(j)}, \hat{\vec{\Sigma}}_{u(j)}} + \frac{\partial \hat{e}}{\partial vec(\hat{\Gamma})}\Big|_{\hat{\Gamma}_{(j)}, \hat{\vec{\Sigma}}_{u(j)}} \cdot vec(\Delta\Gamma_{(j)}) + \frac{\partial \hat{e}}{\partial \hat{\vec{\Sigma}}_u}\Big|_{\hat{\Gamma}_{(j)}, \hat{\vec{\Sigma}}_{u(j)}} \cdot \Delta\hat{\Sigma}_{u(j)}, \tag{4.28c}$$

where $\hat{\varepsilon} \in \mathbb{R}^{n_{Sy}}$ is the linearized form of $\hat{e}$:

$$\hat{e} = svec(S_y - \hat{\Gamma}_{(j)} diag^{-1}(\hat{\vec{\Sigma}}_{u(j)})\hat{\Gamma}_{(j)}^{\intercal} - \Sigma_v), \tag{4.28d}$$

evaluated around the last computed matrices.

It is also subjected to the variables from Section 4.3.2, modified to take into account the change of decision variables:

$$C_1' : \hat{\Gamma}_{(j)}[a,b] + \Delta\Gamma_{(j)}[a,b] \geq 0 \;||\; \hat{\Gamma}_{(j)}[a,b] + \Delta\Gamma_{(j)}[a,b] \leq 0, \tag{4.28e}$$

$$C_2' : \Gamma_0[a,b] + \mu_1 \leq \hat{\Gamma}_{(j)}[a,b] + \Delta\Gamma_{(j)}[a,b] \leq \Gamma_0[a,b] + \mu_2, \tag{4.28f}$$

$$C_3' : \Delta\Gamma_{(j)}[a,b] := 0, \tag{4.28g}$$

$$C_4' : \lambda_1(\hat{\Gamma}_{(j)}[c,d] + \Delta\Gamma_{(j)}[c,d]) \leq \hat{\Gamma}[a,b] + \Delta\Gamma_{(j)}[a,b] \leq \lambda_2(\hat{\Gamma}_{(j)}[c,d] + \Delta\Gamma_{(j)}[c,d]), \tag{4.28h}$$

$$C_5' : \hat{\vec{\Sigma}}_{u(j)} + \Delta\hat{\Sigma}_{u(j)} \geq \mathbf{0}, \tag{4.28i}$$

$$C_6' : \vec{\Sigma}_{uBU_1(j)} \leq \hat{\vec{\Sigma}}_{u(j)} + \Delta\hat{\Sigma}_{u(j)} \leq \vec{\Sigma}_{uBU_2(j)}. \tag{4.28j}$$

3. After the solver obtains $\Delta\Gamma_{(j)}$ and $\Delta\hat{\Sigma}_{u(j)}$, $\hat{\Gamma}_{(j)}$ and $\hat{\vec{\Sigma}}_{u(j)}$ are updated into $\hat{\Gamma}_{(j+1)}$ and $\hat{\vec{\Sigma}}_{u(j+1)}$:

$$\begin{cases} \hat{\Gamma}_{(j+1)} = \hat{\Gamma}_{(j)} + \Delta\Gamma_{(j)}, \\ \hat{\vec{\Sigma}}_{u(j+1)} = \hat{\vec{\Sigma}}_{u(j)} + \Delta\hat{\Sigma}_{u(j)}. \end{cases} \tag{4.29}$$

4. If $\left|\Delta\Gamma_{(j)}\right| \geq \delta$ (where $\delta \in \mathbb{R}^{n_y \times n_u}$ is a vector with assigned low magnitude values), increase the counter variable ($j = j+1$) and go back to step 2; else, define the final adjusted model elements as

$$\begin{cases} \hat{\Gamma} \equiv \hat{\Gamma}_{(j+1)}, \\ \hat{\vec{\Sigma}}_u \equiv \hat{\vec{\Sigma}}_{u(j+1)}, \end{cases} \tag{4.30}$$

and end the algorithm.

### 4.4.2 Justification

#### 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. We define a scalar variable $t$ that bounds the maximum value of squared $\hat{e}$. Considering that $\hat{e}^\intercal \cdot \hat{e}$ will always be zero or positive, we rewrite the optimization problem (4.25a) into

$$(\hat{\Gamma}, \hat{\vec{\Sigma}}_u) = \arg \min_{\hat{\Gamma}, \hat{\vec{\Sigma}}_u, t} \; t \tag{4.31a}$$

$$s.t.\ \hat{e}^\intercal \cdot \hat{e} \leq t. \tag{4.31b}$$

Constraint (4.31b) is rewritten into

$$t - \hat{e}^\intercal \cdot I_{n_{Sy}} \cdot \hat{e} \geq 0. \tag{4.32}$$

Using the properties of the Schur Complement, we affirm that (4.32) will be positive semi-definite if and only if the following matrix $H$ is also positive semi-definite.

$$H = \begin{bmatrix} t & \hat{e}^\intercal \\ \hat{e} & I_{n_{Sy}} \end{bmatrix} \succeq 0. \tag{4.33}$$

Thus, both expressions are exchangeable without losing their inequality properties. Linear Matrix Inequality (4.33) is applied in constraint (4.28b).

#### Linearization

The error function $\hat{e}$ (4.17) is a 3$^{\text{rd}}$ order polynomial. We reduce its order using a first-order Taylor series linearization around an initial estimation of $\hat{\Gamma}$ and $\hat{\vec{\Sigma}}_u$.

These series require several iterations until the terms stabilize, so for a given iteration $l$, the proposed relaxation presents the form

$$\hat{\varepsilon}_{(l)} = \hat{e}\Big|_{\hat{\Gamma}_{(l)}, \hat{\vec{\Sigma}}_{u(l)}} + \frac{\partial \hat{e}}{\partial vec(\hat{\Gamma})}\Big|_{\hat{\Gamma}_{(l)}, \hat{\vec{\Sigma}}_{u(l)}} \cdot vec(\Delta\Gamma_{(l)}) + \frac{\partial \hat{e}}{\partial \hat{\vec{\Sigma}}_u}\Big|_{\hat{\Gamma}_{(l)}, \hat{\vec{\Sigma}}_{u(l)}} \cdot \Delta\hat{\Sigma}_{u(l)},$$

where $\hat{\varepsilon}$ is the linearized form of $\hat{e}$. As explained before, $\Delta\Gamma_{(j)} \in \mathbb{R}^{n_y \times n_u}$ and $\Delta\hat{\Sigma}_{u(j)} \in \mathbb{R}^{n_u}$ are the new decision variables for the iterative procedure.

Given that $vec(\hat{\Gamma}) = \left[\hat{\Gamma}[1,1]\ \hat{\Gamma}[2,1]\ ...\ \hat{\Gamma}[n_y, n_u]\right]^\intercal$ and knowing from (4.14) that $\hat{\vec{\Sigma}}_u = \left[\hat{\vec{\Sigma}}_u[1],\ ...,\ \hat{\vec{\Sigma}}_u[n_u]\right]^\intercal$, then

$$\frac{\partial \hat{e}}{\partial vec(\hat{\Gamma})} = \left[\frac{\partial \hat{e}}{\partial \hat{\Gamma}[1,1]} \quad \frac{\partial \hat{e}}{\partial \hat{\Gamma}[2,1]} \quad \cdots \quad \frac{\partial \hat{e}}{\partial \hat{\Gamma}[n_y,n_u]}\right],$$

$$\frac{\partial \hat{e}}{\partial \hat{\vec{\Sigma}}_u} = \left[\frac{\partial \hat{e}}{\partial \hat{\vec{\Sigma}}_u[1]} \quad \frac{\partial \hat{e}}{\partial \hat{\vec{\Sigma}}_u[2]} \quad \cdots \quad \frac{\partial \hat{e}}{\partial \hat{\vec{\Sigma}}_u[n_u]}\right].$$

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 equation (4.29).

As a final summary, in Figure 4.3 we present a diagram of the adjustment methodology.

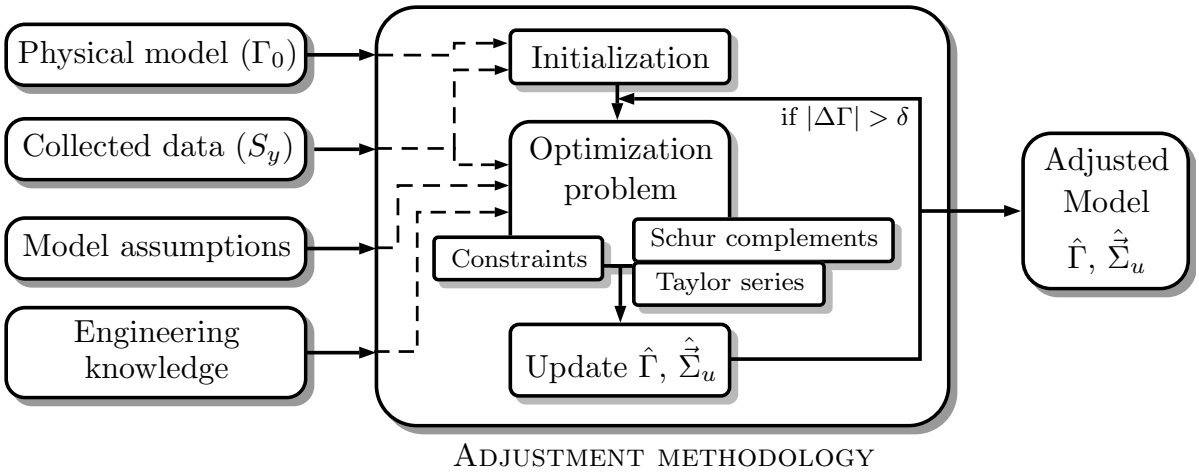



**Figure 4.3.** Summary diagram of the adjustment methodology.

## 4.5 Validation of the model

The proposed adjustment algorithm (4.27)-(4.30), through an iterative optimization procedure, uses an estimation of $\Sigma_y$ by means of $S_y$, and a metric that estimates the sum of the square of the difference of the elements of $\Sigma_y$ 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 $\Sigma_y$, $\Sigma_u$, $\Sigma_v$ and $\Gamma$. 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 $\hat{\vec{\Sigma}}_u$:

$$[\hat{\Gamma},\ \hat{\vec{\Sigma}}_u] = f\left(S_{yTr},\ \Gamma_0\right). \tag{4.34}$$

The model is validated using performance indexes, which compare the sample covariance values and the theoretical values from the benchmark with the estimated covariance of the KPCd measurements ($\hat{\Sigma}_y$), obtained using equation (4.16). These performance indexes are vectored using symmetrical vectorization in order to avoid the duplication of the effect of the elements located outside the diagonal.

First, we present the performance indexes that can only be used in simulations, as they use the theoretical real values of matrices $\Sigma_y$, $\Sigma_u$, $\Sigma_v$ and $\Gamma$:

- 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 $\Sigma_v$ is assumed to be known, index $I_1$ presents this form:

$$I_1 = svec\left(\Sigma_y - \hat{\Sigma}_y\right)^{\intercal} svec\left(\Sigma_y - \hat{\Sigma}_y\right) = \\ = svec\left(\Gamma\Sigma_u\Gamma^{\intercal} - \hat{\Gamma}\ diag^{-1}(\hat{\vec{\Sigma}}_u)\hat{\Gamma}^{\intercal}\right)^{\intercal} svec\left(\Gamma\Sigma_u\Gamma^{\intercal} - \hat{\Gamma}\ diag^{-1}(\hat{\vec{\Sigma}}_u)\hat{\Gamma}^{\intercal}\right). \quad (4.35)$$

- 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:

$$I_2 = \left(\frac{I_1}{svec\left(\Sigma_y - \Sigma_v\right)^{\intercal} svec\left(\Sigma_y - \Sigma_v\right)}\right)^{\frac{1}{4}} \cdot 100 = \\ = \left(\frac{svec\left(\Gamma\Sigma_u\Gamma^{\intercal} - \hat{\Gamma}\ diag^{-1}(\hat{\vec{\Sigma}}_u)\hat{\Gamma}^{\intercal}\right)^{\intercal} svec\left(\Gamma\Sigma_u\Gamma^{\intercal} - \hat{\Gamma}\ diag^{-1}(\hat{\vec{\Sigma}}_u)\hat{\Gamma}^{\intercal}\right)}{svec\left(\Gamma\Sigma_u\Gamma^{\intercal}\right)^{\intercal} svec\left(\Gamma\Sigma_u\Gamma^{\intercal}\right)}\right)^{\frac{1}{4}} \cdot 100. \quad (4.36)$$

- 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.

$$I_3 = \max\left(\left|\sqrt{\vec{\Sigma}_y} - \sqrt{\hat{\vec{\Sigma}}_y}\right|\right). \quad (4.37)$$

- Index $I_4$ expresses the maximum ratio of the estimation error of the standard deviations w.r.t. the standard deviation of $\vec{\Sigma}_y$:

$$I_4 = \max\left(\frac{\left|\sqrt{\vec{\Sigma}_y} - \sqrt{\hat{\vec{\Sigma}}_y}\right|}{\sqrt{\vec{\Sigma}_y}}\right) \cdot 100. \quad (4.38)$$

We also present the practical performance index $I_P$, which can be used by practitioners in real cases:

- 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.

$$I_P = svec\left(S_{yTs} - \hat{\Sigma}_y\right)^{\intercal} svec\left(S_{yTs} - \hat{\Sigma}_y\right). \quad (4.39)$$

As a summary, $I_P$ 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 following section we will show, through numerical examples in a case study, how $I_P$ can be related with other indexes and how it can help to decide if we have reached a proper model of the process.

## 4.6 Simulations

### 4.6.1 Case Study

#### Benchmark

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 assume that matrix $\Gamma_0$, which has been obtained using physical models, presents notable deviations w.r.t. the real values of $\Gamma$ as a result of simplifications and unreliable modelizations of some parts of the process.

We have generated a simulated benchmark for this case study. In this case, for the sake of dimensional simplicity but without loss of generality, we have used as $\Gamma$ an adapted version of the matrix presented in a case study from [76]:

$$\Gamma = \begin{bmatrix} 0.093 & 0.577 & -0.120 \\ -0.093 & 0 & 0.843 \\ 0.093 & 0.577 & -0.120 \\ 0.647 & 0 & -0.120 \\ -0.370 & 0.577 & 0.482 \\ 0.647 & 0 & -0.120 \end{bmatrix}. \tag{4.40}$$

The simulated values of the KPCd measurements of each part at the last stage ($y_M(i)$) are generated using $\Gamma$ and randomly generated values of the effect of the variation sources ($u(i)$) and the measurement noises ($y_M(i)$). The theoretical variance of the effect of the variation sources ($\Sigma_u$) is obtained from [56], which used a variation of (4.40) as a benchmark. Here, we assume that $u$ presents a Gaussian distribution with zero mean and a variance $\sigma_u^2 = 1.111 \cdot 10^3\ \mu m^2$. Thus, $\Sigma_u = \sigma_u^2 \mathbf{I}$, given that we assume that all variation sources behave with the same variance.

Measurement noise is applied as a zero-mean Gaussian signal. In the proposed simulations we propose several variances ($\sigma_v^2$) for the measuring instruments, in order to evaluate the effect of the measurement noise variance on the performance of the proposed algorithm. We consider that all KPC deviations are measured with the same instrument, so the variances of all measurement noises are identical ($\Sigma_v = \sigma_v^2 \mathbf{I}$). Each case is explained in Section 4.6.1.

Values for $\Gamma_0$ are randomly generated within a 30% range w.r.t. the values of $\Gamma$, in order to represent the aforementioned deviations.

We also consider that the following constraints are known from the process; therefore, they are used in the optimization procedure to adjust the model:

- The values of $\Gamma$ 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 w.r.t the initialization matrix $\Gamma_0$.
- The values of $\hat{\Sigma}_u$ have been forced to be positive, and within the $500 - 1500\ \mu m^2$ range, which would be obtained using backup data from other similar process.

### Simulation settings

The performance of the proposed methodology is evaluated for different amounts of processed parts in the training and the testing set, as well as for different variances of the measurement noise. Given that for each experiment the values of $\Gamma_0$ are randomly generated, and the individual states of the variation sources and the measurement noise are generated randomly for each processed part, we have applied the Monte Carlo method, where for each combination of number of training and testing parts, as well as a given measurement noise variance, the experiments are repeated 200 times. The simulation parameters are shown in Table 4.1. Note that the amount of testing parts will only affect index $I_P$.

**Table 4.1.** Simulation parameters.

| | |
|---|---|
| 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} |

In Table 4.2, we propose several experiment cases using different measurement noises, including the proportion in percentage between the standard deviations of the measurement noise and the standard deviations of the KPCd measurements, defined as:

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

**Table 4.2.** Simulation experiment cases depending on the measurement noise variance.

| Experiment | A | B | C | D | E |
|---|---|---|---|---|---|
| $\sigma_v^2\ (\mu m^2)$ | 1 | 4 | 16 | 64 | 256 |
| $Prop_v\ (\%)$ | 3.72 | 7.42 | 14.75 | 28.77 | 52.36 |

For each new iteration of the Monte Carlo method, we generate a new set of data for the training and the testing sets, as well as a new $\Gamma_0$. Then, in order to adjust the model by solving the proposed optimization problem, we use the YALMIP parser [102] and the optimization software *mosek*. Then, we calculate and average the performance indexes for each combination of training and testing amounts of processed parts.

## 4.6.2 Results and discussion

### Evaluation of the performance indexes

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

Index $I_1$ is presented in Figure 4.4 for the different experiment cases presented in Table 4.2. As it can be observed, index $I_1$ (the error in the estimation of $\Sigma_y$; equation (4.35), which uses $\hat{\Gamma}$ and $\hat{\bar{\Sigma}}_u$) is reduced when the number of processed parts used in the training set increases.

As expected, given that a higher amount of samples of the effects of the variation sources (considering that they are independent) implies that the sample covariance of the variation sources will be similar to our assumptions for $\Sigma_u$. It can also be observed in Figure 4.4 that, when the number of processed parts is zero (i.e. when the estimation is performed using the initialization equations from (4.27a) and (4.27b)), certain error in the estimation is observed, which suddenly increases and then decreases as the number of processed parts on the training set increases. When there are around 20-25 parts in the training set, index $I_1$ already has the same magnitude as the initial value had, and after that it keeps diminishing with the amount of parts in the training set. The 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 $\Gamma \in \mathbb{R}^{6\times 3}$ and $\hat{\vec{\Sigma}}_u \in \mathbb{R}^{3\times 1}$, 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. In both Figures 4.4 and 4.5 it can be observed that the performance of the proposed methodology is reduced when the standard deviation of the measurement error (noise) increases. However, the reduction of the performance is minimal.

Index $I_2$ (equation (4.36)) is presented in Figure 4.6 for the different experiment cases. Its general behavior is very similar to index $I_1$, as increasing the variance of the measurement noise also increases the value of this performance index.

Indexes $I_3$ and $I_4$ (equations (4.37) and (4.38)) are presented in Figures 4.7 and 4.8, 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 $I_P$ (equation (4.39)) is presented in Figure 4.9a for measurement noises of $\sigma_v^2 = 4\ \mu m^2$ (Exp. B) and in Figure 4.9b for measurement noises of $\sigma_v^2 = 256\ \mu m^2$ (Exp. C), using several amounts of processed parts in the testing set. As it can be observed in both cases, it is necessary to use a notably high amount of parts in order to use $I_P$ as a reliable substitute of $I_1$ in a practical case.

### Behavior of performance index $I_1$ as a process estimator

Figure 4.10 shows the performance index $I_1$ in several experiments. Each experiment presents a different measurement noise standard deviation, in order to evaluate the effect of the measurement noise on the proposed methodology. The obtained surface can be fitted into the following form:

$$I_1 \sim a + \frac{b\sigma_v^2 + c\sigma_v + d}{N_{tr}},$$

where $N_{tr}$ is the number of parts of the training set. The initial peak before the 25 parts is omitted, as it is a high-error zone where there is not enough data to obtain acceptable estimations. Thus, performance index $I_1$ is inversely proportional to the number of parts used in the training set, similarly to other process estimators [98]. $I_1$ is also affected by the precision

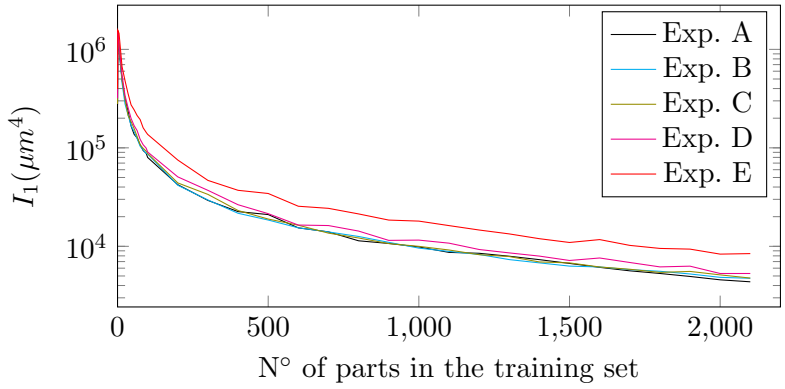



**Figure 4.4.** Index $I_1$ w.r.t. number of parts used in the training set.

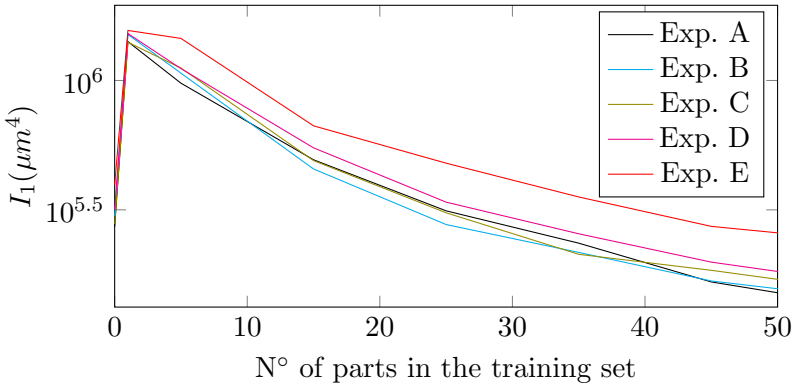



**Figure 4.5.** Detail of the evolution of the index $I_1$ cases presented in Figure 4.4.

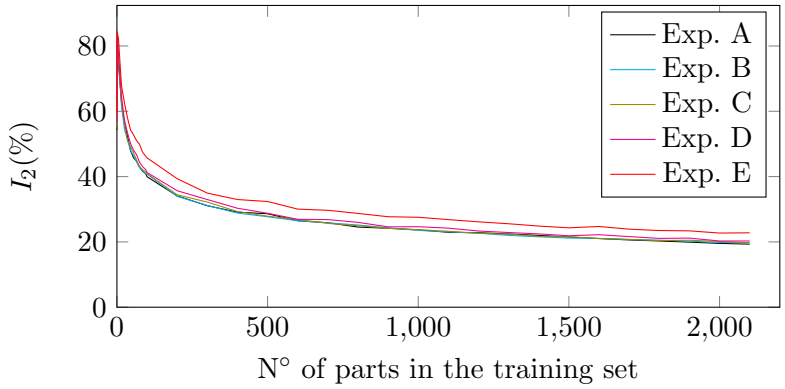



**Figure 4.6.** Index $I_2$ w.r.t. number of parts used in the training set.

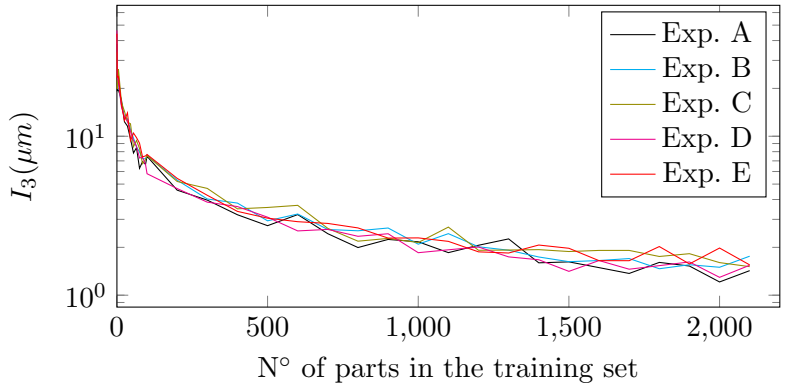



**Figure 4.7.** Index $I_3$ w.r.t. number of parts used in the training set.

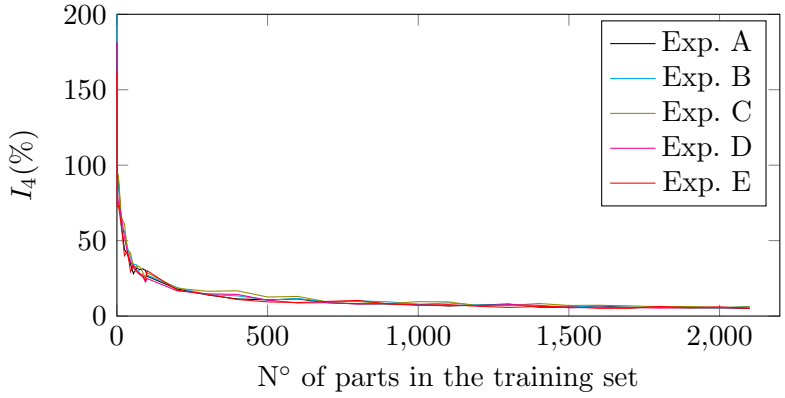
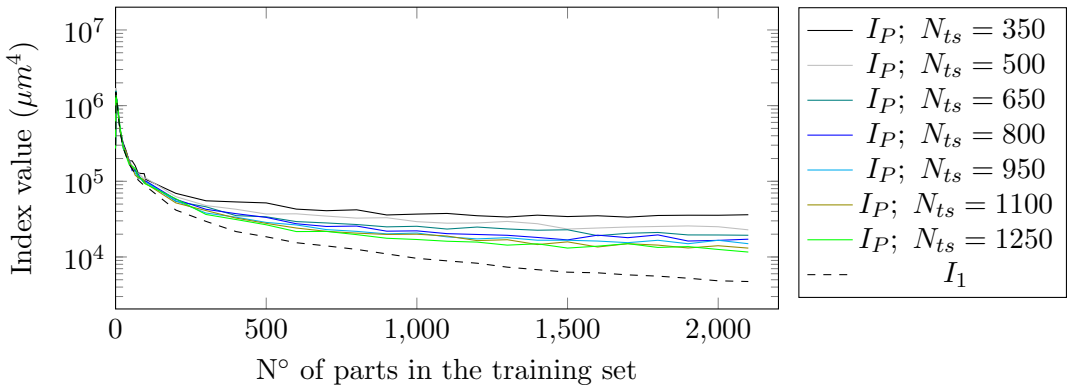
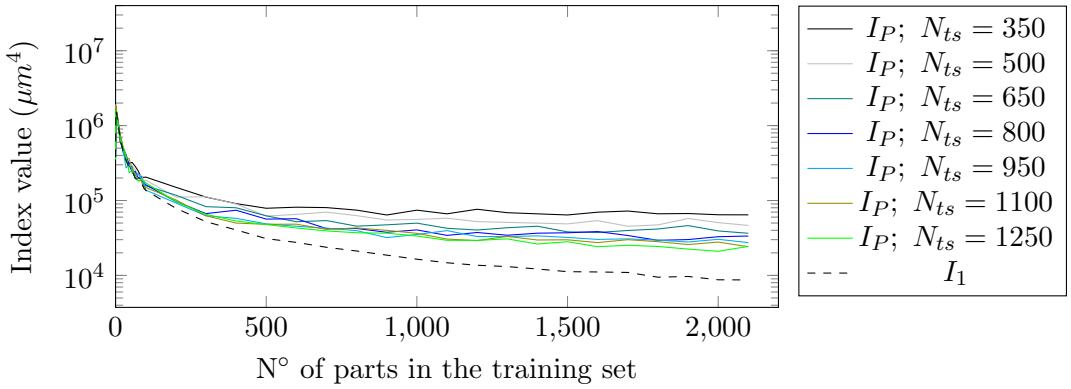



**Figure 4.8.** Index $I_4$ w.r.t. number of parts used in the training set.




**(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).

**Figure 4.9.** Comparison of $I_P$ for different $N_{ts}$ and $I_1$ for different measurement noises.

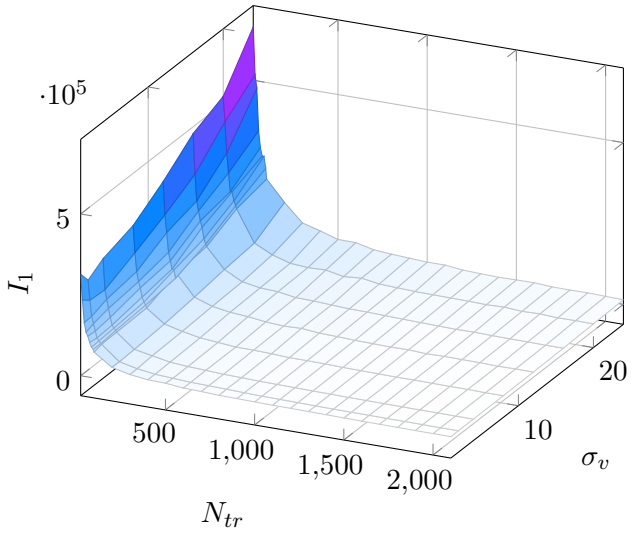



**Figure 4.10.** Evolution of performance index $I_1$ for different number of processed parts in the training set and different standard deviations of the measurement noise.

of the measuring instrument, as it is proportional to a polynomial function of the standard deviation of the measurement noise.

## 4.7 Conclusions

This chapter presents a methodology to reduce the linearization, approximation and modeling errors that arise during the development of the linear input-output variation propagation model of an MMP with a large amount of stages and/or with components that, due to their configuration, cannot be reliably modeled using physical models.

The proposed methodology consists of recursively solving an optimization 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 $\Gamma$ and the estimated covariance of the variation sources. This optimization problem is initialized using physical models and it is bounded using prior engineering knowledge and backup data.

After applying the proposed methodology to a simulated case study, we validated the 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 to its respective sample covariance, although as it is always assumed that their values will be notably lower than those of the KPCd measurements covariance, so this assumption may not notably affect the accuracy of the model adjustment. Additionally, we consider that the magnitude of the errors in the model caused by approximations, linearizations 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 ensure that the optimization problem is solved within a finite time. Future research may include a greater refinement of the linearization methods in the algorithm and analyzing the impact of the adjusting methodology in processes with higher dimensionality.

Chapter 5

# A sequential inspection procedure for fault detection in multi-stage manufacturing processes


## Abstract

Fault diagnosis in multistage manufacturing processes (MMPs) is a challenging task where most of the research presented in the literature considers a predefined inspection scheme to identify the sources of variation and make the process diagnosable. In this chapter, a sequential inspection procedure to detect the process fault based on a sequential testing algorithm and a minimum monitoring system is proposed. After the monitoring system detects that the process is out of statistical control, the features to be inspected (end of line or in process measurements) are defined sequentially according to the expected information gain of each potential inspection measurement. A case study is analyzed to prove the benefits of this approach with respect to a predefined inspection scheme and a randomized sequential inspection considering both the use and non-use of fault probabilities from historical maintenance data.


## 5.1 Introduction

In the last years, international institutions such as the European Factories of the Future Research Association (EFFRA) have promoted the development of strategies for modeling, monitoring and controlling complex manufacturing systems to achieve zero-defects [103].

Multistage Manufacturing Processes (MMPs) are sequential manufacturing processes where workpieces move throughout different stages in order to perform specific manufacturing operations (e.g., welding, machining, etc.). Typical MMP in industry are automotive body assemblies, machining lines, rolling processes, tile manufacturing processes, etc. One of the main characteristics of MMPs is the complex interactions among stages that define the final quality of the

product. This is mainly due to the fact that the output quality at one stage is affected by the output quality of preceding stages. This complexity makes their control and quality assurance challenging.

If attention is focused on quality assurance in MMP, inspection allocation, monitoring and fault diagnosis/identification are key issues that should be studied in detail. Many research works have been published on these topics in the last decade, and interesting surveys and reviews can be found in recent works [74, 104–106].

In the field of fault diagnosis, a model that relates key product characteristics (KPCs) to sources of variation is needed for an effective root cause analysis. This model can be defined by engineering or data-driven approaches. A model based on engineering approaches can be obtained by deriving the physical laws that explain the process, e.g., kinematic relationships in assembly processes. A well-known engineering-based model in MMPs is the Stream of Variation (SoV) model [9] which has been successfully applied for fault diagnosis in different researches. Zhou et al. [107] showed in detail the characteristics of the MMP for a fully diagnosable system considering the SoV model as a linear mixed-effects model. Conditions for the diagnosability property and the concept of minimal diagnosable class to analyze partial diagnosable systems were also illustrated. Ding et al [98] compared different on-line variation estimators given continuous dimensional measurements for fault diagnosis purposes. In [78], the root-cause identification is formulated as a problem of estimation and hypothesis testing. In this work, on-line batch algorithms for the mean and variance estimation together with the hypothesis-testing methods for root-cause identification are illustrated. Sales-Setién et al [56] proposed a recursive algorithm to estimate the process variance instead of on-line batch estimators, which reduces the computational cost and the data storage needs. Ding et al. [80] used the engineering model and the measurements at the inspection stage to identify fixture faults by a pattern recognition strategy based on principal component analysis. Although some fixtures presented the same pattern error on KPCs and, therefore, cannot be diagnosable, the fault patterns between stations were diagnosable. Xiang and Tsung [108] described how to define a control chart for statistical process control in a MMP based on the SoV model. The complex multi-stage monitoring problem is converted to a simple multi-stream monitoring problem by applying group exponential weighted moving average (EWMA) charts to the one-step ahead forecast errors of the model. The faulty stage is identified according to the results of the one-step ahead forecast errors. In a similar work, Li and Tsung [109] used the SoV model and EWMA charts for detecting and identifying the faults that affect the process covariance matrix in MMPs.

On the other hand, data-driven models are based on shop-floor data to extract the spatial pattern vectors (SPVs) that define the relationships between KPCs and sources of variation. Jin and Zhou [81] extracted the SPVs from the inspection data (sample covariance matrix) and they are compared with SPVs that have been previously extracted and whose sources of variation have been identified. Shan and Apley [86] proposed various blind source separation criteria to estimate the SPVs. Liu et al [88] proposed the use of a qualitative model to relate KPCs with sources of variation instead of the SoV model and used this information to adjust in a proper way the SPVs extracted from data-driven approaches. The use of both engineering

approaches (i.e, the qualitative model) and the data-driven approaches let explain in a better way the extracted SPVs from the data.

Other advance modeling techniques such as Hierarchical Bayesian Networks (HBNs) have been also applied for monitoring and fault diagnosis in MMPs. In [110], a HBN is built using only data (process model is unknown) and once the network has been trained, the HBN is used to infer the unobserved inputs of the process (sources of variation). The identification of the fault and its type (mean shift or variance change) is accomplished by a control chart using the measured data and the inferred value from the HBN. Another HBN is proposed in [111] to deal with fault diagnosis in MMPs when the process is under-determined. Under the assumption that less process faults are more likely to occur in MMPs, the problem of fault diagnosis is transformed into searching the sparse solution of abnormal variance changes for process faults. A similar problem is covered in [112], where the authors proposed a spatially correlated Bayesian learning algorithm for fault diagnosis. The algorithm is based on the relevance vector machine (RVM) exploiting the spatial correlation of dimensional variation from various process errors and a real automotive assembly is used to validate the effectiveness of the algorithm. Other artificial intelligence techniques have been explored for defect detection in similar contexts of MMPs and interesting reviews can be found in [113, 114]. In [115], supervised and unsupervised learning approaches were explored to estimate healthy and unhealthy parts along the manufacturing process using different sensors data such as dynamometers, accelerometers, thermocouples, etc. Although this research does not deal with fault diagnosis, the estimation is used to reduce the number of inspections to be conducted since only those where the estimation cannot be ensured within a certain level of confidence are conducted. Beruvides et al. [116] presented a fault pattern identification methodology for multistage assembly processes with non-ideal sheet metal parts. Three different supervised and unsupervised neural network topologies (multi-layer perceptron network -MLP-, self-organized map -SOM-, and a MLP with genetic algorithms) with a Q-learning algorithm were implemented to compose a fault pattern identification library. All three methods were validated in a case study and the SOM network presented the best accuracy for fault pattern identification.

However, despite the large contributions in the field of fault diagnosis in MMPs, most of the research works are based on the existence of diagnosability conditions [107], which means that enough measurements are available to detect and identify the source of variation. Furthermore, these measurements are available at any time and almost at any station, since the diagnosability condition requires a large amount of data with enough information to isolate and identify the sources of variation. However, this approach may be not easy to be implemented in industry. Despite current trends of Industry 4.0, the cost of implementing and using at any time all measurements in a MMP may produce an important cost. Note that not only on machine measurements which could be non-invasive and without operator's action are considered, but also in process measurements that may require use of CMM, gaging systems, etc. Therefore, a more conservative approach where the measurements are conducted only when the search of a root cause is necessary may be of great interest.

This chapter proposes a sequential inspection procedure for fault diagnosis in MMP where, instead of measuring at any time most of the stages needed for full diagnosticability, the fault

diagnosis is conducted in a sequential way. The proposed system is based on two parts. In the first part, a monitoring system is implemented to identify if the process is out of control. In the second part, a sequential inspection based on the evaluation of the information gain of each potential inspection measurement is conducted to detect the existing fault in the process. Note that the purpose of the system is to detect and isolate the fault, but there is no need for a complete identification of the fault, i.e, we want to know which fault exists without estimating its value. The methodology presented in this chapter is based on a qualitative model of process faults and KPCs, which is derived using a type of tree diagram commonly applied in tolerance charting. This model is used instead of engineering models (e.g., SoV model) which can be difficult to derive for practitioners.

This chapter is organized as follows. Section 5.2 shows the problem description and the proposed methodology for the sequential inspection procedure. Section 5.3 illustrates how to derive the qualitative model between sources of variation and KPCs using a graphical tree commonly applied in tolerance charting. Section 5.4 shows the minimum monitoring system that is needed to ensure all sources of variation can be detected. Section 5.5 presents the proposed sequential inspection methodology for a rapid inspection sequence and fault detection and isolation. Several case studies are analyzed under the proposed inspection approach and the results are compared with other possible inspection schemes in Sections 5.6 and 5.7. Finally, Section 5.8 points out the main conclusions of the chapter.

## 5.2 Problem description

Let us consider a MMP as shown in Figure 5.1, where the raw material starts at stage 1 and undergoes a series of manufacturing operations until the last stage, $N$. At each stage, critical process characteristics may affect the result on part quality, for instance, a fixture locator which plays a critical role in determining the dimensional quality of an assembled or machined part. These critical characteristics are called key control characteristics (KCC), and their deviations from their nominal values at stage $k$ are denoted as $u_k$. The quality of the part is evaluated through an inspection stage or by on machine measurements and the deviations of KPCs from nominal values at stage $k$ are denoted by $y_k$. If a linear model links the deviations of KCCs (i.e., sources of variation) with the deviations of KPCs derived from measurements, the following Equation is defined:

$$y = \Gamma\ u + \varepsilon, \tag{5.1}$$

where $y = [y_1^\intercal\ y_2^\intercal\ ...y_N^\intercal]^\intercal$ is an $m \times 1$ vector that represents the measured dimensional deviation of KPCs from station 1 to station $N$; $u = [u_1^\intercal\ u_2^\intercal\ ...u_N^\intercal]^\intercal$ is an $n \times 1$ vector that represents the deviations of KCCs up to station $N$; $\Gamma$ is the fault pattern matrix ($m \times n$) that can be derived from engineering or data-driven approaches; and $\varepsilon$ denotes a term that includes both the modeling uncertainty and the measurement noise ($v_k$).

As shown in Equation (5.1), to fully identify all sources of variation, measurements along the MMP should be conducted. The diagnosability of these sources of variation and the final

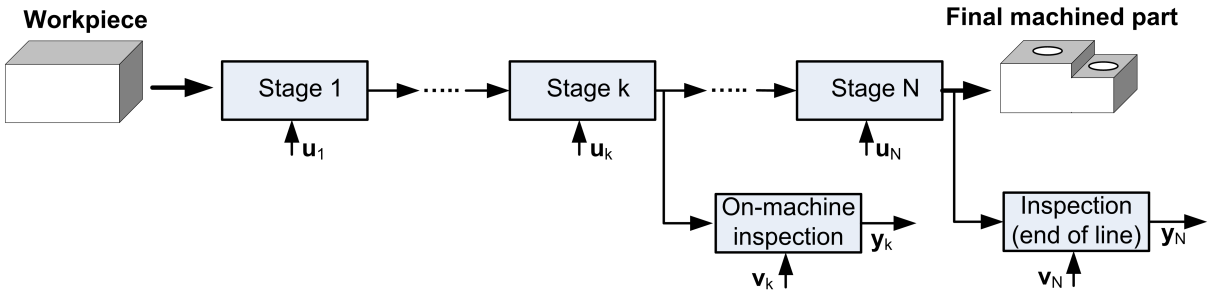



**Figure 5.1.** Multistage Manufacturing Process (MMP) with N stages. Notation: variation sources (u), measurement noise (v), inspection measurements (y).

inspection cost are the main issues in the design of the inspection scheme in MMPs for fault diagnosis and quality assurance.

Given this MMP, the following questions may arise: which KPCs should be inspected for monitoring the process at the end of line? Which stages/KPCs should be inspected for fault diagnosis purposes? Which inspection sequence should be followed to identify the sources of variation with a minimum number of measurements? Note that previous research have dealt with similar problems but, after the definition of the inspections stations, the measurements were assumed to be obtained at any time. In the presented problem, a sequential approach is proposed and thus, the decision of which stage or KPC should be inspected depends on the results of previous inspections.

To solve this problem, the following 3-steps methodology is proposed:

1. Derivation of a qualitative model between sources of variation and KPCs.
2. Definition of a minimum monitoring system to trigger the sequential inspection procedure.
3. Sequential inspection procedure based on the Information Gain (IG).

For the research in this chapter the following is assumed:

- The analyzed MMP is composed of stations that conduct machining operations, and therefore, the potential process faults are related to fixtures and cutting tools.
- Only one fault exists at the same time in the MMP.
- Type I errors (true conforming parts are considered nonconforming after inspection) and Type II errors (true nonconforming parts are considered conforming after inspection) are assumed to be negligible.

The following sections show in detail the 3-step methodology proposed, which is illustrated in Figure 5.2.

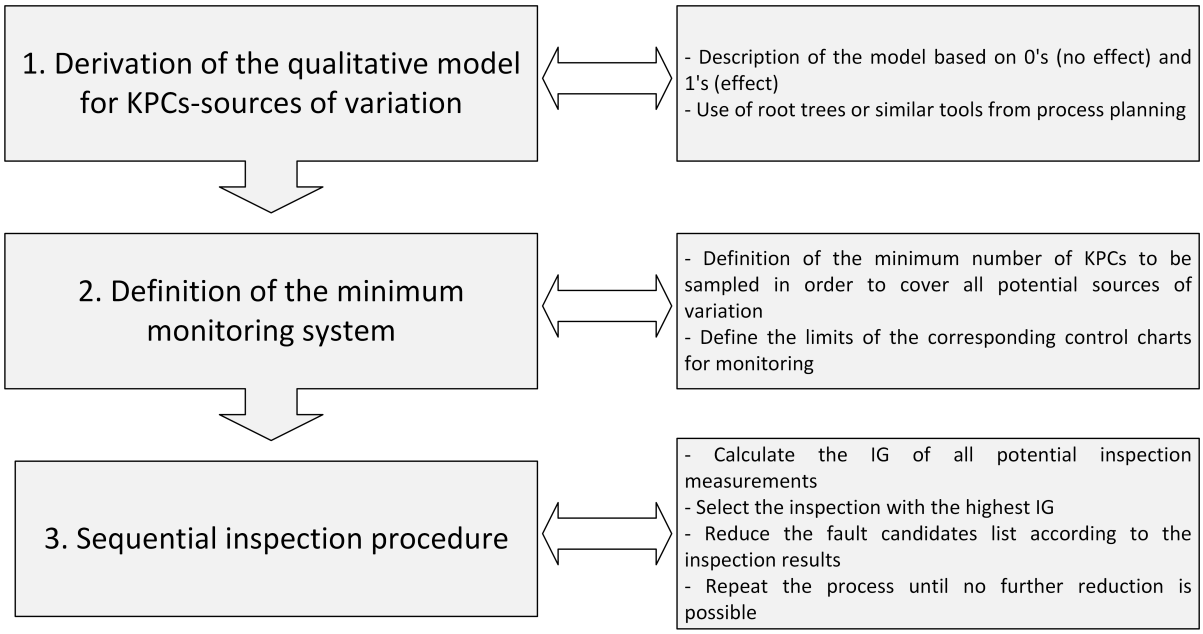



**Figure 5.2.** Methodology overview for fault detection based on sequential inspection.

## 5.3 Qualitative model of KPCs - process faults

The qualitative model of KPCs-process faults refers to the qualitative estimation of matrix $\Gamma$ from Equation (5.1). As explained above, this matrix can be obtained from engineering or data-driven approaches. However, in this chapter the use of a simpler model considering the qualitative relationships of the MMP to indicate which source of variation influences on which KPCs is explored. If a relationship exists, the corresponding $\Gamma$ coefficient has a value of 1. Otherwise, the value is 0.

The qualitative model is extracted from the process planning information, more specifically from tolerance charting. Tolerance charting is a common activity that is performed in process planning to ensure that design tolerances can be achieved. To analyze the variation propagation and estimate if the part is within specifications, a root tree and a tolerance chart is built. The root tree is a graphical representation of the process where the sequence of machined surfaces and datums (surfaces used for locating the workpiece in the fixture) can be extracted. A brief explanation of the rooted tree is given in [117].

In this chapter, the following modification of the rooted tree for deriving a qualitative model of KPCs-process faults is proposed:

- Machining operations that are conducted with the same tool are represented with the same type of arrow at each subjob/stage.
- If a feature previously machined is used as datum downstream, the feature is drawn two times connected by a thick line.

- Whether on-machine measurement inspections are conducted and the potential process faults are indicated on the right hand side of the rooted tree. Two types of process faults are distinguished:
  - i) cutting tool faults (excessive wear or breakage), denoted as $u_f$;
  - ii) fixture faults (deviations of locators or workholding devices), denoted as $u_m$.

  Similarly, two types of on-machine inspections are distinguished:
  - i) tool inspection or KPC inspection, denoted as $y_{um}$;
  - ii) fixture inspection, denoted as $y_{uf}$.

Furthermore, it is assumed that for the purpose of fault detection and isolation, the machining error due to machine-tool precision is negligible, and thus, the machining error only refers to cutting tool errors due to excessive tool wear or tool breakage.

To illustrate the rooted tree for a MMP with the above modifications, let us consider the MMP shown in Figure 5.3. The process plan is as follows. At stage 1, the workpiece is clamped using as datum the raw surfaces B2 and B3, and it is machined with the same cutting tool to obtain surfaces S1 and S2. At stage 2, the workpiece is located using the datum surfaces S2 and B2. At this stage, surfaces S6 and S7 are machined with the same end mill tool; surface S4 is generated using a drilling tool. The KPCs that are of interest according to the drawing specifications are: KPC1, distance between S7 and S1; KPC2, distance between S3 and S2; KPC3, distance between S6 and S4. Under this process plan, the resulting rooted tree is shown in Figure 5.4.

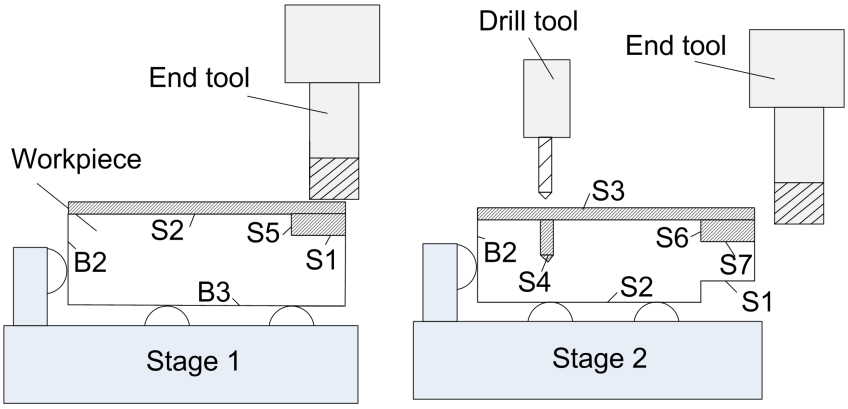



**Figure 5.3.** Example of an MMP to illustrate the qualitative model.

Given the information from the rooted tree and the KPCs, the derivation of matrix $\Gamma$ that connects the sources of variation with the KPCs can be easily obtained. The matrix is drawn following the procedure shown below:

- Look for the features that define the KPCs. For instance, KPC1 is the distance between S7 and S1.

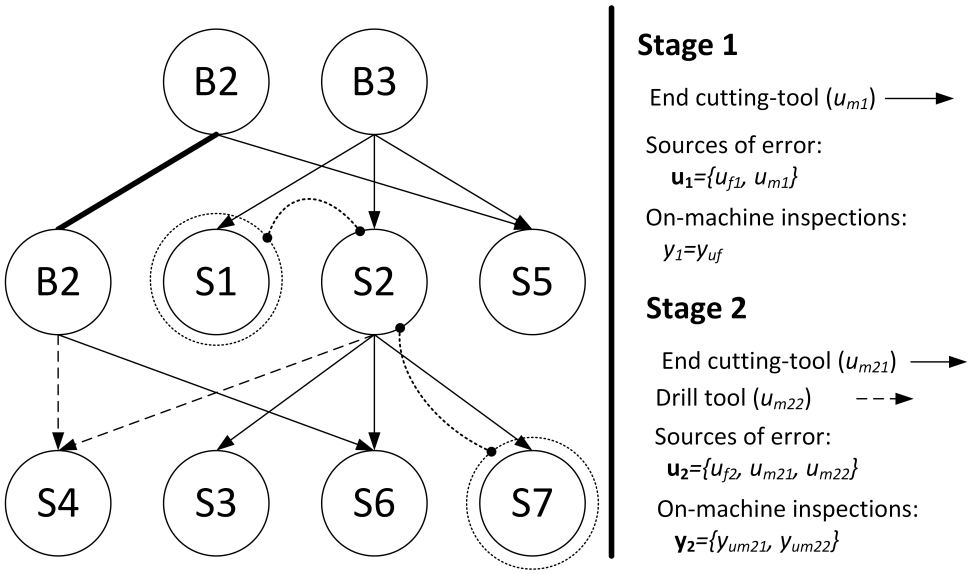



**Figure 5.4.** Resulting rooted tree of previous MMP example. The path related to KPC1 (distance between S1 and S7) is shown in dotted lines to clarify the methodology to obtain the matrix $\Gamma$ in the text. text. B2–B3, raw surfaces; S1–S7, machined surfaces.

- Find the path that connects both features.
- Each path defines the row of matrix $\Gamma$. This row is defined by 1's or 0's as follows:
  - An arrow means a cutting tool error, thus 1 is set to the corresponding column of this cutting tool error.
  - When the path moves from one stage to the next one, a fixture error is added from the first stage, thus, 1 is set to that fixture error.
  - If a path includes a thick line, this line does not add any value in the model.
  - If the path includes two machined features in the same stage, no fixture error is added (the fixture errors are compensated), thus, a 0 is set to that fixture error. Similarly, if the cutting tool is used to machine both surfaces, a 0 is also set in the corresponding cutting tool error. If different cutting tools are used, a 1 is set to each corresponding cutting tool error.
  - Any error that is not identified in the path is set to 0.
  - For on-machine measurements of fixtures, set 1 to that fixture errors.
  - For on-machine measurements of cutting tools (surface inspections with a touch probe on machine or direct inspection of tools), a 1 is set to cutting tool errors at that stage.

To illustrate the procedure, let us consider the KPC1 which is defined by the distance between surface S7 and surface S1. The path that connects both surfaces is illustrated in Figure 5.4 using dotted lines. As it can be seen, from S7 to S2 there is an arrow that represents the machining process with the end cutting tool, so this source of error is set to 1 ($u_{m21}$). Then,

surface 2 is used as datum and thus, the fixture error of stage 2 is added ($u_{f2}$). Finally, the path moves from surface 2 to surface 1 using the datum B3. Both surfaces are machined with the same tool and same datum, so no additional errors are added. Therefore, the row of $\Gamma$ matrix for the KPC1 is $[0, 0, 1, 1, 0]$. Note that the source of errors is $u = [u_{f1}, u_{m1}, u_{f2}, u_{m21}, u_{m22}]^{\intercal}$.

As a result of applying this procedure, the qualitative model KPCs-sources of variation is defined as:

$$y = \begin{bmatrix} y_{on-machine} \\ y_{end-line} \end{bmatrix} = \Gamma \cdot u = \begin{bmatrix} \Gamma_{on-machine} \\ \Gamma_{end-line} \end{bmatrix} \cdot \begin{bmatrix} u_{f1} \\ u_{m1} \\ u_{f2} \\ u_{m21} \\ u_{m22} \end{bmatrix}, \tag{5.2}$$

$$y_{on-machine} = \begin{bmatrix} y_{uf1} \\ y_{um2} \end{bmatrix} = \Gamma_{on-machine} \cdot \begin{bmatrix} u_{f1} \\ u_{m1} \\ u_{f2} \\ u_{m21} \\ u_{m22} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_{f1} \\ u_{m1} \\ u_{f2} \\ u_{m21} \\ u_{m22} \end{bmatrix}, \tag{5.3}$$

$$y_{end-line} = \begin{bmatrix} KPC_1 \\ KPC_2 \\ KPC_3 \end{bmatrix} \Gamma_{end-line} \cdot \begin{bmatrix} u_{f1} \\ u_{m1} \\ u_{f2} \\ u_{m21} \\ u_{m22} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} u_{f1} \\ u_{m1} \\ u_{f2} \\ u_{m21} \\ u_{m22} \end{bmatrix}. \tag{5.4}$$

Therefore, the $\Gamma$ matrix is:

$$\Gamma = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}. \tag{5.5}$$

## 5.4 Definition of the monitoring system

The purpose of the sequential inspection approach is to conduct the search for the root causes only when the process is detected to be statistically out of control. Up to this moment, only a minimum number of KPCs should be inspected, reducing the inspection costs. Therefore, it is important to define the minimum KPCs to be inspected in order to be sensitive to all sources of variation. In some MMPs, due to variation propagation, only the inspection of some KPCs at the end of line may be enough to have a good indicator about the general state of the process. If these KPCs are within statistical control, it can be assumed that all sources of variation are under admissible levels and no further inspections are required.

Given the qualitative model previously defined, the minimum monitoring system that includes the effects of all sources of variation can be derived through a basic search algorithm. Algorithm 1 shows the proposed search sequence to identify the minimum KPCs that are required to be monitored.

**Algorithm 1** Algorithm to define the minimum KPCs to be inspected for indirectly monitoring all sources of variation.

**while** $u_{detected} > 0$ **do**
  **for** $i = 1, ..., m$ **do**
    Evaluate the index $I_i = \sum_{j=1}^{n} \Gamma_{ij}$ for each $KPC_i$
  **end for**
  Select $KPC_i$ to be inspected with $I_i$ $max$
  **for** $j = 1, ..., n$ and $i|I_i = max(I)$ **do**
    **if** $\Gamma_{ij} = 1$, **then**
      $u_{detected} - 1$
    **end if**
  **end for**
  Remove the $jth$ columns of $\Gamma$ where $\Gamma_{ij} = 1$, $i|I_i = max(I)$,
  Remove the $ith$ row of $\Gamma$ where $I_i$ is max
**end while**

Given the set of KPCs to be inspected at the end of the line, a quality control system based on control charts can be built to monitor the state of the process. After setting the control limits of the control chart for each KPC, the monitoring system can be used to detect if the process is out of statistical control. See [118] for more details of setting control chart limits. At that moment, the sequential inspection procedure, derived in the following section, can be executed to detect the existing fault process.

## 5.5 Sequential inspection methodology

The sequential inspection methodology is based on the evaluation of the information gain every time an inspection is conducted, and the source of variation has not been identified yet. The proposed methodology is based on a sequential approach that has been successfully applied in the field of software testing [119, 120].

### 5.5.1 Bayesian approach for diagnostic explanation

The sequential inspection approach defines which sequential measurements along the process should be conducted based on the fault probabilities estimated by the Bayesian reasoning, which is updated after an inspection measurement is carried out.

The starting point is a set of diagnostic explanations that indicate which fault process may exist in the system, denoted as $D = \{d_1, ..., d_n\}$. Since it is assumed that only one fault is active

at the same time, $d_k$ refers to a specific process fault $u_k$ that is present in the system, thus, $D = \{u_1,\ ...,\ u_n\}$. The finite set of inspection measurements is defined as $Y = \{y_1,\ ...,\ y_m\}$, and the result of the inspection can be 0 (inspected feature is within statistical control) or 1 (the feature inspected is out of control). The result of the $y_i$ inspection is defined as $o_i$, and $o_i = 0$ or 1. The prior probability of the process fault is obtained according to maintenance data or, if it does not exist, an equal probability of all faults is given.

According to previous nomenclature, the prior probability of a diagnostic explanation where $u_k$ is faulty is

$$Pr(d_k) = Pr(u_k) = \frac{1}{n}, \tag{5.6}$$

if no maintenance data is applied.

In order to apply the sequential inspection procedure, the probability of this diagnostic explanation needs to be estimated if the inspection result from $y_i$ (i.e., $o_i$) is that the feature is out of control. Therefore, according to Bayes' rule:

$$Pr(d_k|o_i = 1) = \frac{Pr(o_i = 1|d_k)}{Pr(o_i)} \cdot Pr(d_k), \tag{5.7}$$

In this equation, $Pr(o_i = 1|d_k)$ represents the probability of the observed outcome, if that diagnostic explanation $d_k$ is the correct one, given by

$$Pr(o_i = 1|d_k) = 1 - Pr(o_i = 0|d_k) = \Gamma_{ik}, \tag{5.8}$$

Note that according to the qualitative model, if $u_k$ is faulty the ith inspection measurement will be out of control if $\Gamma_{ik} = 1$. The term $Pr(o_i)$ represents the probability of the observed outcome, independently of which diagnostic explanation is the correct one. The value of $Pr(o_i)$ is a normalizing factor that is given by

$$Pr(o_i) = \sum_{d_k \in D} Pr(o_i|d_k) \cdot Pr(d_k). \tag{5.9}$$

### 5.5.2 Prioritization based information gain

The prioritization of the inspection measurement is based on maximizing the Information Gain ($IG$) index defined by Johnson [121]. The IG is defined as

$$IG(D, y_i) = H(D) - Pr(o_i = 0) \cdot H(D_0) - Pr(o_i = 1) \cdot H(D_1), \tag{5.10}$$

where $D_0$ and $D_1$ represent the updated diagnosis explanation if inspection $y_i$ results in a feature within control or out of control, respectively. The entropy of a set of diagnostic candidates $D$, denoted as $H(D)$, is defined as

$$H(D) = -\sum_{d_k \in D} Pr(d_k) \cdot log_2(Pr(d_k)), \tag{5.11}$$

which can be understood as the average information we are missing until we can be certain about the diagnosis [119, 120]. Therefore, IG diagnostic prioritization integrates Bayesian diagnosis in

the inspection sequence selection and uses the information gain as the main indicator to express the diagnostic utility of executing a specific inspection measurement. From a isolation point of view, the best inspection to be conducted is the one that yields the highest IG.

The algorithm to be implemented for the sequential inspection procedure is shown in Algorithm 2.

**Algorithm 2** Algorithm for the sequential inspection procedure.

**Step 1:** Set the probabilities of process faults and the list of candidates.

Define $Pr(d_k)$.

Define the list of fault candidates $D = \{d_1, ..., d_k\}$.

(The following steps apply when the monitoring system reaches an out-of-control state)

**Step 2:** Update $\Gamma$ matrix.

Remove $ith$ rows from the monitored KPCs.

Remove the $jth$ columns of the $u_j$ whose KPCs are in-control.

Define the list of fault candidates $D$ according to the faults related to the KPCs that are out of control.

Update the probabilities of the remaining process faults candidates $Pr(d_k)$ according to the inspection results from the monitoring system.

**Step 3:** Calculate the IG for each potential inspection measurement $y_i(IG_i)$.

$IG_i = IG(D,\ y_i) = H(D) - Pr(o_i = 0) \cdot H(D_0) - Pr(o_i = 1) \cdot H(D_1)$

$H(D) = -\sum_{d_k \in D} Pr(d_k) \cdot log_2(Pr(d_k))$

$D_0$ is the updated $D$ value if yi is within control; $D_1$ if $y_i$ is out of control.

**Step 4:** Conduct the ith inspection measurement with the highest $IG_i$.

**Step 5:** According to the result of the inspection $i|IG_i = max(IG)$:

**if** the inspection result is within normal values **then**

Remove the $jth$ columns of the $\Gamma$ matrix where $\Gamma_{ij} = 1$.

Update the list of fault candidates $D$, removing those that are not related to the $ith$ inspection.

**end if**

**if** the inspection result is out of normal values **then**

Remove the $jth$ columns of the $\Gamma$ matrix where $\Gamma_{ij} = 0$.

Update the list of fault candidates $D$, removing those that are related to the $ith$ inspection.

**end if**

Remove the row of $\Gamma$ related to the $ith$ inspection that has been conducted.

Update the probabilities of the remaining process faults candidates $Pr(d_k)$ given the result of the inspection $y_i$.

**Step 6:** Repeat steps 3-5 until a fault is isolated.

### 5.5.3 Effectiveness of the IG approach

In order to analyze the effectiveness of the sequential inspection approach based on the IG versus an inspection approach based on random selection, let us consider a process with $n$ sources of

variation, and denote $\rho$ as the coverage density that indicates the coverage of each inspection with respect to the sources of variation, i.e., the inspection is related to $\rho \cdot n$ sources of variation. This coverage factor is applied throughout all the sequential process, thus each inspection will be able to detect $\rho \cdot n_r$ sources of variation, where $n_r$ is the remaining sources of variation that have not been discarded yet. According to [120], the IG index for a $\Gamma$ matrix with a coverage density $\rho$ is defined as

$$IG(\rho) = -\rho \cdot log_2(\rho) - (1-\rho) \cdot log_2((1-\rho)). \tag{5.12}$$

At this point, two extreme cases can be studied to analyze the effectiveness of the IG approach. First, the best-case scenario corresponds to a sequential inspection scheme where the sources of variation are split in two equal sets of fault candidates, i.e., when $\rho = 0.5$. Under this scenario, the IG index is maximum ($IG = 1$) and the average number of inspections required to detect the final fault is defined as $log_2(n)$.

Secondly, the worst-case scenario is when the inspections only detect the effect of one single fault. This case is given when the coverage density is $\rho = 1/n$ and thus, the IG is minimum. Under this scenario, the average number of inspections required is $(n-1)/2 - (1/n)$. It can be noted that in this worst case scenario, there is no benefit of using the IG index since all potential inspections present a minimum value of IG, and the resulting sequential inspection is equal to a random sequential inspection approach.

Figure 5.5 shows the expected evolution of the required number of inspections for a given coverage density $\rho$ under the sequential inspection approach based on the IG and based on a random selection. As it can be seen, the effectiveness of the IG approach increases when the coverage density increases. It is worth mentioning that MMP with a higher error propagation between stages present higher coverage densities and therefore, the IG index may have an important impact on sequential inspection approaches. Please, note that in Figure 5.5 the random selection curve refers to the worst case within the random selection approach considering that, besides the inspections according to the given $\rho$, additional inspections to check single faults are available. Therefore, the real average number of inspections required under the random selection for a given MMP is expected to be between this worst case curve and the IG curve and it will depend on the specific structure of $\Gamma$ matrix.

## 5.6 Case study

To illustrate the performance of this sequential inspection methodology for fault detection and isolation, let us consider the part shown in Figure 5.6 that is manufactured according to the process plan presented in Table 5.1. Tables 5.2 and 5.3 show the KPCs to be inspected and the on-machine measurements that can be conducted in the process. To evaluate the resulting cost of the inspection scheme, the inspection from KPC1 to KPC7 is set to 100 € and the inspection from KPC8 to KPC13 is set to 115 €. The costs for on-machine inspections are set to 85 €.

From the above process plan, the rooted tree shown in Figure 5.7 can be derived. As it can be seen, there are 11 potential process faults, 13 potential inspection measurements and 4

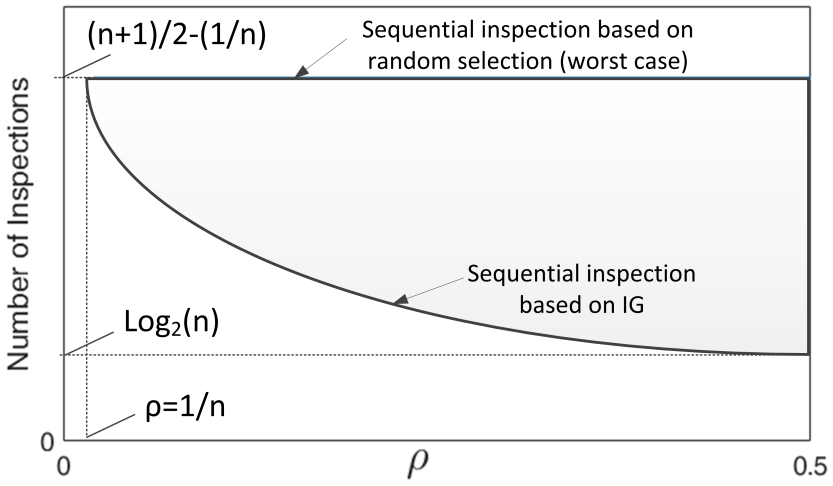



**Figure 5.5.** Expected evolution of the required average number of inspections for fault diagnosis under a sequential inspection based on IG and a random selection.

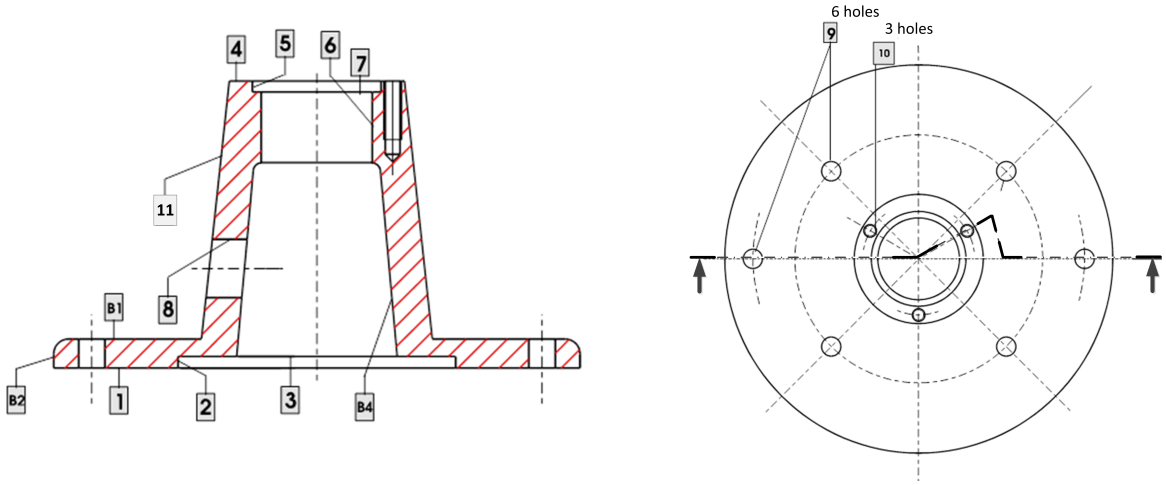



**Figure 5.6.** Part to be machined with numbered surfaces. B1, ..., means raw surfaces. 1, 2..., means machined surfaces and are referred in the text as S1, S2, etc.

on-machine measurements. From the rooted tree, the qualitative model that links process faults and inspection measurements is:

$$y_{on-machine} = \begin{bmatrix} y_{uf1} \\ y_{um21} \\ y_{uf3} \\ y_{uf4} \end{bmatrix} = \Gamma_{on-machine} \cdot u, \tag{5.13}$$

$$y_{end-line} = \begin{bmatrix} y_{uf1} \\ y_{um21} \\ y_{uf3} \\ y_{uf4} \end{bmatrix} = \Gamma_{end-line} \cdot u, \tag{5.14}$$

**Table 5.1.** Manufacturing process plan for the case study.

| | Stage 1 | Stage 2 | Stage 3 | Stage 4 |
|---|---|---|---|---|
| Machine-tool | Lathe | Lathe | Mach. center | Mach. center |
| Datum surfaces | B1, B2 | S4, S11 | S1, S2 | S1, S9 |
| Workholding system | 3-jaw chuck & positioners | 3-jaw chuck & positioners | 3-jaw chuck & positioners | 3 locators & concentric & radial locators |
| Mach. features | S5, S7, S4, S11 | S6, S2, S3, S1 | S9, S10 | S8 |

**Table 5.2.** KPCs for the part analyzed in the case study.

| KPCs | Characteristic | KPCs | Characteristic | KPCs | Characteristic |
|---|---|---|---|---|---|
| KPC1 | Distance S1-S4 | KPC5 | Distance S8-S1 | KPC9 | Position S9-S2 |
| KPC2 | Distance B1-S8 | KPC6 | Concentricity S2-S5 | KPC10 | Concentricity S11-S5 |
| KPC3 | Distance S7-S1 | KPC7 | Distance S4-S8 | KPC11 | Diameter S8 |
| KPC4 | Distance S10-S1 | KPC8 | Distance B1-S1 | KPC12 | Diameter S6 |
| | | | | KPC13 | Diameter S9 |

$$y_{inspection} = \begin{bmatrix} y_{on-machine} \\ y_{end-line} \end{bmatrix} = \begin{bmatrix} \Gamma_{on-machine} \\ \Gamma_{end-line} \end{bmatrix} \cdot u = \Gamma \cdot u, \tag{5.15}$$

where

$$\Gamma_{on-machine} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \tag{5.16}$$

$$\Gamma_{end-line} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{5.17}$$

According to Section 5.4, the KPCs that should be monitored to include all sources of variation are KPC2 and KPC3.

**Table 5.3.** Possible on-machine measurements.

| On-machine inspection | Characteristic |
|---|---|
| $y_{uf1}$ | Fixture inspection stage 1 |
| $y_{um21}$ | Tool inspection stage 2 |
| $y_{uf3}$ | Fixture inspection stage 3 |
| $y_{uf4}$ | Fixture inspection stage 4 |

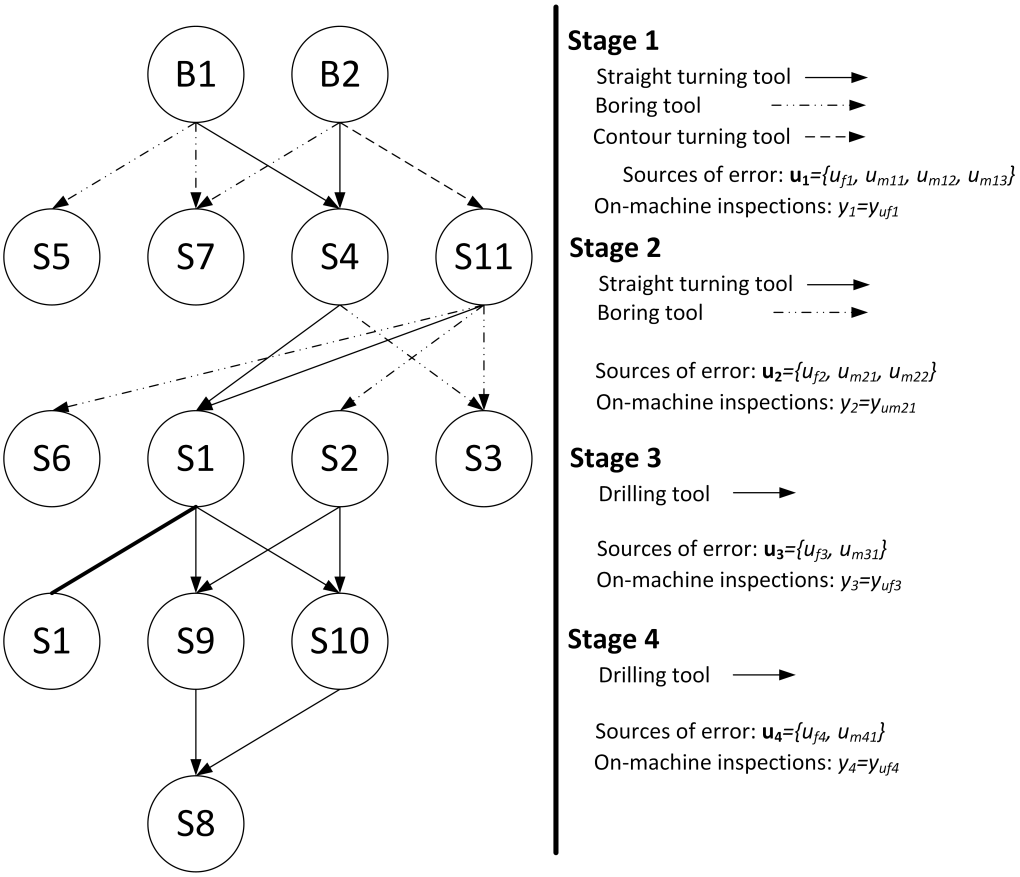



**Figure 5.7.** Rooted tree of the case study.

### 5.6.1 Fault detection and isolation results and discussion

In order to compare the performance of the proposed sequential inspection procedure, the number of inspections required to successfully detect the process faults under 3 different inspection schemes are compared:

1. A full inspection system in order to make the process faults fully diagnosable. In this case, there is no sequential inspection since the minimum number of KPCs to be inspected is always measured for fault detection. For this case study, to fully detect any process fault, the following inspections are required: KPC4, KPC6, KPC7, KPC8, and on-machine inspections in stage 3 and 4. Therefore, a total of 6 inspections are needed. Note that any

of the eleven potential faults can be fully detected and isolated by the combination of these 6 inspection measurements since the pattern fault defined by any of these 11 potential faults are different from each other.

2. A random sequential inspection procedure. In this case, the proposed sequential inspection process is applied but, instead of using the IG index, the KPCs to be inspected are randomly selected and the inspection result is used to reduce the potential process faults candidates of the system.

3. The proposed sequential inspection procedure, where the required inspection measurements are selected according to the IG index.

The comparison is conducted in terms of both costs and number of KPCs to be inspected before a fault detection and isolation is reached. For the first scheme (fully diagnosable system), the number of the KPCs needed is 6 as stated above. For the other two schemes, Monte Carlo simulations are evaluated where, at each simulation, a fault is added into the system and the sequential procedure is launched in order to finally detect it. The average number of inspections needed after 11,000 simulations is considered as the performance value for comparison purposes. Additionally, two situations are analyzed: a first situation where there is no information about the prior fault probability, thus, equal fault probabilities are assumed; a second situation where the information from maintenance data is used and then, the ratios of fault probabilities are known.

The results of the 3 schemes and the two situations are shown in Table 5.4. As it can be seen, the use of a sequential inspection procedure can reduce the number of inspection measurements needed with respect to a predefined inspection scheme. The fixed scheme requires a continuous inspection of 6 KPCs whereas the sequential inspection reduces the average number of measurements needed to less than 5, which means more than 15% of reduction. Furthermore, the use of a random search in the sequential approach can sometimes give a smaller number of inspections required, but taking into account the average from 11,000 simulations, the random approach requires more measurements than the sequential approach, 4.9 versus 4.2. Additionally, if the probabilities of process faults are known in advance, the IG algorithm can reach an average number of measurements of 4.0, slightly better than 4.2 that was obtained using equal probabilities of all process faults. Note that, for this case study, the number of process faults is not too large (only 11 faults), and the use of the prior probabilities from maintenance for a faster fault detection may not have a high impact. Comparing the predefined scheme with the sequential IG algorithm, the reduction of the number of inspections is from 6 to 4, which means a reduction of 33%. In terms of cost, the sequential approach based on the IG index can reduce the cost of inspection from 585 € to 389 €, which means a similar percentage of cost reduction.

[1] All process faults have random probabilities and these probabilities are not known, therefore all faults probabilities are set to equal probable (1/11) in the algorithm.

[2] All process faults have random probabilities and these probabilities are used in the IG algorithm.

**Table 5.4.** Number of inspection measurements required for fault identification and cost. Three schemes: fully diagnosable, proposed sequential inspection with random selection of inspections and proposed sequential inspection with IG index. Two situations: all fault probabilities are equal; fault probabilities are defined according to maintenance data. Note that prior fault probabilities information is only used in the proposed sequential algorithm.

| Prior fault probabilities | Fully diagnosable | Sequential inspection but random selection | Sequential inspection with IG index |
|---|---|---|---|
| Equally probable[1] | 6 (585 €) | 5.0 (508 €) | 4.2 (404 €) |
| Based on maintenance data[2] | 6 (585 €) | 5.0 (508 €) | 4.0 (389 €) |

## 5.7 Additional case studies for validation

The previous case study has shown the benefit of applying the proposed sequential approach based on IG for fault diagnosis in a 4-stage machining process. However, as it was pointed out in Section 5.5.3, the effectiveness of the methodology depends on the structure of the $\Gamma$ matrix, i.e, depends on the coverage density $\rho$. To have a more complete validation of the proposed methodology, two different scenarios are evaluated with a random generation of $\Gamma$ matrices.

For both scenarios, the number of sources of variation is set to $n = 18$, and the number of potential inspections is set to $m = 30$. For the first scenario, the $\Gamma$ matrix is randomly generated forcing that 4/5 of the inspections present a $\rho$ of 0.5 and a 1/5 of the inspections present a $\rho$ of 0.1. This $\Gamma$ matrix is considered a high density matrix which would be the result of a MMP with a high error propagation via datums. The second scenario presents a $\Gamma$ matrix randomly generated where the 4/5 of the inspections present a $\rho$ of 0.1 and a 1/5 of the inspections present a $\rho$ of 0.5. This is an opposite scenario where a low error propagation exists along the MMP. Both scenarios are compared in terms of number of inspections required and total cost of the inspection scheme for fault diagnosis. The cost of each inspection is randomly set to $100 \pm 20$ € using a uniform distribution. All sources of variation have the same a priori probability of failure.

As it is shown in Table 5.5, the results validate the proposed methodology since the reduction of number of inspections and cost is relevant. However, as it was pointed out in Section 5.5.3, the benefit of the methodology increases when $\Gamma$ matrix presents a higher coverage density. In this case study, the reduction of number of inspections for the first scenario (a process with high error propagation and therefore higher $\rho$ values) is 55% (from 1158 € to 521 €), whereas the reduction in the second scenario (a process with less error propagation and therefore lower $\rho$ values) is only 13% (from 512 € to 445 €).

## 5.8 Conclusions

Sequential inspection in MMPs can be of interest to reduce the inspection cost and provide fast fault detection and isolation. This chapter has proposed a methodology to implement a

**Table 5.5.** Average number of inspections required and cost for two different scenarios applying a sequential inspection based on IG index and random selection.

| | Sequential inspection with random selection | Sequential inspection with IG index |
|---|---|---|
| First scenario (MMP with high error propagation) | 11.5 (1158 €) | 5.2 (521 €) |
| Second scenario (MMP with low error propagation) | 5.4 (512 €) | 4.7 (445 €) |

sequential inspection procedure based on the information gain index of the inspection measurement. To evaluate this index, a qualitative model that links the sources of variation with the KPCs is derived. The methodology is composed of three parts. A qualitative model extracted from process planning; a monitoring system to detect if the process is out of statistical control; and a sequential inspection procedure applied for a fault detection and isolation search which is based on maximizing the information that can be obtained from a specific set of inspection measurements.

The proposed methodology has been theoretically analyzed showing that the IG algorithm can highly reduce the number of inspections required when the MMP presents a high error propagation behavior, which means that the coverage density of the inspections tend to be high. Otherwise, when the MMP presents low error propagation and the coverage density of the inspection is low, the benefit of maximizing the IG instead of a random selection is lower. A more specific MMP based on four machining stages was also analyzed to prove the generation of the quality model through the graphical representation of the MMP and the reduction of the inspections required when a sequential inspection approach based on IG is applied. In this case study, a reduction of 33% in the inspection effort and cost was obtained with respect to common practices where a predefined inspection scheme for fault detection and isolation is given.

Chapter 6

# Model-based observer proposal for surface roughness monitoring


## Abstract

In the literature, many different machining monitoring systems for surface roughness and tool condition have been proposed and validated experimentally. However, these approaches commonly require costly equipment and experimentation. In this chapter, we propose an alternative monitoring system for surface roughness based on a model-based observer considering simple relationships between tool wear, power consumption and surface roughness. The system estimates the surface roughness according to simple models and updates the estimation fusing the information from quality inspection and power consumption. This monitoring strategy is aligned with the industry 4.0 practices and promotes the fusion of data at different shop-floor levels.

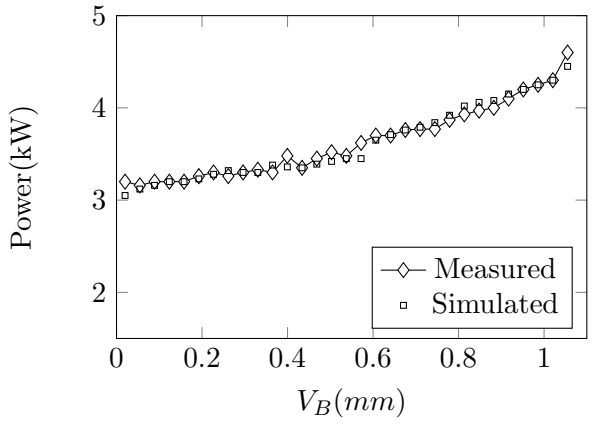

## 6.1 Introduction

Machining monitoring systems have been an important topic of research for decades with important contributions in the field. In the literature, a large number of monitoring systems have been proposed and validated experimentally, especially for surface roughness prediction and cutting tool wear estimation. In surface roughness prediction, monitoring systems seek to estimate surface roughness according to cutting conditions and real-time measurements on forces, vibrations, temperatures, or current/power consumptions. These performance indicators may partially explain the quality of the machined surface and using a proper Design of Experiments (DoE), a mathematical model may be obtained. For instance, Pimenov et al. [122] proposed the use of artificial intelligent methods for real-time prediction of surface roughness considering the main drive power and current machining time. Different models based on regression trees and artificial neural networks (ANN) models (Multilayer Perceptron –MLP– and Radial Basis Function–RBF–) were tested proving the use of drive power for surface roughness prediction. The authors observed a linear relationship between drive power and roughness in small ranges

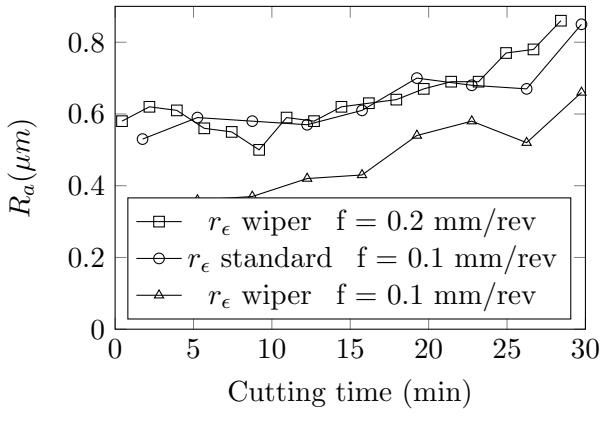
of processing time and, when exceeding a specific processing time, the drive power had to be carefully monitored due to its strong influence on surface roughness. In [123], the authors proposed an Adaptive Control Optimization (ACO) system based on a dynamometer and ANN models to estimate both cutting tool wear and surface roughness in micro-milling operations. Under the ACO proposed, the cutting conditions were changed in real-time according to the current tool state in order to ensure part quality with minimum cost. Since the system requires an efficient optimization procedure to be conducted in real-time, the authors compared the performance of different optimization approaches such as particle swarm optimization (PSO), genetic algorithms (GA) and simulated annealing (SA) in terms of accuracy, precision, and robustness. In [124], the authors analyzed the correlation of surface quality with cutting force, vibration signals and acoustic emission signals, applying fusion data methods and ANN models.

The research on un-manned machining systems has also led to the development of a large number of tool condition monitoring systems based on different sensor systems. A detailed explanation of the components of monitoring system such as sensor systems, signal processing, feature extraction methods, and modeling tools (e.g. regression models, artificial intelligent models) can be found in recent review works [125–127]. However, monitoring systems require models previously obtained through Design of Experiments (DoE) methods that are usually costly and time consuming, and in some cases the modeling tools are too advanced for being applied in industrial environments. Furthermore, most of the proposed systems are based on costly/invasive systems or unfeasible experimental practices which prevent their implementation in real environments.

For improving monitoring systems, straightforward relationships that are well-known in the literature could be used. For instance, different researches have tested the close relationship between power consumption and tool wear [36] (see Figure 6.1a) and the relationship between tool wear and surface roughness is also commonly identified as a key factor for roughness estimation [128, 129] (see Figure 6.1b).




**(a)** Relationship between power consumption and tool flank wear in [36].




**(b)** Relationships between surface roughness and cutting time (tool wear) in [129].

**Figure 6.1.** Relationship between different machining parameters, adapted from [36] and [129].

In this chapter, we have proposed an alternative monitoring system for surface roughness

based on a model-based observer considering simple relationships between tool wear, power consumption and surface roughness. The system estimates the surface roughness according to simple models and updates the estimation fusing the information from quality inspection and power consumption every inspection sampling. Model-based observers such as Kalman filters have been successfully applied for tool wear monitoring [130–132], but their use for improving surface roughness monitoring systems fusing inspection data and sensor data has not been yet investigated. This monitoring strategy is aligned with the industry 4.0 practices, where the increase of the interconnectivity of different equipment in the shop-floor may promote the fusion of data from different nature.

This chapter is organized as follows. Section 6.2 presents the proposed monitoring systems based on simple models and sampling information from power sensors and inspection measurements. Section 6.3 mathematically explains the derivation of the model-based observer using steady-state Kalman filters for surface roughness estimation based on data fusion. Section 6.4 shows the application of the methodology in terms of a series of simulations and Section 6.5 concludes the chapter.

## 6.2 Methodology

The proposed monitoring system is based on two sensors which provide information about the state of the cutting process and the quality of the machined parts. The first sensor is a non-invasive and low-cost power sensor, which provides information about the average power used during each machining process. This measurement is available during the cutting process, but the reliability of this measurement is low because of its uncertainty due to measuring noise. The second sensor, a profilometer, measures the surface roughness. This measurement is executed during the inspection procedure of the machined parts, which are conducted according to the sampling scheme adopted in the company, i.e. one part inspected every N manufactured parts (from now on, the information obtained with each inspected part will be known as *a sample*). This measurement provides information about the part quality and, at the same time, it indirectly gives information about the cutting tool wear state.

The information provided by both sensors is subsequently fused using a model-based observer in order to improve the surface roughness prediction. The benefit of using a model-based observer in the monitoring system makes possible to use this low-cost monitoring system even if the previous models built for the system have low accuracy since the fusion of the information will correct any deviation from the models up to certain point.

Figure 6.2 describes the proposed system for improving surface roughness prediction using a model-based observer. As shown there, the system uses both the information of the surface roughness and the power consumption as well as a theoretical model of the behavior of these variables when the tool wear increases. The proposed methodology is based on the following assumptions:

- The information from quality inspection and the machining process is straightforward thanks to the interconnectivity of the different areas at the shop-floor (industry 4.0 practices are moving towards this paradigm). Thus, the inspection values, which are available at a certain frequency, can be added into a monitoring system to fuse this information with power sensor measurements.

- The data provided by the power sensor and the inspection station has the same frequency, which means that both data are fused for better surface roughness estimation, but there is no information between samples. The use of different frequencies, for instance, a real time measurement from the power sensor and sampling measurements from inspection, which is a more common approach in industry, is out of the scope of this study and will be considered in future work.

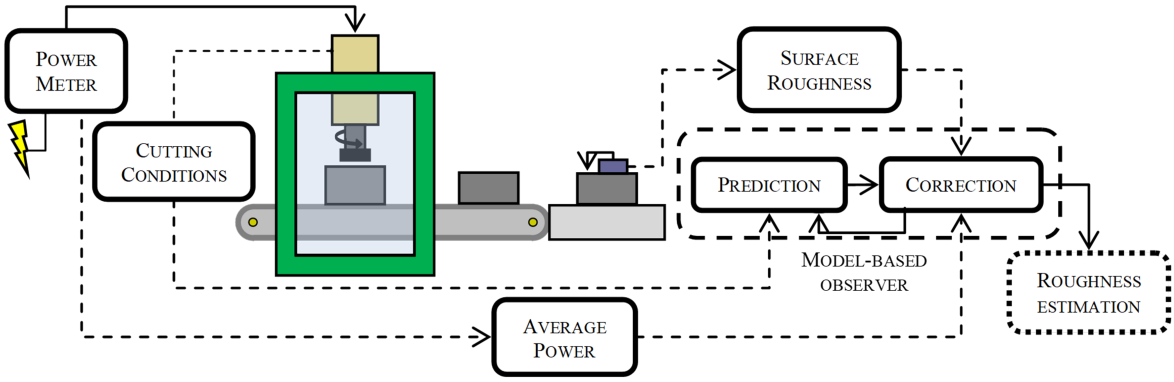



**Figure 6.2.** Diagram of the proposed monitoring system.

## 6.3 Obtaining a model-based observer

As seen in Section 6.1, the evolution of the surface roughness $R_k$ and the power consumption $P_k$ usually follows a certain behavior, consisting of a gradual increase (let us call it $\Delta R_k$, $\Delta P_k$) from a base or nominal value ($f_R$, $f_P$). Furthermore, the available measured information (that we call $R_{m,k}$, $P_{m,k}$) is affected by some measuring noise ($v_{R,k}$, $v_{P,k}$). With this, we model the measurement of the surface roughness and power consumption as

$$\begin{cases} R_{m,k} = \Delta R_k + f_R + v_{R,k}, \\ P_{m,k} = \Delta P_k + f_P + v_{P,k}. \end{cases} \tag{6.1}$$

The increases $\Delta R_k$ and $\Delta P_k$ from the base value are modeled as monotonic increasing functions, that we propose to model as constant slopes (being the respective slopes $\delta_R$ and $\delta_P$) as follows:

$$\begin{cases} \Delta R_k = \Delta R_{k-1} + \delta_R + \omega_{R,k}, \\ \Delta P_k = \Delta P_{k-1} + \delta_P + \omega_{P,k}. \end{cases} \tag{6.2}$$

Elements $\omega_{R,k}$ and $\omega_{P,k}$ are zero mean random signals that represent deviations of the behavior from that slope, i.e., the uncertainty of the proposed model. As we will detail later, this allows us to model other behaviors different from the proposed one thanks to a right tuning of the available parameters.

The previous proposed model can be interpreted as a state-space model, a standard modeling found in control theory. Considering the increases $\Delta R_k$ and $\Delta P_k$ as inner states and $R_{m,k}$ and $P_{m,k}$ as the measurable outputs of the system, the space-state representation of the system takes this form:

$$\begin{cases} x_k = Ax_{k-1} + B\delta + G\omega_k, \\ y_k = Cx_k + f(u_k) + v_k, \end{cases} \tag{6.3}$$

being

$$x_k = \begin{bmatrix} \Delta R_k \\ \Delta P_k \end{bmatrix},\ \delta = \begin{bmatrix} \delta_R \\ \delta_P \end{bmatrix},\ \omega_k = \begin{bmatrix} \omega_{R,k} \\ \omega_{P,k} \end{bmatrix},\ y_k = \begin{bmatrix} R_{m,k} \\ P_{m,k} \end{bmatrix},\ f(u_k) = \begin{bmatrix} f_{R,k} \\ f_{P,k} \end{bmatrix},\ v_k = \begin{bmatrix} v_{R,k} \\ v_{P,k} \end{bmatrix},$$

where $f(u_k)$ is the base or nominal value, which depends on several cutting conditions denoted by $u_k$, and where $A = B = G = C = I$, i.e., the Identity Matrix.

Based on these assumptions, we have developed and implemented a model-based observer that will allow the prediction and estimation of the outputs and inner states even when there are not available measurements.

The application of this observer has two steps, the prediction and the correction step. At the prediction step, the surface roughness and power consumption values are predicted. There, the predicted states $\hat{x}_k^-$ are considered to be the same as the previous corrected state $\hat{x}_{k-1}$, plus an increment $\delta$, as seen in the previous model. At this point, the prediction of the values that will be measured are also estimated (they are called $\hat{y}_k^-$), and they are obtained as the sum of the predicted states $\hat{x}_k^-$ and an estimation of the nominal values ($f_0(u_k)$), following the previous models, and assumed zero-mean measurement noise. Therefore, the predictions of he inner states $\hat{x}_k^-$ and the outputs $\hat{y}_k^-$ are defined as:

$$\begin{cases} \hat{x}_k^- = \hat{x}_{k-1} + \delta, \\ \hat{y}_k^- = \hat{x}_k^- + f_0(u_k). \end{cases} \tag{6.4}$$

At the correction step, the observer corrects the predicted states $\hat{x}_k^-$. The corrected states $\hat{x}_k$ consist of the sum of the predicted state and a correction term. The correction term consists of the difference between the measured values $y_k$ and the predicted ones $\hat{y}_k^-$ multiplied by a correction gain, named $L$. The corrected value of the measurements, $\hat{y}_k$, is defined as the sum of the function $f_0(u_k)$ and the corrected states $\hat{x}_k$. Therefore:

$$\begin{cases} \hat{x}_k = \hat{x}_k^- + L\left(y_k - \hat{y}_k^-\right), \\ \hat{y}_k = \hat{x}_k + f_0(u_k). \end{cases} \tag{6.5}$$

Under this observer, the surface roughness predicted by the system is based on the data fusion from the simple model based on increments due to tool wear and the sensor measurements ($y_k$). This simple model carries an important modeling error as the real behavior of the system is far more complex. The sensor measurements provide data based on the sampling scheme. It must be taken into account that the provided data is not perfect due to measurement noise. Using all this information, the model-based observer is able to provide better surface roughness estimations, especially in the periods when no data is available.

The key parameter of the model-based observer for an adequate fusion scheme is the $L$ correction gain matrix. It can be obtained via several methods, such as pole placement or optimal estimation. In this case, we have chosen to implement a Kalman Filter.

As we assume that neither the variance of the measurement noise nor the variance of the tool wear change over time, we used a specific variant of the Kalman Filter: the Steady-State Kalman Filter (SSKF), which only requires an initial single calculation of the $L$ correction matrix (opposed to the complete Kalman Filter option, which would require one in each cycle).

This calculation can be performed using the Matlab function *dlqe*, which designs a Kalman estimator for discrete-time systems. This function calculates $L$ using the space-state matrices $A$, $C$ and $G$, as well as the covariance matrices $V = E\{v_k^\intercal v_k\}$ and $W = E\{\omega_k^\intercal \omega_k\}$, which include the variances of the measurement noises $v_k$ and the variances of the zero-mean signal $\omega_k$ (which represents the base model deviation due to wear and model error), respectively.

The dimensions of the obtained $L$ matrix are $N°states \times N°measurements$. In this case, the resulting matrix is square, and its values depend on the relative values assigned to the measurement noise variance and the model deviation variance. As the true behavior of the model deviation is unknown, the values of $W$ are used as tuning parameters. Matrix $W$ is symmetric and presents the form

$$W = \begin{bmatrix} \omega_{11} & \omega_{12} \\ \omega_{12} & \omega_{22} \end{bmatrix}, \tag{6.6}$$

where the diagonal terms refer to the uncertainty of the model for each submodel (roughness and power consumption), and where the non-diagonal terms refer to the correlation within these two uncertainties. The relative values between the different elements in $W$ and also w.r.t the values in V, determine the behavior of the observer during the initialization, and how it weights differently the measurements and the predictions in the correction step.

## 6.4 Case study

### 6.4.1 Real model simulations

In order to validate the capability of the model-based observer for improving surface roughness estimations, a case study is analyzed by a set of simulations. For this case study, we assume that the real behavior of the surface roughness follows the equation shown below from Kovac et al. [37]

$$R = 10.916\ V_c^{-0.894} f_z^{-0.046} a_a^{-0.015} V_B^{0.456}, \tag{6.7}$$

where $V_c$, $f_z$, $a_a$ and $V_B$ refer to cutting speed, feed per tooth, axial depth of cut and flank wear value, respectively. Furthermore, we assume a proportional increase of power consumption with respect to tool wear as

$$P = P_0 + \alpha V_B P_0, \tag{6.8}$$

where $\alpha$ is the proportional coefficient, assumed to be $\alpha = \frac{0.15}{0.4} = 0.375$, which means an increase of 15% when there are 0.4 $mm$ of tool flank wear. $P_0$ is the power consumption when new inserts are used, and it is a function of several cutting parameters. For this case study, we assume that $P_0$ follows the behavior shown in [133]:

$$P_0 = 6127 - 0.42V_c - 3616f_z + 83.1V_c f_z. \tag{6.9}$$

Finally, the tool wear behavior is assumed to follow a third order equation with respect to the time variable, as suggested in most of the machining handbooks [134]. Since the time variable is related to the number of the parts that have been processed, we assume the following relationship:

$$V_B(k) = 1.5\left(\frac{k}{k_{lim}}\right)^3 - 1.915\left(\frac{k}{k_{lim}}\right)^2 + 0.815\left(\frac{k}{k_{lim}}\right), \tag{6.10}$$

where $k$ is the number of parts processed up to this moment, and $k_{lim}$ is the total number of parts processed when the tool flank wear reaches 0.4 $mm$. Note that the relationships shown above are used for simulating the machining process and they are unknown for the model-based observer.

### 6.4.2 Proposed estimations

In order to study the performance of the model-based observer for improving surface roughness estimations, we have analyzed three different cases. First, we have considered an "off-line system", which consists of estimating the surface roughness as an open-loop system, only assuming that the system behaves as a simple first order slope model. The second case consists of using a SSKF that only monitors the power consumption, and the surface roughness is estimated depending on the variations of the power consumption (i.e. considering that all variations of the power consumption affect proportionally to the surface roughness). The measurement of power consumption is conducted each 10 processed parts. The third case also consists of using a SSKF, but in this case, both the power consumption and the surface roughness measurements (also conducted each 10 processed parts) are known. In this case, the roughness data should help to reduce any initialization errors.

For comparison purposes, all strategies are analyzed when the machining process is conducted at the following cutting conditions: $V_c = 150\ m/min$, $f_z = 0.05\ mm/tooth$, $a_a = 1\ mm$. For these cutting conditions and given the real behavior of the machining process defined by equations (6.7) - (6.10), the increment of power consumption and surface roughness when the tool flank wear reaches 0.4 $mm$ is 7470 $W$ and 3.49 $\mu m$, respectively. Thus, $\delta_P$ and $\delta_R$ are obtained as:

$$\begin{cases} \delta_P = \frac{P_{worn} - P_{new}}{k_{lim}} = \frac{7470-6510}{260} = 3.7\ W/piece, \\ \delta_R = \frac{R_{worn} - R_{new}}{k_{lim}} = \frac{3.49-1.00}{260} = 0.0096\ \mu m/piece, \end{cases} \tag{6.11}$$

where $R_{new} = 1.00\ \mu m$ is approximately the surface roughness for the given cutting conditions and $P_{new} = 6510\ W$ the power consumption when the cutting tool is new. Note that in Equation (6.7), when $V_B$ tends to zero, the real surface roughness value is fixed to 1.00 $\mu m$. Note that both $R_{new}$ and $P_{new}$ are used to define the term $f_0(u_k)$ in Equations (6.4) and (6.5).

For this case study, the measurement noise from power sensor and profilometer is assumed to be Gaussian with $\pm 3\sigma$ bounds given by $n_P = \pm 300\ W$ and $n_R = \pm 0.63\ \mu m$, respectively, thus $V = \left[10^4, 0; 0, 0.044\right]$. The model divergence variance $W$ is tuned as $W_2 = \left[0.104, 19.13; 19.13, 1.141 \cdot 10^4\right]$ and $W_3 = \left[0.104, 38.26; 38.26, 1.41 \cdot 10^4\right]$ for cases 2 and 3, respectively, as they provided proper results.

In the "off-line" case, the estimation of surface roughness follows the simple model and no information used from sensors. In the second case, as the surface roughness is not measured, the $L$ gain matrix is obtained with a SSKF, but forcing to 0 the terms that use the surface roughness. Therefore, the $L$ gain matrix used is:

$$L = \begin{bmatrix} 0 & 7.98 \cdot 10^{-4} \\ 0 & 0.294 \end{bmatrix}. \tag{6.12}$$

For the third situation, as it uses the power sensor and the profilometer information, the $L$ gain matrix is:

$$L = \begin{bmatrix} 0.733 & 1.832 \cdot 10^{-4} \\ 41.22 & 0.644 \end{bmatrix}. \tag{6.13}$$

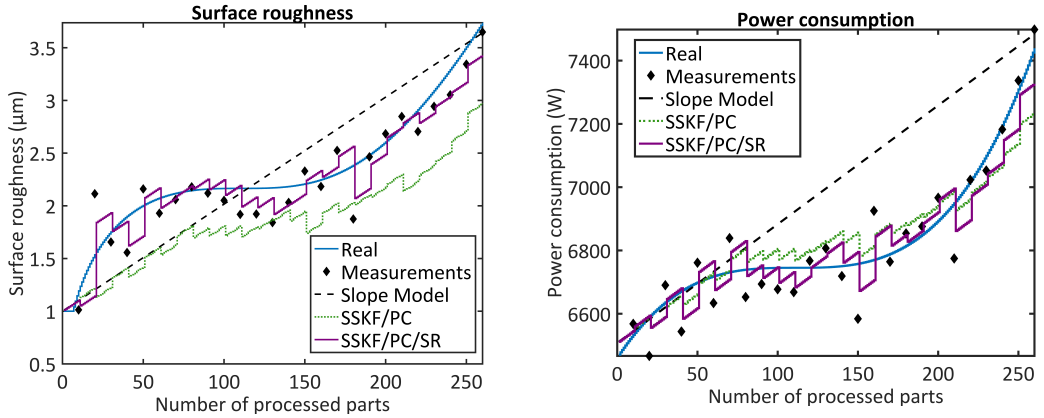
### 6.4.3 Results

In this section, we will explain the results of the simulations. First, we have designed an experiment, which consists of the simulation of the "real" behavior of the tool parameters (Equations (6.7) - (6.10)) and the application of the three considered cases. In each experiment, 260 pieces are processed, which is the limit for a proper tool flank wear. For comparison purposes, we have calculated the maximum prediction error and the root-mean-square error (RMSE) after applying all the proposed prediction models of the surface roughness. Applying a Monte Carlo method (as the noise is randomly added each time), each experiment has been repeated $10^6$ times, and the results can be observed in Table 6.1. As shown in Table 6.1, the model-based observer which uses both measurements has a far lower RMSE and maximum prediction error. It is also worth noting that the maximum prediction error is almost equal for the slope model and the model that only uses the information from the power sensor.

**Table 6.1.** Comparison of prediction errors for the three analyzed situations. MBO stands for Model-based observer, P for power and R for roughness.

| | Off-line system (Slope model) | MBO (SSKF) (P sampling) | MBO (SSKF) (P & R sampling) |
|---|---|---|---|
| Max. pred. error ($\mu m$) | 0.5829 | 0.5857 | 0.4297 |
| RMSE ($\mu m$) | 5.8583 | 4.5569 | 2.7588 |

After executing the simulations, we have obtained Figure 6.3a and Figure 6.3b. As shown in both figures, the off-line slope fails to predict the whole behavior of both signals, so this model cannot be reliably used to predict the behavior of the surface roughness along the whole cutting tool life.




**(a)** Surface roughness predictions.

**(b)** Power consumption predictions.

**Figure 6.3.** Predictions obtained using the proposed model-based observer.

The second case uses a SSKF and only the power consumption measurements (SSKF/PC). While it seems to follow properly the real behavior of the power consumption –as it only gets information each 10 parts, and it contains noise– (Figure 6.3b), it is not accurate enough in the case of the surface roughness (Figure 6.3a). The observer is able to predict the surface roughness behavior but there are zones where the prediction is far away from the real values. Its precision also depends on how similar are the behavior of both the surface roughness and the power consumption. Lastly, the third situation uses a steady-state Kalman filter with both the power consumption and the surface roughness measurements (SSKF/PC/SR). This prediction model uses the information that gets from those two sensors every 10 processed parts. In this case, it is shown in both figures that it follows the real model quite accurately as the observer fuses the information of both measurements to get a better estimation.

## 6.5 Conclusions

Surface roughness monitoring is a critical issue to optimize cutting parameters and ensure product specifications. Current monitoring systems do not consider the potential use of both sensor data from machine-tools and sampling measurements from part quality inspection to improve current surface roughness estimations. In this chapter we have proposed a monitoring system where data from power sensors and inspection measurements are fused using a model-based observer. This first work has validated the applicability of model-based observers for improving surface roughness monitoring system under a series of simulations.

As future work, the effect of tuning the gain matrix $L$ on surface roughness monitoring will be discussed and the influence of the sampling frequency on the fusing scheme will be analyzed. Furthermore, the use of sensor data with different frequencies of sampling will be studied.

Chapter 7

# Model-based tool condition prognosis using power consumption and scarce surface roughness measurements


## Abstract

In machining processes, underusing and overusing cutting tools directly affect part quality, entailing economic and environmental impacts. In this chapter, we propose and compare different strategies for tool replacement before processed parts exceed surface roughness specifications without underusing the tool. The proposed strategies are based on an online part quality monitoring system and apply a model-based algorithm that updates their parameters using Adaptive Recursive Least Squares (ARLS) over polynomial models whose generalization capabilities have been validated after generating a dataset using theoretical models from the bibliography. These strategies assume that there is a continuous measurement of power consumption and a periodic measurement of surface roughness from the quality department (scarce measurements). The proposed strategies are compared with other straightforward tool replacement strategies in terms of required previous experimentation, algorithm simplicity and self-adaptability to disturbances (such as changes in machining conditions). Furthermore, the cost of each strategy is analyzed for a given benchmark and with a given batch size in terms of needed tools, consumed energy and parts out of specifications (i.e., rejected). Among the analyzed strategies, the proposed model-based algorithm that detects in real-time the optimal instant for tool change presents the best results.


## 7.1 Introduction

Machining processes are manufacturing processes frequently used in industry where excess material from the surface of a workpiece is removed using different cutting tools. This removal

process causes an increase of tool wear and when it reaches a certain severity, it deteriorates both the macrogeometry (dimensions out of the required tolerances) and the microgeometry (surface roughness values) of the processed parts. In practice, the process may no longer produce acceptable parts when surpassing an admissible tool wear, and parts out of specifications may also need to be reprocessed or discarded, leading to the corresponding increase in costs. Besides, a heavily worn tool may lead to its complete breakage, which can cause higher machining downtime, potential damage to the machining center and, without the appropriate safety systems, may cause personal damage.

According to the bibliography [135], cutting tool failures may represent about twenty percent of the downtime of a machining system and it is estimated that the expense of cutting tools and their maintenance grosses about three to twelve percent of overall manufacturing cost [136,137]. In order to avoid these issues, early tool replacement strategies are commonly applied in industrial shopfloor with the subsequent increase in costs due to higher machining downtime for tool replacement, lower productivity and higher cutting tool costs.

Under these challenges, a robust and reliable online tool condition monitoring (TCM) system with an adequate online remaining useful life (RUL) estimation for proper tool replacements is crucial in industrial applications. TCM techniques estimate the current state of the cutting tool where the type of wear that is usually monitored is the tool flank wear since it is the type of wear that mainly affects surface roughness and dimensional quality in machining systems [138]. Tool condition can be monitored directly, by observing the tool, or indirectly, using available measurements from the machining process. Since direct methods require stopping the machining process to measure or inspect the tool, the research has been mostly oriented towards developing indirect monitoring [33]. For example, recent research has estimated the deterioration level of a tool using the applied forces during the machining process, and has used neural networks to differentiate the effect of the tool wear and other tool deterioration forms [139]. Additionally, in [140], physics guided neural networks have been developed to predict the state of the tool wear using deep learning techniques supported by physical equations.

Within TCM, RUL methods are focused on the prognostics of the remaining life of the tool, which lead to conduct efficient cutting tool replacement strategies considering the uncertainty of the process and confidence intervals.

RUL methods are classified as physics-based, data-based and model-based [16]. Physics-based approaches directly use formulae from theory, such as Taylor's tool life equation [34] or other more sophisticated ones [141], to estimate the remaining useful life of the cutting tool. Data-based approaches can mainly be classified into statistical methods and artificial intelligence methods [142]. In statistical methods, researchers use failure data from plenty of tests and apply statistical criteria to choose the best fit statistical distribution to get distribution of lifetime. A thorough review of statistical data-based approaches can be found in [143]. Autoregressive moving average models (ARMA) and logistic regressions are common techniques applied in this field [144]. In artificial intelligence methods, techniques such as artificial neural networks (ANN) [145], Support Vector Regression (SVR) [146], Adaptive Neuro-Fuzzy Inference Systems (ANFIS) [147] or Fuzzy systems [148] have been investigated. The tendency during the latest years is the research of deep learning techniques [149–151].

According to [142], methods based on physics, mechanics and dynamics may become more intractable because of the high complexity of the life prediction theory and the error of model prediction may increase with the enhancement of model nonlinearity and complexity. Data-driven approaches for tool wear prediction have demonstrated satisfactory accuracy for tool replacement in different machining applications such as milling, turning, and grinding. However, these approaches require sufficient historical data for training, the accuracy is highly affected by the sensor noise and measurement uncertainties [152] and the networks are suitable only under specific cutting conditions; if any of those conditions change, they should be retrained and thus, they cannot adapt to sudden changes nor natural degradation of the process [153].

Unlike the physics- and data-based approaches, the model-based approach is a more appropriate approach for tool wear estimation since it can be considered as a hybrid approach between physics-based and data-based methods [152]. Model-based approaches are based on stochastic methods where the tool wear state cannot be directly measured and it is estimated or predicted from online measurements, in which Bayesian inference provides a rigorous mathematic framework. The physical knowledge that defines the tool wear growth is included into the model in the form of a state-space model to represent the evolution of tool wear with time and the estimation of tool wear is updated using new online measurements. The main benefit of model-based approaches is that it needs less data because it is modeled with certain knowledge and assumptions of the tool wear degradation process [154]. Depending on system type and noise assumption, different approaches have been investigated such as Hidden Markov Models (HMM) [155, 156], State-Space Models (SSM) with Kalman filters [157], SSM with particle filters [158] and SVR applied to a physics-based tool condition degradation model [159]. Additionally, other types of hybrid approaches have been investigated, such as fusing ANN with Wiener processes [160] or Gauss importance resampling particle filters [161], or using multiple-scenario calibration methods [162].

Some of these model-based research works overcome previous RUL limitations and present a feasible industrial solution where minimum invasive sensor systems and minimum experimentation are applied. For instance, the authors in [157] proposed a model-based system to estimate flank wear through a Kalman filter. Tool wear is estimated using a state-space model under a linear function respect to the removed material volume. Kalman filter corrections are based on the grey level average from processing an image of the machined surface. In [163], the authors modeled tool wear evolution through a linear empirical function w.r.t material removal rate. This function was updated with an Extended Kalman Filter that used spindle power consumption and compared its performance with deterministic methods. In [152], the authors proposed the use of a third order empirical wear-time model as a state-space model for tool wear, and spindle motor current was used to infer the tool wear state. The measurement model was built using ANFIS techniques, and Particle Filtering was applied to update the algorithms instead.

One of the main problems of these works is the inability to adapt in front of behavioral changes, such as modification of cutting conditions and variations in the workpiece materials. This happens due to the fact that once trained, the models cannot be changed, as their parameters are fixed. Further research has dealt with this issue proposing model-based approaches with updating algorithms. For instance, the authors in [158] used Paris' law [164] as the model, and

used Kullback–Leibler divergence from several sensor signals to carry out the update during the first cuts in order to get more reliable predictions towards the end of the tool life. In [154], the authors used a first order linear function to model tool wear, and the model was updated using a linear regression from the measured RMS vibration signal. It also underlined the presence of tool-to-tool stochastic variations, as under identical workpiece and cutting conditions, model parameters changed slightly between tools. A more advanced work is presented in [153], which uses a similar model approach as [158]. Their authors proposed the use of autoregressive models trained with historical data in order to make estimations when sensor measurements are not available. These approaches, however, require a learning period during the first stages of each cutting tool life where no prediction can be carried out.

To the best of authors' knowledge, no model-based prognosis system has been presented with the following key characteristics for RUL under Industry 4.0 manufacturing paradigm: 1) A non-intrusive low-cost monitoring system, easy to install; 2) with minimal experimental data or even without the need of previous experimentation; 3) with the ability to learn, adapt and self-adjust depending on shopfloor data from the machining center or other equipment; 4) and being able to take advantage of Industry 4.0 capabilities, where connectivity between equipment allows instant availability of measurements throughout shopfloor. A recent research [165] considers the connectivity of the equipment at the shop-floor event to conduct the monitoring and RUL prediction online. Multi-source events are used to consider the right time to trigger the monitoring system, avoid the use of large volume of unwanted data. However, the use of data from inspection for triggering the system and improve the model is overlooked, as it mainly focuses on the connectivity frame.

The objective of this chapter is to propose two main approaches that fulfill previous key characteristics for RUL systems and lead to an optimal tool change under any cutting conditions, and compare them with simpler straightforward techniques. Unlike previous works, tool wear is not directly estimated since, in many finishing operations, tool change is conducted when the surface roughness of a processed part reaches an unacceptable value instead of a specific tool wear value.

In this system, the conducted measurements are a continuous measurement of power consumption at the machine-tool level, and a surface roughness value after a processed part is inspected according to the sampling scheme from the quality department. Furthermore, a part counter is also included to quantify the total number of processed parts. All these measurements are assumed to be of an acceptable low cost and are acquired through non-invasive procedures during the manufacturing process (online). The proposed model-based system develops a generic model to express the evolution of power consumption during the whole tool life, and a generic model that relates power consumption with surface roughness. The latter model allows estimations of the surface roughness in the periods where no roughness measurements are received.

The generic models are based on polynomial approximations which are versatile enough to be used in any machining process, such as milling or turning, and they require a low number of parameters which can be updated to adapt the system to any cutting condition change and tool-to-tool stochastic variations. The properties of these generic models were selected from several variants after being validated using datasets that were developed considering different machining

models available in the literature in relation with surface roughness, power consumption and tool wear; thus, the generalization capability of the chosen general models is ensured. The updating process is performed using an *Adaptive Recursive Least Squares* (ARLS) algorithm, which updates the coefficients of the polynomial function using the measurements received by the monitoring system. In order to validate the proposed approach, the performance of the approaches are compared with common tool change strategies in terms of number of consumed tools, number of processed parts out of specifications and total consumed energy.

As a summary, this chapter uses the aforementioned measurements, obtained online using non-invasive procedures, to develop a tool replacement algorithm. To obtain a general model that relates the surface roughness with the power consumption and a model that relates the power consumption with the number of processed parts, we first develop a dataset using theoretical models, which is used to test several proposed base models. The most fitting models, which are based on polynomial approximations, are used in the ARLS algorithm, where the parameters of the polynomials are updated depending on the received measurements. Finally, we use this algorithm to define the tool replacement procedure, and we test it against other tool replacement procedures in a simulated case study.

This chapter is organized as follows: Section 7.2 states the problem and the direct strategies, Section 7.3 presents the proposed model-based approaches, and Section 7.4 develops the data that will be used to validate the proposed approaches. In Section 7.5, a simulation using the data is used to select the most fitting models. Section 7.6 explains the ARLS updating algorithm, Section 7.7 develops a benchmark using the previous data, and validates the performance of the updating algorithm. After that, the performance of all approaches is evaluated using a simulated experiment, whose results are subsequently discussed. Finally, Section 7.8 concludes the chapter.

## 7.2 Problem statement

Useful tool life is defined as the total cutting time that a tool takes to attain certain conditions; in practical workshop situations, useful tool life ends when the tool processes parts out of specifications. However, the principal procedure to determine the end of the useful tool life is measuring the tool flank wear of the tool [32]. Nevertheless, tool flank wear cannot be directly measured without interrupting the machining process. Despite this, there are several available measurements during the machining process that can be taken without halting it, and depend on the tool flank wear evolution. These measurements can be used to monitor indirectly the state of the tool. However, the functions that relate this dependency are also affected by other machining conditions.

In this section, we shortly review the different phenomena that are affected by tool flank wear in order to deduce indirect measurements that can be useful for our propose. After that, we will analyze the availability and properties of the measurements, and we will present some simple straigthforward strategies that will be use later to validate our proposed strategies.

### 7.2.1 Cutting tool wear phenomena

The evolution of tool flank wear with cutting time can be separated in three stages: the *initial wear* stage, where the tool flank wear grows exponentially with time; the *steady wear* stage, where tool wear increases mostly linearly; and the *severe wear* stage, where tool flank wear grows exponentially again.

In this work, we assume that the time required to process a part consumes a given fraction of the total useful tool life, therefore, we will indistinctly use the usage time and the number of processed parts.

As tool flank wear implies the deformation of the cutting tool, the required forces to carry the machining process increase accordingly [36], leading to an increment of the average power consumption in each part. Therefore, average power consumption can be used to monitor tool life, as a certain correlation between tool flank wear and power consumption can be observed. We can express this idea as

$$P_c(k) \propto F_c(k) \propto V_b(k) \propto k, \tag{7.1}$$

where $k$ is the number of parts currently machined since the last tool change, $P_c$ is the characteristic power consumption for the present part (i.e. the average power consumption during the cutting process), $F_c$ the cutting forces and $V_b$ the characteristic tool flank wear for that part.

That cutting tool deterioration also affects the surface roughness of the processed parts. According to [37], the values of the surface roughness ($R_a$) also increase when the tool flank wear increases, and we can express this idea as

$$R_a(k) \propto V_b(k) \propto k. \tag{7.2}$$

As the tool flank wear has a given evolution along time, and under the assumption of low rate of machining time in each part w.r.t total tool life, we can state that each processed part has a characteristic surface roughness related to the characteristic tool flank wear during the machining of that part, i.e., the surface roughness also evolves with each processed part.

The evolution of $P_c$ and $R_a$ throughout the whole cutting tool life depends on the cutting conditions given by the cutting speed, the feed and the cutting depth. These cutting conditions are usually constant for a given manufacturing purpose, but may be adapted along time if the product requirements or materials change.

### 7.2.2 Available measurements

In this chapter, we try to obtain techniques with the aim of industrial applicability, so we assume that the measurements that can be obtained present an acceptable low cost, and can be acquired through a non-invasive procedure during the manufacturing process (i.e. they are acquired online).

The types of measurements in this chapter are indicated by the expression **MX**, where X is the assigned number of a given type of measurement. The proposed setup has the following available measurements (Figure 7.1):

**M1. Power consumption.** The power consumption measured within each processed part. This value is the average of the consumed power during each machining iteration, i.e., during all the loaded period. This measurement is taken using an a current clam (given a constant input voltage) or a power meter, which applies high uncertainty to the measured value. These measurements are available at all times during the process. As the machining movements and power consumption may present a pulsating nature, this value is not monotonic, and we assume that we can obtain some characteristic value for the power consumption for each processed part. One way to obtain this value is to compute an average value during a given time window of the processing in each part.

**M2. Surface roughness.** An indicator of the processed part quality, the surface roughness of the processed parts is analyzed with a profilometer at a given localization in each selected part, which applies low uncertainty to the measured value. Due to the slow sampling process, we assume that we can only select one from a given number of processed parts, as this measurement can affect the production rate. Thus, these measurements are scarcely available w.r.t. the number of processed parts.

**M3. Number of processed parts.** We assume that we have available a counter of the number of parts being processed until the present time. Each time a tool is replaced, that counter is reset, so we measure directly the usage of the tool. This measurement is proportional to the usage time of the tool.

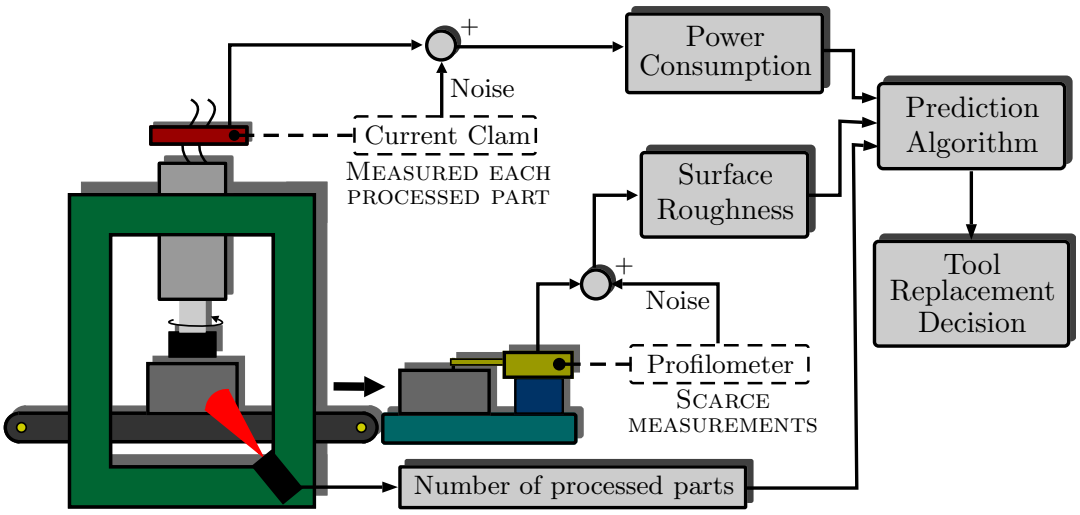



**Figure 7.1.** Problem case description.

We consider that these measurements can be easily acquired and that, in most real applications, they are monitored.

### 7.2.3 Straightforward strategies

Our aim is to develop algorithms that optimize the tool utilization in the proposed sensor-constrained setup. As both power consumption and surface roughness are indirect indicators

of the tool wear, which is itself an indicator of the remaining useful time of a tool, monitoring these variables using mostly-raw data with a simple algorithm should lead to an acceptable tool usage.

We first present straightforward strategies, expressed as different approaches, that are based on the direct comparison of the available measurements with some given thresholds. The first one is a quasi-optimal approach, but not directly viable in industry. The second and third one are quite direct, while the last one is a more complex strategy to take profit from the scarcely measured roughness.

**A1. Persistent measurement of the roughness.** In this approach, the surface roughness of the processed parts is always measured. When the measured roughness value surpasses a certain limit, the tool is replaced and the corresponding processed part is rejected. This approach is not effective in production as the procedure to acquire the roughness measurement needs a no negligible time w.r.t. to machining time, and the production would be delayed. We present this approach for comparison purposes as, in this case, the usage of the tool would be quasi-optimal in the sense of underusage and overusage (except for the last rejected processed part).

**A2. Fixed number of processed parts approach.** This simple approach consists of changing the tool once it has processed a predetermined number of parts. This fixed threshold for the part counter should be set manually. Nevertheless, the initial value must be estimated using previous experimentation. This approach is quite straightforward as it lacks of an updating mechanism. It is only valid if the machining and material conditions are quite stable, yet it can be useful if the quality requirements are not strict.

**A3. Power-limited change approach.** This approach consists of changing the tool when the power consumption reaches a certain threshold. As the available signal of the power consumption may carry measurement noise, we must use a low-pass filter. This approach requires previous experimentation in order to calculate the power consumption threshold. These experiments consist of completely using several tools, constantly measuring the surface roughness of all the processed parts, in order to determine the power consumption range in which the parts' surface properties begin to fail the demanded specifications. This approach is useful if the machining and material conditions are stable. Its initialization leads to a tool change policy that results more precise than the **A2** approach, due to the needed previous experimentation, but it is more expensive. This approach also lacks from an adaptation procedure.

**A4. Roughness interpolation approach.** This approach uses the scarce measurements of the surface roughness to estimate the remaining useful life of the tool when it is working near to roughness specifications. The algorithm is first initialized by using a single tool, measuring scarcely the roughness, and storing the first measured value which has exceeded the desired surface roughness threshold, as well as the immediate measured previous one, including the current number of processed parts. An interpolation between those values gives us an approximated value of the useful life of the tool. When we use a new tool

and the number of processed parts is close (but below) to the estimated tool life, we extrapolate linearly the last two roughness measurements to estimate the processed part number in which the roughness limit will be surpassed and, then, we replace the tool at that time. If the roughness measurement at any part exceeds the threshold, the tool is immediately replaced and the approximated useful life is updated. This technique trusts on a linear degradation of the roughness through processed parts for the last period of the useful life. Due to this approximation, this technique is not optimal, but has a simple implementation.

This approach has some update thanks to roughness measurements (when a measurement detects that we are out of bounds, i.e., when the tool life is reduced from what was expected), but it does not check the validity of the predictions. If the change on the machining conditions or material properties (tool or part) is such that the useful tool life is extended, we will not notice it and we may be changing the tool earlier than in an optimal procedure.

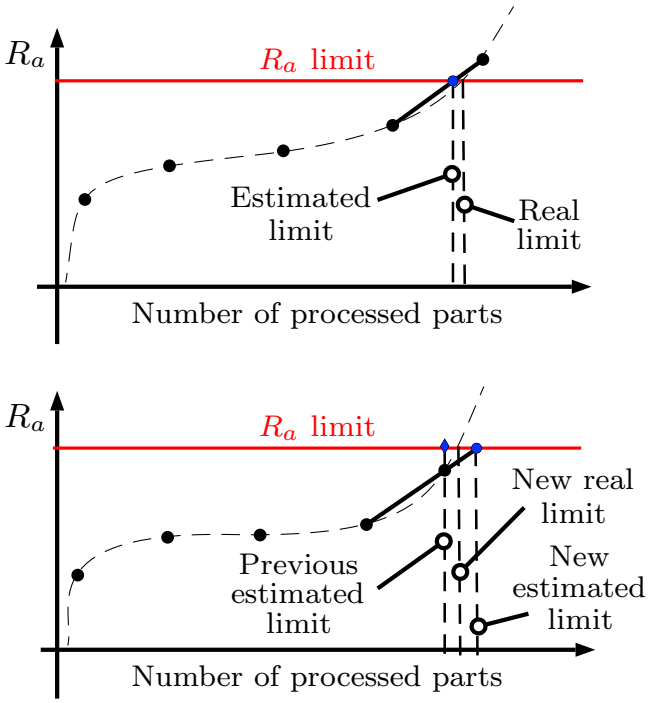



**Figure 7.2.** Interpolation-based approach. Its precision depends on the measurement frequency.

The three last approaches can be effective in very repetitive conditions and their implementations are simple. However, as reality is far from repetitive, these approaches will not work optimally as they lack in flexibility upon changes on the machining conditions or tool and part materials.

## 7.3 Proposed model-based approaches

Taking into account the premises of the previous approaches, we search now for an algorithm that is flexible enough to detect any changes in the whole cutting process behavior, while avoiding the need of performing several experiments any time the machining or the material conditions do change. Improving the previous approaches requires using the different measurements indicated in Section 7.2.2, which are assumed to be of an acceptable low cost and acquired online through non-invasive procedures, and the links between the behavior of the power consumption and the surface roughness of the processed parts (as both are affected by the tool flank wear shown in Section 7.2). We wonder if the fusion of the available data can lead us to predict the remaining useful life of the tool or to detect when the roughness thresholds are surpassed, and, thus, can lead us to optimize the usage of the machining tools. We also wonder if we can use any measurements (power and surface roughness) to update and improve those predictions when materials or machining conditions do change.

The strategies from this section are expressed as two different approaches depending on the final tool replacement decision. These approaches are based on the use of two models: one that relates the power consumption as a function of the number of processed parts, i.e., $P_c(k) = f(k)$, and a second one that relates the roughness with that power consumption, i.e., $R_a = f(P_c)$. In the following sections we will detail how to obtain, identify and update those models in real-time. Once we have a model and an updating method, we propose the following two approaches for optimal tool change:

**A5. Tool lifetime prediction.** When a tool has finished its useful life, the algorithm uses the gathered data from that tool to predict the behavior of both the power consumption and the surface roughness signals when using a new tool. With that, the approach estimates the maximum number of parts a new tool will be able to produce before surpassing the surface roughness' limits. When that number of parts is processed, the tool is changed again and the procedure is repeated.

**A6. Next-step prediction.** In this approach, the algorithm is constantly predicting the surface roughness value of the next part. If the prediction indicates that the surface roughness limit will be surpassed, the tool is changed. The models used for predictions are constantly updated with each new power or roughness measurement.

These approaches may lead to better tool replacement than the previous strategies. However, in order to achieve a general algorithm, we must first obtain general models that can be used for any application. Our aim is that those models do not depend on cutting conditions or prior knowledge of materials and flank wear phenomena.

In the following sections we develop these model-based approaches:

- In Section 7.4 we first obtain a set of data to explore possible models able to represent several conditions.

- In Section 7.5 we obtain general models that can fit the previous dataset.
- In Section 7.6 we present the updating procedure that allows us to perform any initial experimentation and update the models to adapt the to changing cutting and material conditions.

## 7.4 Dataset generation for model search

Several authors have studied the evolution of tool flank wear for several cutting conditions or materials in both tools and machined parts. Also, one can find studies about the influence of the cutting conditions and tool flank wear in the forces during machining, as well as the corresponding power consumption. Furthermore, there are different studies that explore the influence of cutting conditions and tool flank wear on surface roughness. In this section we explore the results of different authors to generate a set of data including tool flank wear, power consumption and surface roughness that allows us to search for general models in the aim of predicting the remaining useful tool life.

### 7.4.1 Tool flank wear dataset generation

Previous research on the evolution of tool flank wear has been compiled in [166, 167]. Studies like [35] state that the tool flank wear $V_b$ has an evolution on time $t$ given by

$$V_b(t) = A\,\log(B\,t+1) + C\,t^3, \tag{7.3}$$

with $A$, $B$ and $C$ some given model parameters that depend on the cutting and material conditions. This evolution includes the incipient initial wear, steady wear and final severe wear. Other research works as [166,168,169] and references therein state that the evolution may follow a differential equation that depends on the temperature $T$:

$$\frac{d\,V_b(t)}{dt} = A + B\,e^{\frac{-C}{T(t)}}, \tag{7.4}$$

with $A$, $B$ and $C$ some given constants that depend on the cutting and material conditions. This equation focuses on the initial wear and the steady wear, but includes the effects of the variations of the temperature w.r.t time.

Simulated data for several tool flank wear evolutions have been generated using previous equations, employing parameters from Table 7.1. The evolutions are shown in Figure 7.3a.

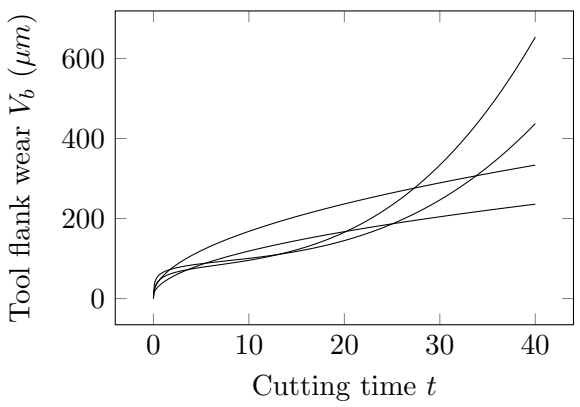
### 7.4.2 Power consumption dataset generation

In order to monitor the tool condition, the authors in [170] related the cutting power with the tool flank wear with a linear relationship function

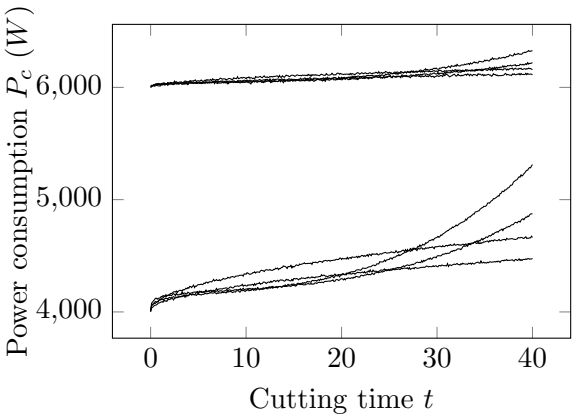
$$P_c = \alpha + \beta\ V_b, \tag{7.5}$$

**Table 7.1.** Tool flank wear ($V_b$) dataset generation parameters.

| # | Eq. | $A$ | $B$ | $C$ | $-C/T$ |
|---|---|---|---|---|---|
| Dataset V1 | (7.3) | 13.06 | 149.5 | 0.00506 | - |
| Dataset V2 | (7.3) | 14.891 | 34.7 | 0.008526 | - |
| Dataset V3 | (7.4) | 0 | 0.0375 | - | 10.39 |
| Dataset V4 | (7.4) | 0 | 0.0750 | - | 10.39 |




**(a)** Tool flank wear ($V_b$) dataset.




**(b)** Power consumption ($P_c$) dataset.

**Figure 7.3.** Tool flank wear and Power consumption vs. Cutting time.

where parameters $\alpha$ and $\beta$ are empirical constants. These empirical constants were developed as empirical functions depending on cutting conditions in [171]. Further research [163] related the power consumption with the resulting cutting forces in the machining process and the current state of the tool flank wear using physics-based functions that depended on machining settings. This research concluded that, under constant settings, power consumption was related with the tool flank wear with a linear function, thus relating parameters $\alpha$ and $\beta$ with real machining settings.

To generate the power consumption dataset, we applied several variants of the linear equation (7.5) to the previously shown tool flank wear datasets, using parameters from Table 7.2. These equation systems are shown as follows:

$$\begin{cases} P_c = \alpha + \beta\ V_b, \\ V_b(t) = A\ \log(B\,t + 1) + C\,t^3. \end{cases} \qquad \begin{cases} P_c = \alpha + \beta\ V_b, \\ V_b(t) = \int_{\tau=t_0}^{\tau=t} A + B\,e^{\frac{-C}{T(\tau)}} d\tau. \end{cases}$$

Their evolution is shown in Figure 7.3b. In order to model uncertainty related to the measurement of characteristic power consumption, zero-mean Gaussian noise has been added.

### 7.4.3 Surface roughness dataset generation

Surface roughness of the processed parts has frequently been related w.r.t. cutting time in the literature [172, 173], along with several cutting conditions, using empirical equations. The

**Table 7.2.** Power consumption ($P_c$) dataset generation parameters.

| # | Eq. | $\alpha$ | $\beta$ |
|---|---|---|---|
| Dataset P1 | (7.5) | 6000 | 0.5 |
| Dataset P2 | (7.5) | 4000 | 2 |

**Table 7.3.** Surface roughness ($R_a$) dataset generation parameters.

| # | Eq. | $\delta$ | $\epsilon$ | $\gamma$ |
|---|---|---|---|---|
| Dataset R1 | (7.6) | 0.1 | 5.5 | 0.455 |
| Dataset R2 | (7.6) | 0.1 | 5 | 0.7 |
| Dataset R3 | (7.6) | 0.1 | 6.5 | 0.8 |
| Dataset R4 | (7.6) | 0.1 | 4.5 | 0.6 |

relationship between surface roughness and tool flank wear has been researched, though. In [37], the authors proposed an empirical equation that expressed the values of the surface roughness as a function of tool flank wear and several other cutting conditions. Considering constant conditions, the function presents the form

$$R_a = \delta + \epsilon \, V_b^{\gamma}. \tag{7.6}$$

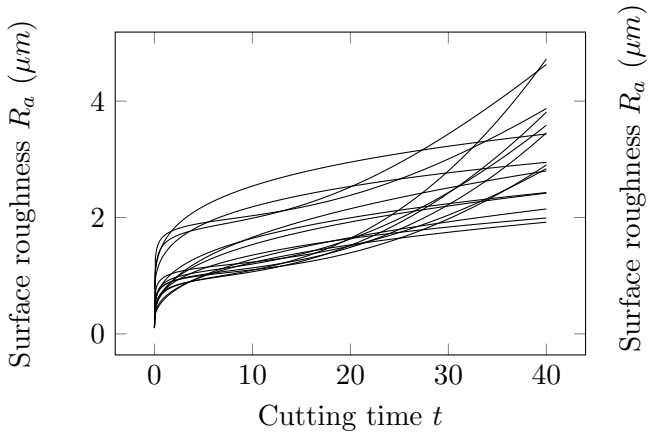
The surface roughness dataset has been generated using the previously shown tool flank wear dataset and equation (7.6), employing parameters from Table 7.3. These equation systems are shown as follows:

$$\begin{cases} R_a = \delta + \epsilon \, V_b^{\gamma}, \\ V_b(t) = A \, \log(B\, t + 1) + C\, t^3. \end{cases} \qquad \begin{cases} R_a = \delta + \epsilon \, V_b^{\gamma}, \\ V_b(t) = \int_{\tau=t_0}^{\tau=t} A + B\, e^{\frac{-C}{T(\tau)}} d\tau. \end{cases}$$

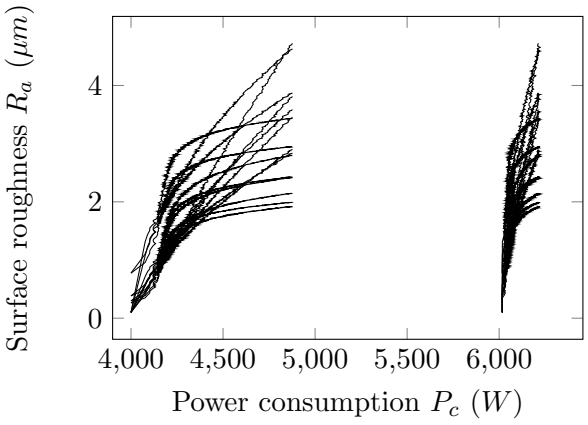
Their evolution is shown in Figure 7.4a. Additionally, in Figure 7.4b it is shown the relationship between the surface roughness dataset w.r.t the power consumption dataset.

## 7.5 General models for power consumption and surface roughness

The next step is finding a starting point (base) model that can express the behavior of the proposed online non-invasive measurements, the power consumption and surface roughness, during the complete tool use. This model comprises a trade-off between generalization (as it must be able to encompass all the possible variations from the dataset), adaptability to changes (so it can be updated with the available data) and the need of simple calculations and data storage on the startup. We have chosen a polynomial model, that is linear on its parameters (thus can be easily updated) and with the minimum number of parameters to optimize data storage and initialization speed.




**(a)** Surface roughness ($R_a$) dataset.




**(b)** Surface roughness ($R_a$) dataset w.r.t Power consumption ($P_c$) dataset.

**Figure 7.4.** Surface roughness vs. Cutting time and vs. Power consumption.

In order to express the different evaluated base models, we need first to express some generator functions. Polynomial models can be expressed as linear regression models in the form

$$z(y) = \phi(x)\,\theta + v, \tag{7.7}$$

where $z(y)$ is a function of the observed value $y$, $\phi$ is the regression vector with independent variables $x$, and $\theta$ is the parameter vector. $v$ is a random term with the non-explained behavior of measurements $z(y)$ due to, for instance, uncertainty on the model or measurement noise.

We will express polynomials through

$$p_n(x) = c_0 + c_1\,x + c_2\,x^2 + \cdots + c_n\,x^n = \underbrace{\begin{bmatrix} 1 & x & x^2 & \cdots & x^n \end{bmatrix}}_{\phi_n(x)} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}}_{\theta}, \tag{7.8}$$

and we will use notation $\phi_n(x) = \begin{bmatrix} 1 & x & x^2 & \cdots & x^n \end{bmatrix}$ to express the generation of the regression vector for that polynomial. We are also interested on the search for both additive and multiplicative functions, so the observed value may be expressed directly or logarithmically, i.e.

$$z(y) = y,\ z(y) = \log(y),$$

as well as the independent variables, that may be expressed directly or in logarithm form. Once we have a model, the estimation of the observed variable can be performed by $\hat{y} = \phi(x)\,\theta$ or $\hat{y} = \exp(\phi(x)\,\theta)$ depending on the selected observation function. Therefore, for each polynomial model, there are two parameters that must be selected: the logarithmic mode and the polynomial degree.

Using the generated dataset from the previous section, we have obtained the fittest parameter vectors $\theta$ for both desired models using the Least Squares (LS) method. The obtaining of parameter vector $\theta$ has been carried out for each logarithmic mode and up to the fifth degree.

The performance of each mode and degree has been validated using the variance of the estimation error for the latter half of the tool life, i.e., the variance of the difference between each dataset and the corresponding predicted values of the model,

$$\sigma^2 = var\left(z(y) - \phi_n(x)\,\theta\right).$$

Each model has been developed using the fittest parameter vector $\theta$ for the corresponding mode and degree. Thus, each definitive base model will be selected as a trade-off between a low number of parameters and a low estimation error variance ($\sigma^2$).

The model for the growth of the power consumption is a function of the number of processed parts. The logarithmic modes that will be compared in both the measurement part and deterministic part of the model are

$$\begin{aligned} P_c(k) = p_n(k) + v^{P_c}(k);\ P_c(k) = p_n(log(k)) + v^{P_c}(k); \\ \log(P_c(k)) = p_n(k) + v^{P_c}(k);\ \log(P_c(k)) = p_n(log(k)) + v^{P_c}(k); \end{aligned} \tag{7.9}$$

where $n$ is the degree of the polynomial, and $v^{P_c}$ includes both the measurement noise and the unmodeled behavior. The comparison of the performance of all the proposed models is shown in Figure 7.5. Same equations in the dataset appear as part of the same line.

Options 7.5b and 7.5d are discarded due to the general high estimation error variance. 7.5a and 7.5c present similar results. In both cases, from the third degree and beyond, the estimation error variance is not substantially reduced; thus, a third degree polynomial is selected. Option 7.5a is chosen before 7.5c because it requires less computational costs, i.e.,

$$P_c(k) = c_0 + c_1\,k + c_2\,k^2 + c_3\,k^3 + v^{P_c}(k) = p_3(k) + v^{P_c}(k) = \phi_3(k)\theta^{P_c} + v^{P_c}(k). \tag{7.10}$$

The surface roughness model is a function of the power consumption. The logarithmic modes that will be compared are

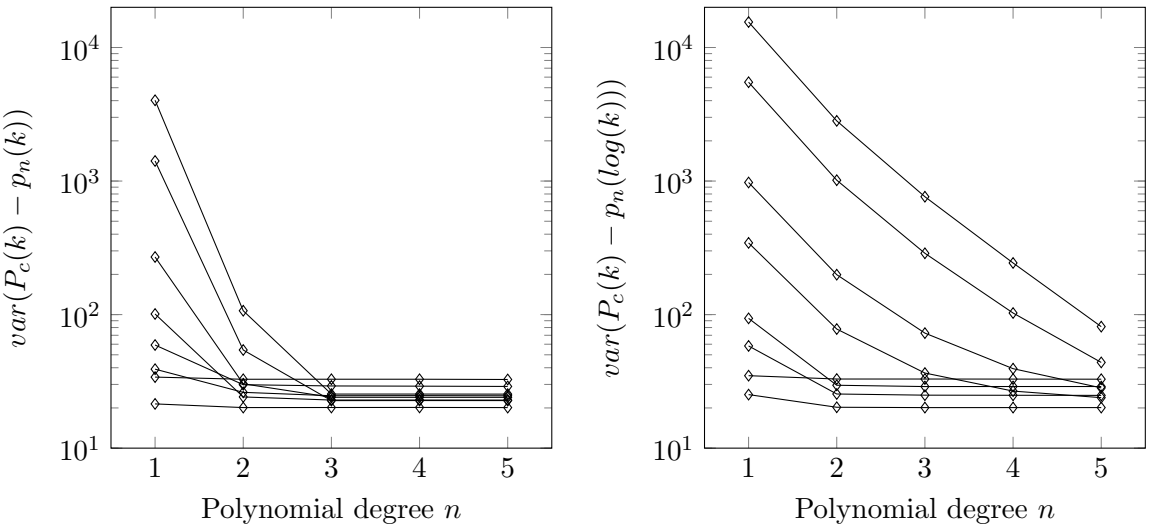
$$\begin{aligned} R_a(k) = p_n(P_c(k)) + v^{R_a}(k);\ R_a(k) = p_n(\log(P_c(k))) + v^{R_a}(k); \\ \log(R_a(k)) = p_n(P_c(k)) + v^{R_a}(k);\ \log(R_a(k)) = p_n(log(P_c(k))) + v^{R_a}(k); \end{aligned} \tag{7.11}$$

where $n$ is the degree of the polynomial, and $v^{R_a}$ includes both the measurement noise and the unmodeled behavior. The comparison of the performance of all the proposed models is shown in Figure 7.6. Same equations in the dataset appear as part of the same line.

Options 7.6c and 7.6d are discarded due to the general high estimation error variance. 7.6a and 7.6b present similar results. In both cases, from the second degree and beyond, the estimation error variance is not substantially reduced; thus, a second degree polynomial is selected. Option 7.6a is chosen before 7.6b because it requires less computational costs, i.e.,

$$R_a(P_c(k)) = d_0 + d_1\,P_c(k) + d_2\,P_c(k)^2 + v^{R_a}(k) = p_2\left(P_c(k)\right) + v^{R_a}(k) = \phi_2\left(P_c(k)\right)\theta^{R_a} + v^{R_a}(k). \tag{7.12}$$

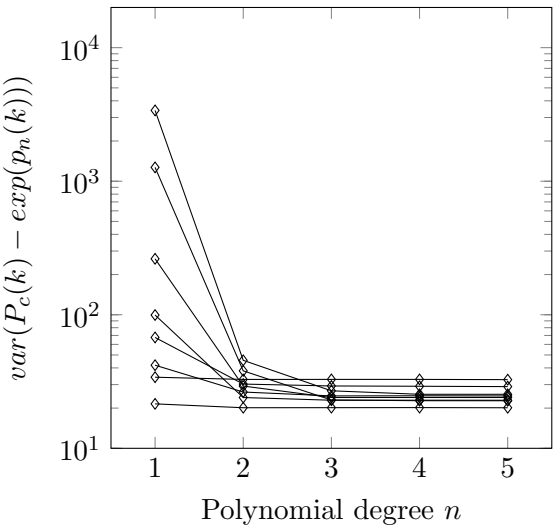
Parameters from $\theta$ have been labeled here as $d_0$, $d_1$,... to differentiate them from their equivalents in the $P_c$ function.



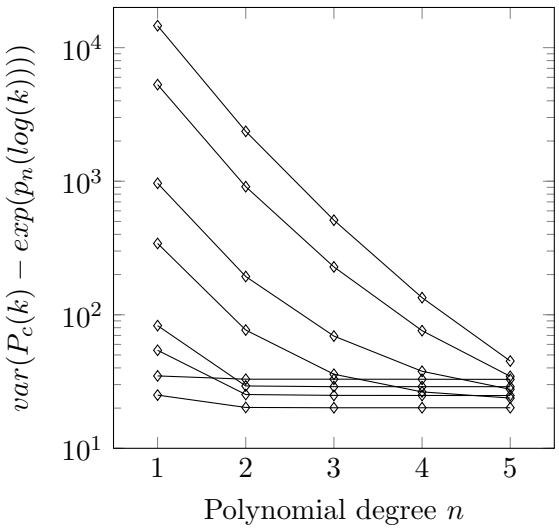

**(a)** Variance of $P_c(k) - p_n(k)$




**(b)** Variance of $P_c(k) - p_n(log(k))$

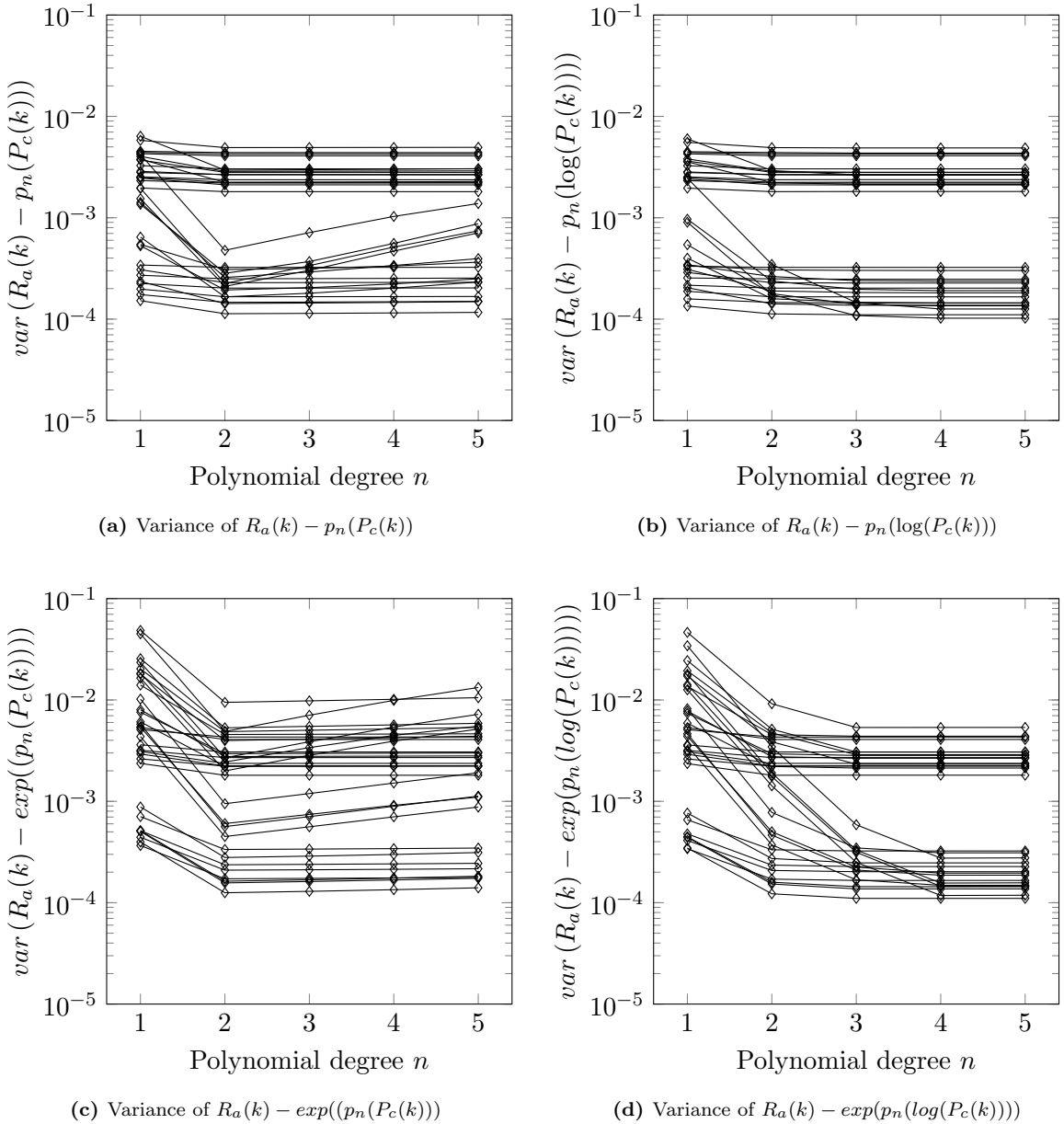



**(c)** Variance of $P_c(k) - exp(p_n(k))$




**(d)** Variance of $P_c(k) - exp(p_n(log(k)))$

**Figure 7.5.** Validation of the $P_c(k)$ models.




**(a)** Variance of $R_a(k) - p_n(P_c(k))$

**(b)** Variance of $R_a(k) - p_n(\log(P_c(k)))$

**(c)** Variance of $R_a(k) - exp((p_n(P_c(k)))$

**(d)** Variance of $R_a(k) - exp(p_n(log(P_c(k))))$

**Figure 7.6.** Validation of the $R_a(k)$ models.

## 7.6 Real-time model update using Adaptive Recursive Least Squares algorithms

Once the model is chosen for each independent variable, we must use an algorithm that allows us to obtain the model parameters that best fit the actual behavior. This is necessary to update the model when there is a change in the materials of the tool and part or in the machining conditions. In this section we first state the algorithm to obtain the model for power consumption prediction as a function of the processed parts within the used tool and how to estimate the power consumption in future processed parts, i.e., $\hat{P}_c = f(k)$. Then, with the use of those power predictions, we state the algorithm to obtain the model for resulting surface roughness as a function of the consumed power, i.e., $\hat{R}_a = f(\hat{P}_c)$. As the chosen models are additive, the starting point is a general model $y = \phi(x)\,\theta$. For each model, we define the measured output $y$, the independent variable $x$ and the regression vector function generator $\phi$.

Let us first introduce some definitions for part counting. We denote with $k$ the number of part being processed within a given tool (a value that is reset with each new tool replacement). We will use $i$ as global counter of the number of processed parts. We will denote with $k(i)$ the function that gives the number of processed part $k$ within the actual tool from the knowledge of the number of total processed parts $i$ (i.e., $k = k(i)$ is a sawtooth-like function that resets when we change the tool). Furthermore, $i$ indicates the progress of the global time. We also define $j$ as a counter of the number of parts in which the surface roughness is measured. This measurement is scarcely acquired for its cost and time consumption. We denote as $i_j$ the number of processed part $i$ in which roughness measurement number $j$ has been performed.

### 7.6.1 Power consumption predictions

For power predictions we chose a polynomial additive model in which the variables from the general model $y = \phi(x)\,\theta$ become

$$y = P_c, \quad x = k, \quad \phi(x) = \phi_3(k).$$

Therefore, we write the generator function for power consumption as

$$\underbrace{y}_{P_c} = \underbrace{\begin{bmatrix} 1 & k & k^2 & k^3 \end{bmatrix}}_{\phi_3(k)} \underbrace{\begin{bmatrix} c_0 \\ \vdots \\ c_3 \end{bmatrix}}_{\theta^{P_c}} + v^{P_c}, \tag{7.13}$$

with $\theta^{P_c}$ the parameter vector to be obtained and updated in real-time. We first apply an initialization step that consists of applying Least Squares when we have acquired more samples than number of parameters ($n = 4$) for the model representing the power consumption on time. For instance, we can acquire the data for the complete life of the first tool. Let us call $N$ the number of acquired data for initialization, being $N > n$. We obtain the initial values for the parameter vector as

$$\hat{\theta}_N^{P_c} = (X^\intercal\, X)^{-1} X^\intercal\, Y, \tag{7.14}$$

where

$$X = \begin{bmatrix} \phi_3(k(1)) \\ \phi_3(k(2)) \\ \vdots \\ \phi_3(k(N)) \end{bmatrix}, \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} P_c(1) \\ P_c(2) \\ \vdots \\ P_c(N) \end{bmatrix}, \tag{7.15}$$

being $\phi(k(i)) = \begin{bmatrix} 1 & k(i) & k(i)^2 & k(i)^3 \end{bmatrix}$. The product $X^\intercal X$ will be invertible because the regressor matrix $X$ will always have a full column rank. This is due to the polynomial dependency between the columns of the matrix, the fact that $k$ will continuously increase during the initialization and that the number of required samples must be greater than the number of parameters (i.e., the degree of the polynomial), as stated above.

Furthermore, we obtain the initial value of the covariance matrix of the parameter errors as

$$\Sigma_N^{P_c} = \mathcal{P}_N^{P_c} V^{P_c}, \tag{7.16}$$

with

$$\mathcal{P}_N^{P_c} = \left(X^\intercal X\right)^{-1}, \tag{7.17}$$

representing the inverse of the information matrix, and being $V^{P_c}$ the variance of power consumption error w.r.t. regression model. This value must contain the effect all the non explained behaviors including the measurement noise as the main source of uncertainty, as well as the lack of fit with the used polynomial model. We can also use the following value if we don't know the measurements' variance noise

$$V^{P_c} = \frac{1}{N-n}(Y - X\,\hat{\theta}_N^{P_c})^\intercal (Y - X\,\hat{\theta}_N^{P_c}).$$

Once the values of the model parameters have been initialized, we update in real-time their values with each new measurement in parts $i > N$ using the following algorithm.

### 7.6.2 Proposed algorithm for adaptive power consumption predictions

The following equations are used to perform a recursive least squares algorithm with adaptive forgetting factor (i.e., the ARLS algorithm) with the aim to predict the power consumption during the machining process.

Before the explanation of the proposed algorithm for power consumption predictions, we define the a priori estimation of the power consumption. This is expressed as

$$\hat{P}_c(i|i-1) = \phi_3(k(i))\hat{\theta}_{i-1}^{P_c}, \tag{7.18}$$

where $\hat{\theta}_{i-1}^{P_c}$ contains the parameters that were estimated in the previous iteration, and $\phi_3(k(i))$ is the regression vector, which uses the values of the present iteration $k(i)$. As a remainder, $i$ is the total number of processed parts during the experiment, and $k = k(i)$ is the number of parts processed by the current cutting tool (function $k = k(i)$ would present a sawtooth-like form). Thus, the regression vector is arranged as follows:

$$\phi_3(k(i)) = \begin{bmatrix} 1 & k(i) & k(i)^2 & k(i)^3 \end{bmatrix}, \tag{7.19}$$

whose general structure was defined in equation (7.8), its degree and logarithmic mode were selected after testing in Section 7.5, equation (7.10), and appeared within the generator function for $P_c$ in equation (7.13).

The first step of any iteration in the ARLS algorithm is the calculation of the a priori error $e_i$. This is achieved using the a priori estimation $\hat{P}_c(i|i-1)$ and the direct measurements of the consumed power during the machining process of the current part, expressed as $P_c(i)$:

$$e_i^{P_c} = P_c(i) - \underbrace{\phi_3(k(i))\hat{\theta}_{i-1}^{P_c}}_{\hat{P}_c(i|i-1)}. \tag{7.20a}$$

Then, we calculate a confidence interval in which the a priori error should be contained in stable conditions.

$$J_i^{P_c} = t_{\alpha_{P_c}} \sqrt{V^{P_c}(1 + \phi_3(k(i))\mathcal{P}_{i-1}^{P_c}\phi_3(k(i))^\intercal)}. \tag{7.20b}$$

Here, $J_i^{P_c}$ represents the confidence interval with the actual model and $\alpha_{P_c}$ is the distribution percentile for a t-distribution variable. To compute the confidence interval threshold $J_i^{P_c}$ we make use of noise variance $V^{P_c}$. We select a forgetting factor depending on whether the a priori error is located within the confidence interval or not. If the a priori error is inside the confidence interval, we use a high forgetting factor ($\lambda_H^{P_c}$ close to 1) but, otherwise, we use a lower value ($0 < \lambda_L^{P_c} < \lambda_H^{P_c} \leq 1$), trying to adapt the model to the new gathered data:

$$\lambda_i^{P_c} = \begin{cases} \lambda_H^{P_c}, & |e_i^{P_c}| < J_i^{P_c}, \\ \lambda_L^{P_c}, & |e_i^{P_c}| \geq J_i^{P_c}. \end{cases} \tag{7.20c}$$

We calculate the gain vector using the forgetting factor and the inverse of the information matrix.

$$L_i^{P_c} = \frac{1}{\lambda_i^{P_c} + \phi_3(k(i))\mathcal{P}_{i-1}^{P_c}\phi_3(k(i))^\intercal}\mathcal{P}_{i-1}^{P_c}\phi_3(k(i))^\intercal. \tag{7.20d}$$

$L_i^{P_c}$ is the gain vector, which depends on the forgetting factor $\lambda_i^{P_c}$. We update the parameter vector $\hat{\theta}_i^{P_c}$ using the calculated gain vector and the a priori estimation error:

$$\hat{\theta}_i^{P_c} = \hat{\theta}_{i-1}^{P_c} + L_i^{P_c}\, e_i^{P_c}. \tag{7.20e}$$

Finally, the inverse of the information matrix $\mathcal{P}^{P_c}$ is updated with the gain vector and the forgetting factor:

$$\mathcal{P}_i^{P_c} = \frac{1}{\lambda_i^{P_c}}(I - L_i^{P_c}\phi_3(k(i)))\mathcal{P}_{i-1}^{P_c}. \tag{7.20f}$$

In order to obtain a prediction of a future value for the power consumption, as well as a filtered version of the actual power consumption, we use the following expression

$$\hat{P}_c(l|i) = \phi_3(k(l))\,\hat{\theta}_i^{P_c}, \tag{7.21}$$

where $i$ represents the actual value of the part counter, and $l \geq i$ represents a future instant of time.

The tuning parameters in this algorithm are values $\alpha_{P_c} \in (0.9, 1)$, $\lambda_H^{P_c} \in (0.9, 1]$ and $\lambda_L^{P_c} \in (0, \lambda_H^{P_c}]$, which must be chosen as a trade-off between robustness against measurement noise, adaptation ability for model changes and convergence speed. With values of $\lambda^{P_c}$ near to 1, the algorithm is less affected by the sensor noise at the cost of a low adaptation in front of model changes, and, contrarily, values of $\lambda^{P_c}$ close to 0.9 make the parameter values more sensitive to sensor noise, but more flexible to adapt to changes. On the other hand, a low value of $\alpha^{P_c}$ reduces the confidence interval width, i.e. $t_{\alpha^{P_c}}$, and assigning $\lambda^{P_c}$ to $\lambda_L^{P_c}$ (the lower value) occurs more frequently, which causes big changes on $\hat{\theta}^{P_c}$. Contrarily, a high value of $\alpha_{P_c}$ (high $t_{\alpha_{P_c}}$) implies the need of big errors for the algorithm to start adaptation to changes, thus, can cause delays on detecting new behaviors, but with the benefit of more stable parameter estimations when the model does not change.

### 7.6.3 Surface roughness predictions

We use a similar strategy for obtaining the model for the roughness prediction as a function of the power consumption. In this case, in order to mitigate the effect of the sensor noise on the power measurement and other uncertainties, we use the predicted power through the available power propagation model $\hat{P}_c(i|i)$ as an input for the identification of the roughness, and we choose a polynomial additive model in which the variables from the general model $y = \phi(x)\theta$ become

$$y = R_a, \quad x = \hat{P}_c(i|i), \quad \phi(x) = \phi_2(\hat{P}_c(i|i)).$$

Therefore, we write the generation function for surface roughness as

$$\underbrace{y}_{R_a} = \underbrace{\begin{bmatrix} 1 & \hat{P}_c(i|i) & \hat{P}_c(i|i)^2 \end{bmatrix}}_{\phi_2(\hat{P}_c(i|i))} \underbrace{\begin{bmatrix} d_0 \\ d_1 \\ d_2 \end{bmatrix}}_{\theta^{R_a}} + v^{R_a}. \tag{7.22}$$

From now on, we use the compact notation $\hat{P}_c(i)$ to denote $\hat{P}_c(i|i)$. As the roughness is scarcely measured, we cannot update the model with each part $i$, and we only update it at instants $j$ when the roughness measurement is acquired. These instants are denoted as $i_j$.

We first apply an initialization step that consists in applying Least Squares when we have acquired more samples than number of parameters ($n = 3$) for the model representing the surface roughness. Let us call $N$ the number of acquired data for initialization, with $N > n$. We obtain the initial values for the parameter vector as

$$\hat{\theta}_N^{R_a} = (X^\intercal X)^{-1} X^\intercal Y, \tag{7.23}$$

where

$$X = \begin{bmatrix} \phi_2(\hat{P}_c(i_1)) \\ \phi_2(\hat{P}_c(i_2)) \\ \vdots \\ \phi_3(\hat{P}_c(i_N)) \end{bmatrix}, \qquad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} R_a(i_1) \\ R_a(i_2) \\ \vdots \\ R_a(i_N) \end{bmatrix}, \tag{7.24}$$

being $\phi_2(\hat{P}_c(i_j)) = \begin{bmatrix} 1 & \hat{P}_c(i_j) & \hat{P}_c(i_j)^2 \end{bmatrix}$. Note that $R_a(i_j)$ refers to the $j$-th measurement of the roughness, not to the $j$-th processed part. The product $X^\intercal X$ will be invertible because the regressor matrix $X$ will always have a full column rank. This is due to the fact that there is a polynomial dependency between the columns of the matrix, that we assume that $\hat{P}_c$ will be monotonically increasing and the restriction that the number of required samples must be greater than the number of parameters (i.e., the degree of the polynomial), as stated above.

Furthermore, we obtain the initial value of the covariance matrix of the parameter errors as

$$\Sigma_N^{R_a} = \mathcal{P}_N^{R_a} V^{R_a}, \tag{7.25}$$

with

$$\mathcal{P}_N^{R_a} = \left(X^\intercal X\right)^{-1}, \tag{7.26}$$

representing the inverse of the information matrix, and being $V^{R_a}$ the variance of roughness error w.r.t. regression model.

Once the values of the model parameters have been initialized, we assume that we have available an estimation of the current power consumption with the previous model (i.e., a filtered version of the power consumption), and we update the values of the parameters of the model with each new measurement in parts $j > N$ with the Adaptive Recursive Least Squares equations.

### 7.6.4 Proposed algorithm for adaptive surface roughness predictions

The following equations are used to perform a recursive least squares algorithm with adaptive forgetting factor (i.e., the ARLS algorithm) with the aim to predict the surface roughness during the machining process.

Before explaining the proposed algorithm for surface roughness predictions, we define the a priori estimation of the surface roughness. We can express it as

$$\hat{R}_a(i_j|i_{j-1}) = \phi_2(\hat{P}_c(i_j))\hat{\theta}_{j-1}^{R_a}. \tag{7.27}$$

In this case, the ARLS algorithm is activated scarcely, depending on the frequency of the surface roughness measurements; thus, $j$ is the counter of those measurements, being $i_j$ the processed part $i$ where surface roughness measurement $j$ took place. Therefore, $\hat{\theta}_{j-1}^{R_a}$ contains the parameters that were estimated in the last time this algorithm was activated, and $\phi_2(\hat{P}_c(i_j))$ is the regression vector, which uses the values of $\hat{P}_c$ that were calculated using the previous algorithm in Section 7.6.2. As a reminder, this regression vector takes this form:

$$\phi_2(\hat{P}_c(i_j)) = \begin{bmatrix} 1 & \hat{P}_c(i_j) & \hat{P}_c(i_j)^2 \end{bmatrix}, \tag{7.28}$$

whose general structure was defined in equation (7.8), its degree and logarithmic mode were selected after testing with the datasets in Section 7.5, equation (7.12) and appeared within the generator function for $R_a$ in equation (7.22).

The first step of any iteration in the ARLS algorithm is the calculation of the a priori error $e_j^{R_a}$. This is achieved using the a priori estimation $\hat{R}_a(i_j|i_{j-1})$ and the direct measurements of the surface roughness during the instant $i_j$ (i.e. at the $j$th surface roughness measurement, which takes place on the $i$th processed part), expressed as $R_a(i_j)$:

$$e_j^{R_a} = R_a(i_j) - \underbrace{\phi_2(\hat{P}_c(i_j))\hat{\theta}_{j-1}^{R_a}}_{\hat{R}_a(i_j|i_{j-1})}. \tag{7.29a}$$

We calculate a confidence interval where the a priori error should remain in stable conditions.

$$J_j^{R_a} = t_{\alpha_{R_a}} \sqrt{V^{R_a}(1 + \phi_2(\hat{P}_c(i_j))\mathcal{P}_{j-1}^{R_a}\phi_2(\hat{P}_c(i_j))^\intercal)}. \tag{7.29b}$$

Here, $J_j^{R_a}$ represents the confidence interval with the actual model and $\alpha_{R_a}$ is the distribution percentile for a t-distribution variable. To compute the confidence interval threshold $J_j^{R_a}$ we make use of noise variance $V^{R_a}$. Afterwards, we select a forgetting factor $\lambda^{R_a}$ depending on whether the a priori error is located within the confidence interval or not. If the a priori error is inside the confidence interval, we use a high forgetting factor ($\lambda_H^{R_a}$ close to 1) but, otherwise, we use a lower value ($0 < \lambda_L^{R_a} < \lambda_H^{R_a} \leq 1$), trying to adapt the model to the new gathered data:

$$\lambda_j^{R_a} = \begin{cases} \lambda_H^{R_a}, & |e_j^{R_a}| < J_j^{R_a}, \\ \lambda_L^{R_a}, & |e_j^{R_a}| \geq J_j^{R_a}. \end{cases} \tag{7.29c}$$

We calculate the gain vector $L^{R_a}$ using the selected forgetting factor and the inverse of the information matrix $\mathcal{P}^{R_a}$. The gain vector is affected by $\lambda_j^{R_a}$:

$$L_j^{R_a} = \frac{1}{\lambda_j^{R_a} + \phi_2(\hat{P}_c(i_j))\mathcal{P}_{j-1}^{R_a}\phi_2(\hat{P}_c(i_j))^\intercal.} \mathcal{P}_{i-1}^{R_a}\phi_2(\hat{P}_c(i_j))^\intercal \tag{7.29d}$$

Then, we update the parameter vector $\hat{\theta}^{R_a}$ using the calculated gain vector and the a priori estimation error:

$$\hat{\theta}_j^{R_a} = \hat{\theta}_{j-1}^{R_a} + L_j^{R_a}\, e_j^{R_a}. \tag{7.29e}$$

Finally, the inverse of the information matrix is updated with the gain vector and the forgetting factor:

$$\mathcal{P}_j^{R_a} = \frac{1}{\lambda_j^{R_a}}(I - L_j^{R_a}\phi_2(\hat{P}_c(i_j)))\mathcal{P}_{j-1}^{R_a}. \tag{7.29f}$$

We use the following expression at any instant $i$ to estimate a future value of the surface roughness at instant $l$

$$\hat{R}_a(l|i) = \phi(\hat{P}_c(l|i))\,\hat{\theta}_j^{R_a} = \phi_2\left(\phi_3(k(l))\hat{\theta}_i^{P_c}\right)\hat{\theta}_j^{R_a}, \tag{7.30}$$

where $i$ represents the instant of time for the most updated model for power predictions (i.e. at the $i$th processed part), and $\hat{\theta}_j^{R_a}$ is the value of $\hat{\theta}^{R_a}$ that was calculated in the last instant $i_j$. The values of $\lambda^{R_a}$ and $\alpha_{R_a}$ are comprised within the same intervals as the ones exposed in the algorithm used to estimate the power consumption. The effects of $J^{R_a}$, $L^{R_a}$ and $\mathcal{P}_j^{R_a}$ on this algorithm are identical to their equivalents in the power consumption estimation algorithm.

Figure 7.7 summarizes the internal steps of the algorithm.

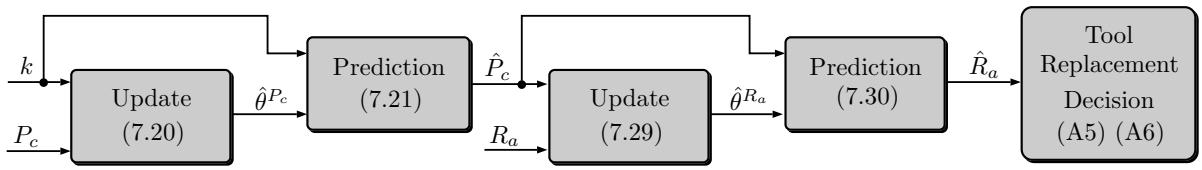



**Figure 7.7.** Operating diagram of the ARLS algorithm.

**Table 7.4.** Multiplying factors (Benchmark).

| | $Tool \leq 100$ | $Tool > 100$ | | |
|---|---|---|---|---|
| | All | **Case 1** | **Case 2** | **Case 3** |
| $V_b$ | 1.00 | 1.00 | 1.05 | 0.95 |
| $P_c$ | 1.00 | 1.00 | 1.03 | 0.97 |
| $R_a$ | 1.00 | 1.00 | 0.90 | 1.10 |

## 7.7 Simulation results

In this section, we will validate the performance of the proposed approaches from 7.3 and will compare them to the direct approaches from Section 7.2.3. Firstly, we will explain the benchmark we have used to execute the simulations. After that, we will check the internal behavior of the ARLS algorithm we have developed in Section 7.6 using the benchmark as source data. Afterwards, we will compare each approach by executing the simulations using the benchmark data, and we will evaluate their performance using several indexes. Lastly, we will discuss the results of the simulation.

### 7.7.1 Benchmark

This benchmark simulates a machining process where 500 tools are exhausted by processing 350 parts each one. It contains the evolution of the values of tool flank wear, power consumption and surface roughness resulting of that process. The general evolution of each variable is based on the dataset from Section 7.4. Tool-to-tool stochastic variations are expressed as third degree functions that are added to the tool flank wear data, acting as disturbances. The parameters of these functions are randomly generated, ensuring that the resulting tool flank wear data evolution remains increasing monotonically. Changes in cutting conditions or material properties of the raw parts are expressed as multiplier factors, which are applied to the previous functions. Shown in Table 7.4, factors for $P_c$ and $R_a$ change to opposite values to check the algorithm against the worst case scenario. Three different cases are proposed. Measurement noise has been simulated by zero-mean Gaussian noise. Its variance is calculated as $var(meas_{noise}) = (m/3)^2$, where $m$ is the uncertainty of each instrument, as shown in Table 7.5. Measurement frequency of surface roughness is also included in the aforementioned table.

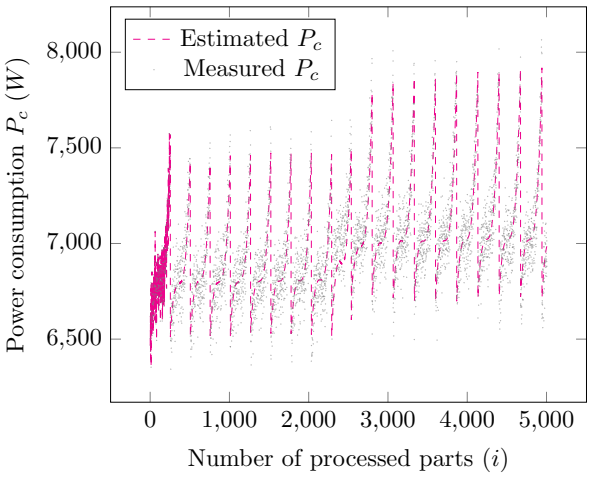
**Table 7.5.** Benchmark parameters.

| Name | Value | Units | Definition |
|---|---|---|---|
| $m^{P_c}$ | $\pm 300$ | W | $P_c$ uncertainty noise |
| $m^{R_a}$ | $\pm 0.2$ | $\mu m$ | $R_a$ uncertainty noise |
| $freq$ | 20 | $\left(\frac{parts}{meas.}\right)$ | $R_a$ measurement frequency |

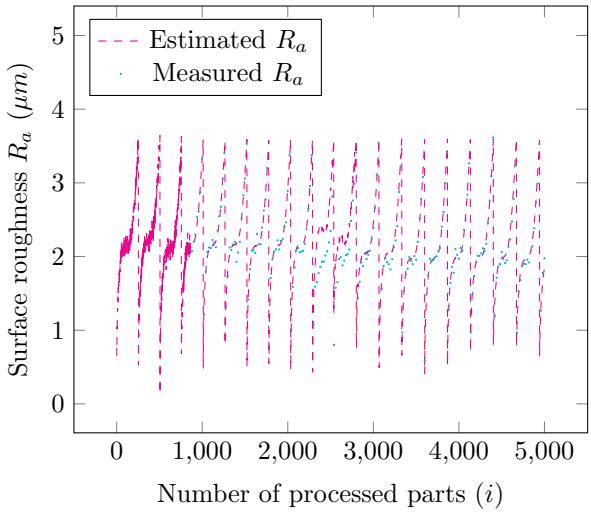
### 7.7.2 ARLS algorithm performance

The performance of the ARLS updating algorithm will be validated via several simulations. Using the benchmark (**Case 2**) as source data, the simulations consist in producing a determinate number parts within desired specifications. Depending on the selected approach (**A5** or **A6**), the tool will be replaced under different considerations. The algorithm parameters for these simulations are found in Table 7.6.

Firstly, the accuracy of the updating algorithm is checked. In this case, the selected approach does not affect the updating performance. Figure 7.8a shows the estimated value of power consumption $P_c$ and it compares it to the benchmark data of $P_c$ (the observed signal), expressed as points. Figure 7.8b shows the next-step estimated values of surface roughness $R_a$, comparing them to the benchmark $R_a$ data. Only the observed values of $R_a$ appear as points, as measurements are scarce. In both cases, the initialization of the algorithm takes place during the first tools; its length can be modified at will, but a reduced time will yield imprecise results in the first stages of the simulation.




**(a)** Estimating power consumption ($P_c$) with the ARLS algorithm.

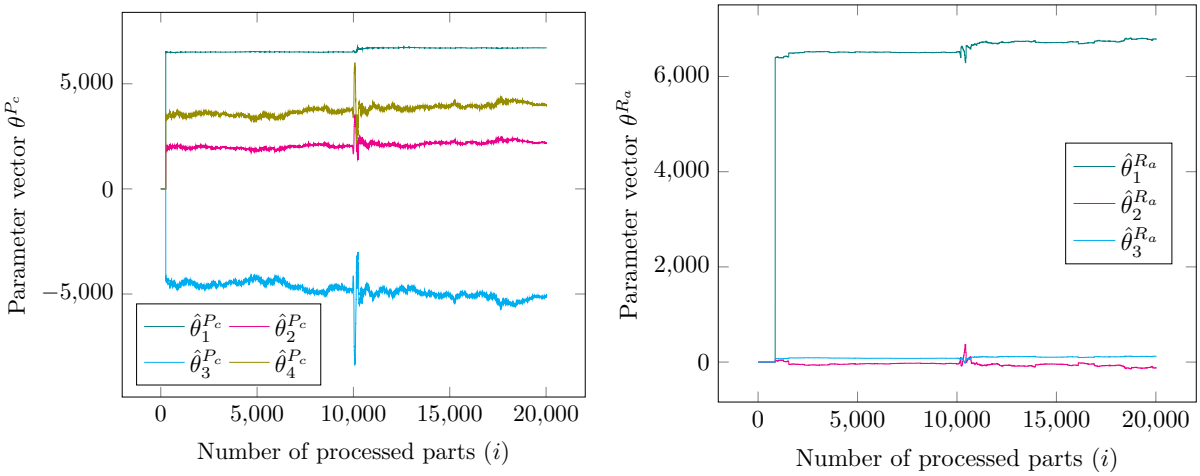



**(b)** Estimating surface roughness ($R_a$) with the ARLS algorithm.

**Figure 7.8.** Validation of the updating algorithm accuracy.

The following step is the validation of the internal stability of vector parameters $\theta$ in order to ensure appropriate predictions. Figures 7.9a and 7.9b show the evolution of $\theta$ for the $P_c$

and $R_a$ models, respectively. Parameters do not become completely stable due to tool-to-tool stochastic variations, but are rapidly adapted when cutting conditions change.




**(a)** Stability of parameter vector $\theta^{P_c}$.

**(b)** Stability of parameter vector $\theta^{R_a}$.

**Figure 7.9.** Stability of parameter vectors.

### 7.7.3 Performance indexes and settings

The performance of the approaches from Sections 7.2.3 and 7.3 is evaluated with a simulated experiment. In these simulations, which use all three cases from Section 7.7.1, each approach processes up to 50,000 acceptable parts (i.e. under specification limits). Their performance is evaluated using the following indexes: **I1. Number of consumed cutting tools**, **I2. Number of rejected parts** (which is the number of processed parts out of specifications), and **I3. Accumulated power consumption**. The latter index is proportional to the total consumed energy during the machining process, and implies a higher cost, as well as a higher ecological impact.

Simulation settings are shown in Table 7.6. Approaches **A1**, **A4**, **A5** and **A6** require the surface roughness limit $R_{a,lim}$. Approach **A2** will be simulated changing the tool at each 215 parts (**A2a**), 255 parts (**A2b**) and 295 parts (**A2c**). Approach **A3** changes the tool when a given $P_c$ limit is reached. This limit has been obtained after previous simulated experimentation. Model-based approaches **A5** and **A6** require several forgetting factors $\lambda$ and $t_\alpha$.

### 7.7.4 Discussion

The results of the simulations are shown in Table 7.7 for each benchmark case. Approach **A6** performs correctly in all cases, and is the one that behaves more similarly to the *ideal* case **A1**, in which surface roughness was constantly measured. Approach **A5** produces an excess of rejected parts, otherwise, indexes **I1** and **I3** perform similarly to the corresponding indexes of **A6**. Approach **A4** performs well in most cases, but it is outclassed by Approach **A6**.

**Table 7.6.** Simulation parameters.

| Name | Value | Units | Definition |
|---|---|---|---|
| $R_{a,lim}$ | 3.6 | $\mu m$ | $R_a$ threshold |
| $P_{c,lim}$ | 7400 | W | $P_c$ threshold (**A3**) |
| $\lambda_L^{P_c}$ | 0.7 | - | Lower $\lambda^{P_c}$ |
| $\lambda_H^{P_c}$ | 1 | - | Higher $\lambda^{P_c}$ |
| $\lambda_L^{R_a}$ | 0.8 | - | Lower $\lambda^{R_a}$ |
| $\lambda_H^{R_a}$ | 1 | - | Higher $\lambda^{R_a}$ |
| $t_{\alpha_{P_c}}$ | 3.0 $_{(\alpha_{P_c} \approx 0.998)}$ | - | Value of $t_{\alpha_{P_c}}$ |
| $t_{\alpha_{R_a}}$ | 5.0 $_{(\alpha_{R_a} > 0.999)}$ | - | Value of $t_{\alpha_{R_a}}$ |

**Table 7.7.** Simulation results.

| | **Case 1** | | | **Case 2** | | | **Case 3** | | |
|---|---|---|---|---|---|---|---|---|---|
| Approach | **I1** | **I2** | **I3** | **I1** | **I2** | **I3** | **I1** | **I2** | **I3** |
| **A1** | 196 | 195 | 345 | 191 | 190 | 351 | 200 | 199 | 339 |
| **A2a** | 233 | 0 | 340 | 233 | 0 | 346 | 233 | 0 | 333 |
| **A2b** | 197 | 102 | 344 | 197 | 41 | 349 | 201 | 1066 | 345 |
| **A2c** | 196 | 7613 | 402 | 191 | 6189 | 399 | 200 | 8903 | 404 |
| **A3** | 197 | 100 | 344 | 207 | 48 | 348 | 201 | 3091 | 360 |
| **A4** | 197 | 567 | 347 | 192 | 693 | 355 | 201 | 584 | 341 |
| **A5** | 196 | 788 | 349 | 194 | 380 | 352 | 200 | 1938 | 351 |
| **A6** | 197 | 64 | 343 | 192 | 64 | 350 | 202 | 88 | 338 |

Approach **A3** performs well in **Case 1**, which is stable, but uses a high amount of tools in **Case 2** and processes an excessive amount of parts out of specifications in **Case 3**. This is due to the fact that **A3** does not react to those internal changes. Approach **A2** behaves in a similar way. **A2b** is an *a posteriori* "optimal" choice; its performance is the most balanced from **A2** variations, but it does not react to internal changes either. **A2a** replaces the tool too early, using an excessive amount of cutting tools, while **A2c** replaces the tool too late, producing a high amount of parts out of specifications.

Note that **A5** and **A6** present the drawback that they require the use of several tuning parameters ($\lambda_H^{P_c}$, $\lambda_L^{P_c}$, $\lambda_H^{R_a}$, $\lambda_L^{R_a}$, ...). In order to locate the adequate values for these parameters, it is required to perform a simulation of the manufacturing process. Also note that these parameters are comprised within the intervals explained in Section 7.6.

## 7.8 Conclusions

In this chapter, we have analyzed several tool replacement strategies in machining processes. In all these strategies we have assumed that the measurements of the power consumed by the cutting machine and the surface roughness of the processed parts are available, although the measurements of the latter are received scarcely. We have also assumed that a processed part counter is available.

The idea behind these tool replacement strategies is to assure that the processed parts fulfill certain quality criteria based on surface roughness thresholds. The tool replacement strategies we have presented in this chapter can be classified in two types depending on its complexity: simple straightforward strategies that directly use the received measurements to decide the tool replacement moment, and model-based strategies that are able to predict the surface roughness during the periods in which no roughness measurements are available while solving the problems implied by the measurement noise.

In order to obtain suitable models for the model-based strategies, we have developed a dataset with different empirical models from the literature that have developed the evolution of tool wear and its effects on the power consumption and surface roughness increase during the cutting tool lifetime. Afterwards, we have validated the generalization capabilities of several base models with this dataset to select the fittest ones. The model-based strategies consist of an algorithm that adapts the parameters of the selected models in front to changes of the machining process. Both the selected models and their algorithms have been designed to be efficient from an implementation perspective, and require a low amount of data to be initialized. These algorithms require several setting parameters; we have included indications of how to adjust them. The tool replacement policies of the model-based strategies consist of two different variants: predicting the number of processed parts the tool will be able to process before surpassing a certain roughness threshold, calculated when a tool gets replaced; the second variant is to replace the tool if the predicted surface roughness of the following part will surpass the given threshold.

In order to compare the presented strategies, we have simulated them with a benchmark where a certain batch of parts had to be manufactured under changing machining conditions. Their performance has been evaluated using several indexes: number of processed parts out of specifications, number of consumed tools and total consumed energy. The model-based strategy that replaced the tool if the following predicted surface roughness surpassed the threshold generally presented the best results in all conditions.

Chapter 8

# Conclusions and future research

## 8.1 Conclusions

This thesis addresses several problems that may arise during the development of strategies under the Zero Defect Manufacturing framework, specifically those that are applied to multistage processes and CNC machines. We have proposed variation propagation models of two different traditional fixtures, as well as methodologies and algorithms to adjust models with process data and engineering knowledge, implement sequential fault detection and isolation, and monitor and predict the cutting tool remaining useful life.

We have focused on developing algorithms that require a relatively low amount of data, which leads to an easy implementation of the methodology, and helps to overcome other problems, such as scarce or expensive measurements. We have validated the proposed models, methodologies and algorithms using different methods. The variation propagation model, obtained using the Stream-of-Variation approach, has been numerically validated using MATLAB, and geometrically validated using AutoCAD. Additionally, it has also been validated by means of a machining experiment. The model adjustment methodology has been validated numerically using MATLAB and YALMIP, by means of a benchmark. Both the sequential fault detection and cutting tool monitoring and prognosis methodologies have been validated using MATLAB; the latter methodologies have been based on a benchmark that has been developed using empirical equations from the literature.

### Modeling and adjustment

In order to apply fault diagnosis in manufacturing processes, it is essential to obtain reliable models of the process, as well as fault propagation models between process and product. Therefore, in this thesis we have focused on developing methodologies to obtain these reliable models, expanding the SoV-based models to traditional fixtures and proposing a model adjustment methodology for linear process models.

In Chapter 3, we developed the behavior of the variation propagation in bench vices and 3-jaw self-centering chucks, which are fixtures that are used in traditional industries, as rigid body (non-deformable) SoV-based models. In the bench vice model, we included the propagation of the position and orientation errors of plain jaws, supports and pins, and the datum errors of the prism raw part. In the model of the 3-jaw chuck, we included the propagation of the position error of the jaws and the locating pin, the orientation errors of the jaws and the datum errors of the cylinder base of the raw part. Most of the elements of the model were obtained using DMVs and the 3-2-1 scheme methodology. Some interactions between the bench vice and the raw part had to be obtained using geometrical methods, as it presented non-linearities that affected the general behavior of the variation propagation. In the 3-jaw self-centering model, the effects of the position error of the jaws caused a self-centering deviation, which also had to be obtained using geometrical methods.

In Chapter 4, we proposed an adjustment methodology to reduce approximations and modeling errors in linear input-output variation propagation models of processes, which are assumed to have been obtained using physical-based methods, using data from the process and engineering knowledge. The proposed methodology includes an optimization algorithm in order to minimize the difference between the covariances of the output measurements of the product and the covariances of an estimated output obtained using the model that is being adjusted. The optimization problem is bounded by engineering knowledge and backup data, and the objective function has been convexified in order to ensure a feasible solution within a finite time.

In Chapter 5, we developed a qualitative model extracted from the process planning, which is later used to develop the sequential fault detection and isolation procedure. This model consists of a binary matrix that relates potential faulty elements of the process with its impact on certain dimensional features of the product. Zeros and ones are assigned depending on the sequence of fixtures and cutting tools used in each stage of path that defines each feature.

Finally, in Chapter 7 we proposed an adaptive recursive least squares algorithm to predict the future state of the tool flank wear in a CNC machine using indirect measurements: the power consumption and the surface roughness of the processed parts, where the latter is measured scarcely. This algorithm required general models that consisted of polynomial approximations of the behavior of these indirect measurements with respect to the amount of parts machined by the cutting tool. The degree and logarithmic mode of the polynomial approximations for each pair of variables were selected after evaluating the performance of several combinations of degrees and modes with a benchmark. This benchmark consisted of several empirical equations for the behavior of these variables with respect to tool flank wear, which were obtained from the literature.

### Monitoring, fault detection and prognosis

The search for zero defects also implies a search to minimize waste and thus, total cost. In this thesis, we have proposed monitoring, fault detection and prognosis methodologies to predict and prevent the moment when are processed with features out of specifications. To ensure an early implementation of the proposed methodologies, which would reduce wasted resources, we have

designed them in such a way that the amount of data required to initialize them is minimized. Additionally, we have dealt with instances where measurements from the process are scarce or costly, which increase the total cost of the product processing if not solved.

In Chapter 4, in the case study we presented, the proposed methodology, which adjusted a linear input-output model using data and engineering knowledge, required a low amount of parts before notably improving the results that would have been obtained using least squares.

In Chapter 5, we proposed a methodology to reduce the inspection cost and provide fast fault detection and isolation, which leads to lower waste and costs. This methodology consisted in a sequential inspection procedure using the information gain index of the inspection measurement. The information gain index is calculated using the distribution of zeros and ones from the qualitative model that was obtained from the process planning.

In Chapters 6 and 7, we have focused on the evolution of tool flank wear in CNC machines. We assume that tool flank wear cannot be directly monitored, so we use indirect variables to estimate its health and remaining useful life. These indirect variables are the power consumed by the machine, which is usually very noisy, and the surface roughness of the processed parts, which we assume that it cannot be continuously measured.

In Chapter 6, we estimated the values of the surface roughness and power consumption during the instants when no surface roughness measurements were received, assuming a constant increase; when new measurements were received, the estimated values were updated using an observer with a Steady-State Kalman Filter, using empirical equations between tool flank wear, surface roughness and consumed power.

In Chapter 7, we keep the same assumptions of scarce surface roughness and noisy power consumption measurements, and we presented a methodology to predict the remaining useful life of the cutting tool, in order to avoid wasting resources (such as underusing cutting tools or generating out-of-specification parts). Here, we assume that tool life ends when surface roughness exceeds certain threshold. The proposed methodology consisted in an Adaptive Recursive Least Squares algorithm, which was designed to deal with different measurement reception frequencies (power consumption is assumed to be accessible at any time). This algorithm is based on low degree polynomials, so it can be initialized with few measurements and, together with the adaptive term of the algorithm, quickly modify the behavior of the polynomial model if cutting conditions change.

## 8.2 Future research

There are several research lines that can be developed from the contributions of this thesis:

- On the basis of the Stream-of-Variation (SoV) methodology and the results of Chapter 3, two research lines arise. The first line consists in the development of SoV-based models for other traditionally-used fixtures, such as 4-jaw chucks. The second line consists in the development of SoV-based models of the fixtures that have been developed in this thesis, now in rotatory frames (as in lathes) and comparing the results with other approaches from the literature.

- In Chapter 4, we have proposed a methodology to adjust a physical linear propagation model using process data and engineering knowledge, which has been validated using a simple case study. A possible future research line consists in validating the results of the proposed methodology using a more realistic simulation environment of a multistage process, which has been developed by recreating physical contact relationships between components, and does not use the linear approximations that the proposed methodology assumes. This simulation may also consider that different products are being processed sharing some of the stages, thus leading to interrelated data that can be used to enhance the proposed methodology.

- Following the general research line of developing monitoring and fault diagnosis methods using reliable models, which have been sequentially obtained in Chapters 3 and 4, the next step is developing model-based observers to estimate and monitor the process variance using product measurements, in order to identify faults. We propose researching the effect of the biases caused by digital filters in online observers.

- In Chapter 5, we have developed a qualitative variation propagation model, consisting of a binary matrix, using the fixture and datum sequences from the process planning. Due to the fact that, in order to manufacture a given product there are usually several different valid process plans, the proposed future research line consists in developing a methodology to narrow down and find those plans that minimize the amount of measurements (and/or the total cost) that are required to detect faults in the process.

- The last future research line is the experimental validation of the proposed methodologies of this thesis, particularly the monitoring and prognosis algorithms from Chapters 6 and 7, and its future application in real cases in industry.

Appendix A

# Calculus of induced errors in SoV Models and implementation guide to practitioners

## A.1 Calculus of datum-induced errors in vices

Following the procedure explained in [10], we define the datum points that touch the secondary datum, denoted as $\mathbf{p_D}$ and $\mathbf{p_E}$, which depends on the relationship between the orientation errors of fixture surfaces 1 and 2 and datum surfaces A and B, and the datum point that touches the locating pin of the vice, $\mathbf{p_G}$. The nominal coordinates of these three points in FCS are denoted as $\mathbf{p_D^F}$, $\mathbf{p_E^F}$, and $\mathbf{p_G^F}$. Thus, we have

$$\mathbf{H}_R^B \cdot \mathbf{H}_F^R \cdot \widetilde{\mathbf{p}}_D^F = \widetilde{\mathbf{p}}_D^B, \tag{A.1}$$

$$\mathbf{H}_R^B \cdot \mathbf{H}_F^R \cdot \widetilde{\mathbf{p}}_E^F = \widetilde{\mathbf{p}}_E^B, \tag{A.2}$$

$$\mathbf{H}_R^C \cdot \mathbf{H}_F^R \cdot \widetilde{\mathbf{p}}_G^F = \widetilde{\mathbf{p}}_G^C. \tag{A.3}$$

Since the contact points between datums and fixture surfaces have a coordinate of 0 in $Z$ axis w.r.t. the each datum coordinate system (note that all datum CS have a $Z$-axis pointing out to the surface), the coordinate $Z$ of points $\mathbf{p_D^B}$, $\mathbf{p_E^B}$ and $\mathbf{p_G^C}$ are equal to 0. Following the steps in [10], previous equations can be simplified to the following expressions

$$\begin{bmatrix} [{}^0\mathbf{a}_B^F]^T & [\mathbf{p}_D^F \times^0 \mathbf{a}_B^F]^T \\ [{}^0\mathbf{a}_B^F]^T & [\mathbf{p}_E^F \times^0 \mathbf{a}_B^F]^T \\ [{}^0\mathbf{a}_C^F]^T & [\mathbf{p}_G^F \times^0 \mathbf{a}_C^F]^T \end{bmatrix} \cdot \mathbf{x}_F^R = \\ \begin{bmatrix} [[\boldsymbol{\theta}_B^R \times^0 \mathbf{n}_F^B]_{(3)} & [\boldsymbol{\theta}_B^R \times^0 \mathbf{o}_F^B]_{(3)} & [\boldsymbol{\theta}_B^R \times^0 \mathbf{a}_F^B]_{(3)} & \boldsymbol{\theta}_B^R \times^0 \mathbf{t}_F^B + \mathbf{d}_B^R]_{(3)}] \cdot \tilde{\mathbf{p}}_D^F \\ [\boldsymbol{\theta}_B^R \times^0 \mathbf{n}_F^B]_{(3)} & [\boldsymbol{\theta}_B^R \times^0 \mathbf{o}_F^B]_{(3)} & [\boldsymbol{\theta}_B^R \times^0 \mathbf{a}_F^B]_{(3)} & \boldsymbol{\theta}_B^R \times^0 \mathbf{t}_F^B + \mathbf{d}_B^R]_{(3)}] \cdot \tilde{\mathbf{p}}_E^F \\ [\boldsymbol{\theta}_C^R \times^0 \mathbf{n}_F^C]_{(3)} & [\boldsymbol{\theta}_C^R \times^0 \mathbf{o}_F^C]_{(3)} & [\boldsymbol{\theta}_C^R \times^0 \mathbf{a}_F^C]_{(3)} & \boldsymbol{\theta}_C^R \times^0 \mathbf{t}_F^C + \mathbf{d}_C^R]_{(3)}] \cdot \tilde{\mathbf{p}}_G^F \end{bmatrix}, \tag{A.4}$$

where $]_{(3)}$ indicates the third component of the vector, $\mathbf{d}_B^R$ and $\boldsymbol{\theta}_B^R$ define the DMV $\mathbf{x}_B^R$, and vectors ${}^0\mathbf{n}_j^i$, ${}^0\mathbf{o}_j^i$, ${}^0\mathbf{a}_j^i$ and ${}^0\mathbf{t}_j^i$ are defined for an HTM as

$${}^0\mathbf{H}_j^i = \begin{pmatrix} {}^0\mathbf{n}_j^i & {}^0\mathbf{o}_j^i & {}^0\mathbf{a}_j^i & {}^0\mathbf{t}_j^i \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{A.5}$$

Since the RCS is the primary datum (A-CS), $\mathbf{x}_F^R$ has only 3 non-zero values in previous Eq. (A.4). For the fixture and part geometry given in Figure 3.4, the resolution of previous equation is

$$\begin{aligned}
\mathbf{x}_F^R(2) &= -d_{Cz}^R + (p_{Gz}^F - t_{Fy}^C)\cdot\theta_{Cx}^R + (t_{Fx}^C - p_{Gx}^F)\cdot\theta_{Cy}^R \\
&\quad + p_{Gz}^F\cdot\theta_{Bx}^R, && \text{(A.6)}\\
\mathbf{x}_F^R(3) &= -d_{Bz}^R - t_{Fy}^B\cdot\theta_{Bx}^R + (p_{Ex}^F - t_{Fx}^B)\cdot\theta_{By}^R, && \text{(A.7)}\\
\mathbf{x}_F^R(4) &= \theta_{Bx}^R, && \text{(A.8)}
\end{aligned}$$

where $\mathbf{p}_G^F = [t_{3x}^F, 0, t_{3z}^F]$ and the value of $p_{Ex}^F$ is the parameter $a$ which can be 0 or $L_s$ depending on the fixture and part assembly.

## A.2 Calculus of self-centering error due to single jaw deviations

Let us consider that the guiding slots of a 3-jaw self-centering chuck can be defined in a 2D plane as three lines which start from the coordinate origin, separated 120° to each other. Each jaw can be defined as a point located in each line, as it can be seen in Figure A.1. In this nominal case, jaw P is located in the vertical line, with jaws Q and R named in clockwise order. Assuming a perfectly working chuck, all jaws are separated a distance $G$ from the origin. Therefore, given a certain $G$, the position of each jaw in this ideal chuck with respect to the nominal FCS is

$$\begin{aligned}
&(x_P, y_P) = (0, G),\\
&(x_Q, y_Q) = (-sin(30°)\cdot G, cos(30°)\cdot G),\\
&(x_R, y_R) = (-sin(30°)\cdot G, -cos(30°)\cdot G).
\end{aligned}$$

A cylinder held by the chuck is represented in the 2D plane as a circumference tangent to points P, Q and R. If the jaws are perfectly self-centered, the center of the circumference will be located in the coordinate origin. However, these jaws may present a displacement from its self-centering position. For the sake of simplicity, the displacement, named $\delta$, is applied to the single jaw P. By clamping the part under this jaw displacement, the following equations hold

$$(x_P, y_P) = (0, G + \delta), \tag{A.9}$$

$$(x_Q, y_Q) = (cos(30°)\cdot G, -sin(30°)\cdot G), \tag{A.10}$$

$$(x_R, y_R) = (-cos(30°)\cdot G, -sin(30°)\cdot G). \tag{A.11}$$

Given that the radius of the cylinder is $r$ and that the circumference must be tangent to all three points, the distance from the center of the circumference to the coordinate origin expressed here as ($x_C$, $y_C$) can be calculated using the following equations

$$(x_P - x_C)^2 + (y_P - y_C)^2 = r^2, \tag{A.12}$$

$$(x_Q - x_C)^2 + (y_Q - y_C)^2 = r^2, \tag{A.13}$$

$$(x_R - x_C)^2 + (y_R - y_C)^2 = r^2. \tag{A.14}$$

Operating with previous equations we obtain

$$x_C^2 + y_C^2 - 2\delta y_C - 2Gy_C + G^2 + 2G\delta + \delta^2 = r^2, \tag{A.15}$$

$$x_C^2 - \sqrt{3}Gx_C + y_C^2 + Gy_C + G^2 = r^2, \tag{A.16}$$

$$x_C^2 + \sqrt{3}Gx_C + y_C^2 + Gy_C + G^2 = r^2. \tag{A.17}$$

Solving these equations, we obtain that the deviation of the center $x_C$ and $y_C$ as

$$x_C = 0, \tag{A.18}$$

$$y_C = \frac{\delta^2 + 2\delta G}{2\delta + 3G}. \tag{A.19}$$

Finally, assuming that the displacement $\delta$ will be some orders of magnitude smaller than the value of the distance of the jaws to the center $G$, the previous equation can be approximated to

$$y_C = \frac{\delta^2 + 2\delta G}{2\delta + 3G} \approx \frac{2\delta G}{3G} = \frac{2}{3}\delta. \tag{A.20}$$

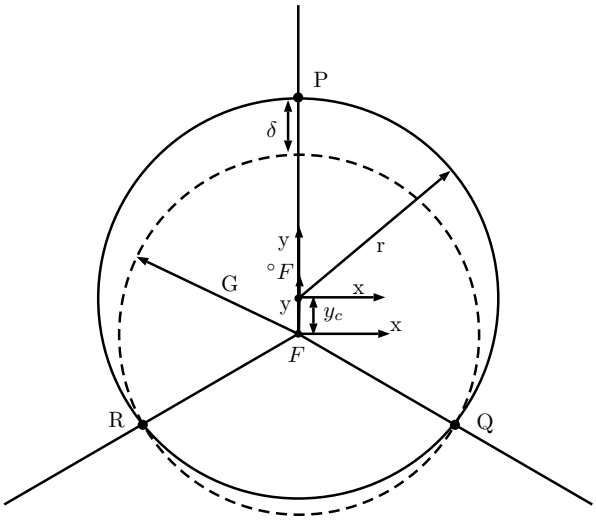



**Figure A.1.** Self-centering error $y_C$ due to single jaw deviation $\delta$. Dimensions of $\delta$ have been exaggerated to improve comprehension.

## A.3 Calculus of datum-induced errors in 3-jaw self-centering chucks

In order to calculate matrix $\mathbf{A}_2$, let $\mathbf{p}_L$ be the locating pin that touches the secondary datum defined by B-CS in a 3-jaw chuck with configuration 1, as it is shown in Fig. 3.9. The nominal coordinates are defined as

$$\mathbf{H}_R^B \cdot \mathbf{H}_F^R \cdot \widetilde{\mathbf{p}}_L^F = \widetilde{\mathbf{p}}_L^B. \tag{A.21}$$

Additionally, the following expressions hold

$$\mathbf{H}_R^B = (\mathbf{H}_B^R)^{-1} = ({}^0\mathbf{H}_B^R \delta\mathbf{H}_B^R)^{-1} = (\delta\mathbf{H}_B^R)^{-1} \cdot {}^0\mathbf{H}_R^B = (\mathbf{I} - \boldsymbol{\Delta}_B^R) \cdot {}^0\mathbf{H}_R^B, \tag{A.22}$$

$$\mathbf{H}_F^R = {}^0\mathbf{H}_F^R \cdot (\boldsymbol{\Delta}_F^R + \mathbf{I}). \tag{A.23}$$

Therefore, substituting Eq. (A.22) and (A.23) in (A.21)

$$(\mathbf{I} - \boldsymbol{\Delta}_B^R) \cdot {}^0\mathbf{H}_R^B \cdot {}^0\mathbf{H}_F^R \cdot (\boldsymbol{\Delta}_F^R + \mathbf{I}) \cdot \widetilde{\mathbf{p}}_L^F = \widetilde{\mathbf{p}}_L^B. \tag{A.24}$$

Neglecting the second order terms

$$(-\boldsymbol{\Delta}_B^R \cdot {}^0\mathbf{H}_F^B + {}^0\mathbf{H}_F^B \cdot \boldsymbol{\Delta}_F^R + {}^0\mathbf{H}_F^B) \cdot \widetilde{\mathbf{p}}_L^F \approx \widetilde{\mathbf{p}}_L^B. \tag{A.25}$$

Considering that the 3 jaws and the locating points are perfect (no fixture errors), the locating point touches the secondary datum and thus, the Z coordinate of $\widetilde{\mathbf{p}}_L^B$ is zero. Therefore,

$$\left[\left( - \boldsymbol{\Delta}_B^R \cdot {}^0\mathbf{H}_F^B + {}^0\mathbf{H}_F^B \cdot \boldsymbol{\Delta}_F^R + {}^0\mathbf{H}_F^B\right) \cdot \widetilde{\mathbf{p}}_L^F\right]_{(3)} = 0. \tag{A.26}$$

Since locating pin touches the datum under nominal conditions, $\left[{}^0\mathbf{H}_F^2 \cdot \widetilde{\mathbf{p}}_L^F\right]_{(3)} = 0$, then:

$$\left[\boldsymbol{\Delta}_B^R \cdot {}^0\mathbf{H}_F^B \cdot \widetilde{\mathbf{p}}_L^F\right]_{(3)} = \left[{}^0\mathbf{H}_F^B \cdot \boldsymbol{\Delta}_F^R \cdot \widetilde{\mathbf{p}}_L^F\right]_{(3)}. \tag{A.27}$$

We can rewrite the left hand of Eq. (A.27) as

$$[\boldsymbol{\Delta}_B^R \cdot {}^0\mathbf{H}_F^B \cdot \widetilde{\mathbf{p}}_L^F]_{(3)} = \begin{bmatrix} [\boldsymbol{\theta}_B^R \times {}^0\boldsymbol{n}_F^B]_{(3)} \\ [\boldsymbol{\theta}_B^R \times {}^0\boldsymbol{o}_F^B]_{(3)} \\ [\boldsymbol{\theta}_B^R \times {}^0\boldsymbol{a}_F^B]_{(3)} \\ [\boldsymbol{\theta}_B^R \times {}^0\boldsymbol{t}_F^B + \mathbf{d}_B^R]_{(3)} \end{bmatrix}^T \cdot \widetilde{\mathbf{p}}_L^F, \tag{A.28}$$

whereas the right hand of Eq. (A.27) is rewritten as

$$[{}^0\mathbf{H}_F^B \cdot \boldsymbol{\Delta}_F^R \cdot \widetilde{\mathbf{p}}_L^F]_{(3)} = \begin{bmatrix} [{}^0\boldsymbol{a}_B^F]^T & [\mathbf{p}_L^F \times {}^0\boldsymbol{a}_B^F]^T \end{bmatrix} \cdot \mathbf{x}_F^R. \tag{A.29}$$

Solving Eq. (A.27) and reorganizing the terms, the deviation of the FCS w.r.t. RCS in Z axis direction is defined as

$$\mathbf{x}_F^R(3) = -d_{Bz}^R - p_{Lx}^F \cdot \theta_{By}^R - p_{Ly}^F \cdot \theta_{Bx}^R. \tag{A.30}$$

## A.4 Implementation guide to practitioners

In order to facilitate the industrial application of the SoV model based on workholding systems such as vices and 3-jaw chucks, we present the following step-by-step implementation guide. Please, note that the final purpose of this guide is to estimate part quality according to workholding specifications.

1. Identify the main key characteristics of the workholding systems to be used. For vices, parameters such as length and height of jaws ($L_v$, $H_v$) and contact width between support and workpiece ($W_s$). Additionally, position of primary, secondary and tertiary locating features (1-CS, 2-CS and 3-CS) w.r.t. FCS should be given. For 3-jaw chucks, parameters such as jaws position w.r.t. machine-tool $+Y$ direction ($\Omega$), and locator position if exists ($r_{loc}$ and $\theta_{loc}$).

2. Build matrices $\mathbf{A}_k^2$ and $\mathbf{A}_k^3$ according to Eqs. (3.12), (3.18), (3.19) and Eqs. (3.21), (3.22), (3.27) for vices and 3-jaw chucks, respectively. In vices, these matrices will depend on the relationship between datum and fixture errors (i.e., matrices $\Gamma_P$ and $\Gamma_F$).

3. Build the SoV model applying the methodology presented in Zhou et al. [10]. The SoV model requires the matrices previously derived $\mathbf{A}_k^2$ and $\mathbf{A}_k^3$. Include these matrices to obtain the SoV model in the form of Eq. (3.1).

4. Identify the technical specifications of the workholding systems provided by vendors. For vices, identify clamping accuracy, parallelism and perpendicularity of jaws; for 3-jaw chucks, identify maximum TIR (total indicator runout) in radial and axial direction and the dimensions of the part tested (diameter and length $D_t$ and $L_t$) in the calibration sheet.

5. Run $M$ Monte Carlo simulations constrained to previous technical specifications to generate $M$ possible sets of fixture errors for each stage ($\mathbf{u}_k^F$). Eqs. (3.14), (3.15), (3.23) and (3.24) show some constrains for vices and 3-jaw chucks according to their technical specifications, so the generated data should be within them.

6. Apply the SoV model using the simulated fixture errors $\mathbf{u}_k^F$ to estimate the deviation of the inspected features from nominal values for the $M$ simulations. An analysis of the deviations of the inspected features for the $M$ simulations will show the capability of the process and the expected quality of the part.

Appendix B

# Calculus of additional matrices for data-driven adjustment of variation propagation models

## B.1 Obtaining $\vec{\Sigma}_y$

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

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

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

$$diag\,(B + C) = diag\,(B) + diag\,(C)\,.$$

According to [174], 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

$$diag\left(AXA^{\intercal}\right) = A^{\circ 2} diag(X). \tag{B.1}$$

Applying (B.1) into (4.5), and assuming that $\Sigma_u$ is a diagonal matrix, then

$$\vec{\Sigma}_y = diag(\Sigma_y) = diag(\Gamma\Sigma_u\Gamma^{\intercal} + \Sigma_v) = diag(\Gamma\Sigma_u\Gamma^{\intercal}) + diag(\Sigma_v). \tag{B.2}$$

Substituting (4.7), (4.9) and (4.10) into (B.2), we obtain (4.11).

# Bibliography


[1] Yi Wang, Hai-Shu Ma, Jing-Hui Yang, and Ke-Sheng Wang. Industry 4.0: a way from mass customization to mass personalization production. *Advances in manufacturing*, 5(4):311–320, 2017.

[2] Bianca Caiazzo, Mario Di Nardo, Teresa Murino, Alberto Petrillo, Gianluca Piccirillo, and Stefania Santini. Towards zero defect manufacturing paradigm: A review of the state-of-the-art methods and open challenges. *Computers in Industry*, 134:103548, 2022.

[3] Foivos Psarommatis, João Sousa, João Pedro Mendonça, and Dimitris Kiritsis. Zero-defect manufacturing the approach for higher manufacturing sustainability in the era of industry 4.0: A position paper. *International Journal of Production Research*, 60(1):73–91, 2022.

[4] Daryl Powell, Maria Chiara Magnanini, Marcello Colledani, and Odd Myklebust. Advancing zero defect manufacturing: A state-of-the-art perspective and future research directions. *Computers in Industry*, 136:103596, 2022.

[5] Foivos Psarommatis. A generic methodology and a digital twin for zero defect manufacturing (ZDM) performance mapping towards design for ZDM. *Journal of Manufacturing Systems*, 59:507–521, 2021.

[6] Foivos Psarommatis, Gökan May, Paul-Arthur Dreyfus, and Dimitris Kiritsis. Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research. *International journal of production research*, 58(1):1–17, 2020.

[7] Foivos Psarommatis and Dimitris Kiritsis. A scheduling tool for achieving zero defect manufacturing (ZDM): a conceptual framework. In *IFIP International Conference on Advances in Production Management Systems*, pages 271–278. Springer, 2018.

[8] Ke-Sheng Wang. Towards zero-defect manufacturing (ZDM)—a data mining approach. *Advances in Manufacturing*, 1(1):62–74, 2013.

[9] Jianjun Shi. *Stream of Variation Modeling and Analysis for Multistage Manufacturing Processes*. CRC press, 2006.

[10] Shiyu Zhou, Qiang Huang, and Jianjun Shi. State space modeling of dimensional variation propagation in multistage machining process using differential motion vectors. *IEEE Transactions on Robotics and Automation*, 19(2):296–309, 2003.

[11] Biao Lu and Xiaojun Zhou. Opportunistic preventive maintenance scheduling for serial-parallel multistage manufacturing systems with multiple streams of deterioration. *Reliability Engineering & System Safety*, 168:116–127, 2017.

[12] Biao Lu and Xiaojun Zhou. Quality and reliability oriented maintenance for multistage manufacturing systems subject to condition monitoring. *Journal of Manufacturing Systems*, 52:76–85, 2019.

[13] Cheng Zhu, Qing Chang, and Jorge Arinez. Data-enabled modeling and analysis of multistage manufacturing systems with quality rework loops. *Journal of Manufacturing Systems*, 56:573–584, 2020.

[14] Yu Ding, Jianjun Shi, and Dariusz Ceglarek. Diagnosability analysis of multi-station manufacturing processes. *J. Dyn. Sys., Meas., Control*, 124(1):1–13, 2002.

[15] Yibo Jiao and Dragan Djurdjanovic. Compensability of errors in product quality in multistage manufacturing processes. *Journal of Manufacturing Systems*, 30(4):204–213, 2011.

[16] Robert Gao, Lihui Wang, Roberto Teti, David Dornfeld, Soundar Kumara, Masahiko Mori, and Moneer Helu. Cloud-enabled prognosis for manufacturing. *CIRP annals*, 64(2):749–772, 2015.

[17] Rubén Moliner-Heredia, José Vicente Abellán-Nebot, and Ignacio Peñarrocha-Alós. Extension of the stream-of-variation model for general-purpose workholding devices: Vices and three-jaw chucks. *IEEE Transactions on Automation Science and Engineering*, 2021.

[18] Rubén Moliner-Heredia, Gracia M Bruscas-Bellido, José V Abellán-Nebot, and Ignacio Peñarrocha-Alós. A sequential inspection procedure for fault detection in multistage manufacturing processes. *Sensors*, 21(22):7524, 2021.

[19] Rubén Moliner-Heredia, José V Abellán-Nebot, and Ignacio Peñarrocha-Alós. Model-based observer proposal for surface roughness monitoring. *Procedia Manufacturing*, 41:618–625, 2019.

[20] Rubén Moliner-Heredia, Ignacio Peñarrocha-Alós, and José Vicente Abellán-Nebot. Model-based tool condition prognosis using power consumption and scarce surface roughness measurements. *Journal of Manufacturing Systems*, 61:311–325, 2021.

[21] José V Abellán-Nebot, Rubén Moliner-Heredia, Gracia M Bruscas, and J Serrano. Variation propagation of bench vises in multi-stage machining processes. *Procedia Manufacturing*, 41:906–913, 2019.

[22] Rubén Moliner-Heredia, Ignacio Peñarrocha-Alós, and Roberto Sanchis-Llopis. Economic model predictive control of wastewater treatment plants based on bsm1 using linear prediction models. In *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, pages 73–78. IEEE, 2019.

[23] David Tena, Ignacio Peñarrocha-Alós, Roberto Sanchis, and Rubén Moliner-Heredia. Ammonium sensor fault detection in wastewater treatment plants. In *ICINCO*, pages 681–688, 2020.

[24] ISO. ISO 9000:2015. Quality management systems — Fundamentals and vocabulary. Technical report, International Organization for Standardization, Geneva, CH, 2015.

[25] Rolf Isermann. Process fault detection based on modeling and estimation methods—a survey. *automatica*, 20(4):387–404, 1984.

[26] Jie Chen and Ron J Patton. *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media, 2012.

[27] Steven X Ding. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media, 2008.

[28] Graham Clifford Goodwin, Stefan F Graebe, Mario E Salgado, et al. *Control system design*, volume 240. Prentice Hall Upper Saddle River, 2001.

[29] ISO. ISO 17450-1:2011. Geometrical product specifications (GPS) — General concepts — Part 1: Model for geometrical specification and verification. Technical report, International Organization for Standardization, Geneva, CH, 2011.

[30] ISO. ISO 5459:2011. Geometrical product specifications (GPS) — Geometrical tolerancing — Datums and datum systems. Technical report, International Organization for Standardization, Geneva, CH, 2011.

[31] R Kesavan. *Process, Planning And Cost Estimation*. New Age International, 2004.

[32] ISO. ISO 8688-2:1989 Tool life testing in milling — Part 2: End milling. Technical report, International Organization for Standardization, Geneva, CH, 1989.

[33] T Mohanraj, S Shankar, R Rajasekar, NR Sakthivel, and Alokesh Pramanik. Tool condition monitoring techniques in milling process—a review. *Journal of Materials Research and Technology*, 9(1):1032–1042, 2020.

[34] Frederick Winslow Taylor. *On the Art of Cutting Metals...*, volume 23. American society of mechanical engineers, 1906.

[35] Kunpeng Zhu and Yu Zhang. A generic tool wear model and its application to force modeling and wear monitoring in high speed milling. *Mechanical Systems and Signal Processing*, 115:147–161, 2019.

[36] H Shao, HL Wang, and XM Zhao. A cutting power model for tool wear monitoring in milling. *International Journal of Machine Tools and Manufacture*, 44(14):1503–1509, 2004.

[37] Pavel Kovac, Dragan Rodic, Vladimir Pucovsky, Branislav Savkovic, and Marin Gostimirovic. Application of fuzzy logic and regression analysis for modeling surface roughness in face milliing. *Journal of Intelligent manufacturing*, 24(4):755–762, 2013.

[38] Lennart Ljung et al. *System Identification: Theory for the user*. Prentice-hall, Inc., 1999.

[39] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[40] Fuzhen Zhang. *The Schur complement and its applications*, volume 4. Springer Science & Business Media, 2006.

[41] Stephen Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[42] Juanan Arrieta, Ander Azkarate, and Marcello Colledani. Zero defect manufacturing FR004. Roadmap. `https://focusonfof.eu/downloads/results`, 2016. Accessed: 2019-05-12.

[43] Jionghua Jin and Jianjun Shi. State Space Modeling of Sheet Metal Assembly for Dimensional Control. *Journal of Manufacturing Science and Engineering*, 1999.

[44] Dragan Djurdjanovic and Jun Ni. Dimensional Errors of Fixtures, Locating and Measurement Datum Features in the Stream of Variation Modeling in Machining . *Journal of Manufacturing Science and Engineering*, 125(4):716–730, 11 2003.

[45] Richard P Paul. *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*. MIT Press, 1981.

[46] José V Abellán-Nebot, Jian Liu, F. Romero Subirón, and Jianjun Shi. State Space Modeling of Variation Propagation in Multistation Machining Processes Considering Machining-Induced Variations. *Journal of Manufacturing Science and Engineering*, 2012.

[47] Jaime Camelio, S. Jack Hu, and Dariusz Ceglarek. Modeling Variation Propagation of Multi-Station Assembly Systems With Compliant Parts. *Journal of Mechanical Design*, 125(4):673, 2003.

[48] Jean Philippe Loose, Shiyu Zhou, and Dariusz Ceglarek. Kinematic analysis of dimensional variation propagation for multistage machining processes with general fixture layouts. *IEEE Transactions on Automation Science and Engineering*, 4(2):141–151, 2007.

[49] Jian Liu, Jionghua Jin, and Jianjun Shi. State space modeling for 3-D variation propagation in rigid-body multistage assembly processes. *IEEE Transactions on Automation Science and Engineering*, 7(2):274–290, 2009.

[50] Tingyu Zhang and Jianjun Shi. Stream of Variation Modeling and Analysis for Compliant Composite Part Assembly— Part II: Multistation Processes. *Journal of Manufacturing Science and Engineering*, 138(12), jul 2016.

[51] José V. Abellán-Nebot, I. Peñarrocha, E. Sales-Setién, and J. Liu. Optimal inspection/actuator placement for robust dimensional compensation in multistage manufacturing processes. In J. Paulo Davim, editor, *Computational Methods and Production Engineering*, Woodhead Publishing Reviews: Mechanical Engineering Series, pages 31 – 50. Woodhead Publishing, 2017.

[52] D Djurdjanovic and J Ni. Online stochastic control of dimensional quality in multistation manufacturing systems. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 221(5):865–880, 2007.

[53] Yibo Jiao and Dragan Djurdjanovic. Compensability of errors in product quality in multistage manufacturing processes. *Journal of Manufacturing Systems*, 30(4):204–213, oct 2011.

[54] Jing Zhong, Jian Liu, and Jianjun Shi. Predictive control considering model uncertainty for variation reduction in multistage assembly processes. *IEEE Transactions on Automation Science and Engineering*, 7(4):724–735, 2010.

[55] L Eduardo Izquierdo, Jianjun Shi, S Jack Hu, and Charles W Wampler. Feedforward control of multistage assembly processes using programmable tooling. *Trans. NAMRI/SME*, 35:295–302, 2007.

[56] Ester Sales-Setién, Ignacio Peñarrocha-Alós, and José V Abellán-Nebot. Estimation of nonstationary process variance in multistage manufacturing processes using a model-based observer. *IEEE Transactions on Automation Science and Engineering*, 16(2):741–754, 2018.

[57] José V. Abellán-Nebot, Jian Liu, and F. Romero Subirón. Design of multi-station manufacturing processes by integrating the stream-of-variation model and shop-floor data. *Journal of Manufacturing Systems*, 30(2):70–82, 2011.

[58] Vincent Mckenna, Yan Jin, Adrian Murphy, Michael Morgan, Caroline Mcclory, Colm Higgins, and Rory Collins. Process selection using variation and cost relations. In *Advances in Manufacturing Technology XXX: Proceedings of the 14th International Conference on Manufacturing Research, Incorporating the 31st National Conference on Manufacturing Research, September 6–8, 2016, Loughborough University, UK*, volume 3, page 465. IOS Press, 2016.

[59] José V. Abellán-Nebot, Jian Liu, and F. Romero Subirón. Process-oriented tolerancing using the extended stream of variation model. *Computers in Industry*, 64(5):485–498, 2013.

[60] Yong Chen, Yu Ding, Jionghua Jin, and Dariusz Ceglarek. Integration of process-oriented tolerancing and maintenance planning in design of multistation manufacturing processes. *IEEE Transactions on Automation Science and Engineering*, 3(4):440–453, 2006.

[61] M Kamali Nejad, Frédéric Vignat, Alain Desrochers, and François Villeneuve. 3D simulation of manufacturing defects for tolerance analysis. *Journal of computing and information science in engineering*, 10(2), 2010.

[62] J.V. Abellán-Nebot and J. Liu. Variation propagation modelling for multi-station machining processes with fixtures based on locating surfaces. *International Journal of Production Research*, 51(15), 2013.

[63] José V Abellán-Nebot, Rubén Moliner-Heredia, Gracia M Bruscas, and J Serrano. Variation propagation of bench vises in multi-stage machining processes. *Procedia Manufacturing*, 41:906–913, 2019.

[64] Jeongmin Byun and C Richard Liu. Methods for improving chucking accuracy. *Journal of manufacturing science and engineering*, 134(5), 2012.

[65] Jean-Philippe Loose, Qiang Zhou, Shiyu Zhou, and Dariusz Ceglarek. Integrating GD&T into dimensional variation models for multistage machining processes. *International Journal of Production Research*, 48(11):3129–3149, 2010.

[66] Bryan R Fischer. *Mechanical tolerance stackup and analysis*. CRC Press, 2011.

[67] G Pahlitzsch and W Hellwig. The clamping accuracy of three-jaw chucks. In *Advances in Machine Tool Design and Research 1967*, pages 97–118. Elsevier, 1968.

[68] Jeongmin Byun and CR Liu. Selection of major locating surface for improving chucking accuracy. In *International Manufacturing Science and Engineering Conference*, volume 43628, pages 489–497, 2009.

[69] Hyun-Jin Kim, Kwang-Jae Kim, and Doh-Soon Kwak. A case study on modeling and optimizing photolithography stage of semiconductor fabrication process. *Quality and Reliability Engineering International*, 26(7):765–774, 2010.

[70] José V Abellán-Nebot. Derivation and application of the stream of variation model to the manufacture of ceramic floor tiles. *Quality Engineering*, 30(4):713–729, 2018.

[71] Gökan May and Dimitris Kiritsis. 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*, pages 142–152. Springer, 2019.

[72] L Eduardo Izquierdo, Jianjun Shi, S Jack Hu, and Charles W Wampler. Feedforward control of multistage assembly processes using programmable tooling. *Trans. NAMRI/SME*, 35:295–302, 2007.

[73] José V Abellán-Nebot, Ignacio Penarrocha, Ester Sales-Setién, and Jian Liu. Optimal inspection/actuator placement for robust dimensional compensation in multistage manufacturing processes. In *Computational Methods and Production Engineering*, pages 31–50. Elsevier, 2017.

[74] Yinhua Liu, Rui Sun, and Sun Jin. A survey on data-driven process monitoring and diagnostic methods for variation reduction in multi-station assembly systems. *Assembly Automation*, 2019.

[75] Yinhua Liu, Rui Sun, Yuwei Lu, and Shiming Zhang. A knowledge-based online fault detection method of the assembly process considering the relative poses of components. *International Journal of Precision Engineering and Manufacturing*, 20(10):1705–1720, 2019.

[76] DW APLEY and J SHI. Diagnosis of multiple fixture faults in panel assembly. *Journal of manufacturing science and engineering*, 120(4):793–801, 1998.

[77] D Ceglarek and PKS Prakash. Enhanced piecewise least squares approach for diagnosis of ill-conditioned multistation assembly with compliant parts. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 226(3):485–502, 2012.

[78] Shiyu Zhou, Yong Chen, and Jianjun Shi. Statistical estimation and testing for variation root-cause identification of multistage manufacturing processes. *IEEE Transactions on Automation Science and Engineering*, 1(1):73–83, 2004.

[79] D CEGLAREK and J SHI. Fixture failure diagnosis for autobody assembly using pattern recognition. *Journal of engineering for industry*, 118(1):55–66, 1996.

[80] Yu Ding, Dariusz Ceglarek, and Jianjun Shi. Fault diagnosis of multistage manufacturing processes by using state space approach. *J. Manuf. Sci. Eng.*, 124(2):313–322, 2002.

[81] Nong Jin and Shiyu Zhou. Signature construction and matching for fault diagnosis in manufacturing processes through fault space analysis. *IIE transactions*, 38(4):341–354, 2006.

[82] Yinhua Liu, Xialiang Ye, and Sun Jin. A bayesian based process monitoring and fixture fault diagnosis approach in the auto body assembly process. *Journal of Shanghai Jiaotong University (Science)*, 21(2):164–172, 2016.

[83] Jeongsu Lee, Young Chul Lee, and Jeong Tae Kim. Migration from the traditional to the smart factory in the die-casting industry: Novel process data acquisition and fault detection based on artificial neural network. *Journal of materials processing technology*, 290:116972, 2021.

[84] Jianjun Shi and Shiyu Zhou. Quality control and improvement for multistage systems: A survey. *Iie Transactions*, 41(9):744–753, 2009.

[85] Gianfranco Genta, Maurizio Galetto, and Fiorenzo Franceschini. Inspection procedures in manufacturing processes: recent studies and research perspectives. *International Journal of Production Research*, 58(15):4767–4788, 2020.

[86] Xuemei Shan and Daniel W Apley. Blind identification of manufacturing variation patterns by combining source separation criteria. *Technometrics*, 50(3):332–343, 2008.

[87] Li Zeng and Shiyu Zhou. Inferring the interactions in complex manufacturing processes using graphical models. *Technometrics*, 49(4):373–381, 2007.

[88] Jian Liu, Jianjun Shi, and S Jack Hu. Engineering-driven factor analysis for variation source identification in multistage manufacturing processes. *Journal of manufacturing science and engineering*, 130(4), 2008.

[89] Fuyong Yang, Sun Jin, and Zhimin Li. A comprehensive study of linear variation propagation modeling methods for multistage machining processes. *The International Journal of Advanced Manufacturing Technology*, 90(5):2139–2151, 2017.

[90] JIONGHUA JIN and JIANJUN SHI. State space modeling of sheet metal assembly for dimensional control. *Journal of manufacturing science and engineering*, 121(4):756–762, 1999.

[91] Jaime Camelio, S Jack Hu, and Dariusz Ceglarek. Modeling variation propagation of multi-station assembly systems with compliant parts. *J. Mech. Des.*, 125(4):673–681, 2003.

[92] Tingyu Zhang and Jianjun Shi. Stream of variation modeling and analysis for compliant composite part assembly—part I: Single-station processes. *Journal of Manufacturing Science and Engineering*, 138(12), 2016.

[93] Tingyu Zhang and Jianjun Shi. Stream of variation modeling and analysis for compliant composite part assembly—part II: Multistation processes. *Journal of Manufacturing Science and Engineering*, 138(12), 2016.

[94] José V Abellan-Nebot, Jian Liu, Fernando Romero Subirón, and Jianjun Shi. State space modeling of variation propagation in multistation machining processes considering machining-induced variations. *Journal of Manufacturing Science and Engineering*, 134(2), 2012.

[95] Jean-Philippe Loose, Shiyu Zhou, and Dariusz Ceglarek. Kinematic analysis of dimensional variation propagation for multistage machining processes with general fixture layouts. *IEEE Transactions on Automation Science and Engineering*, 4(2):141–152, 2007.

[96] V Roshan Joseph and Huan Yan. Engineering-driven statistical adjustment and calibration. *Technometrics*, 57(2):257–267, 2015.

[97] Jinjiang Wang, Yilin Li, Robert X Gao, and Fengli Zhang. Hybrid physics-based and data-driven models for smart manufacturing: Modelling, simulation, and explainability. *Journal of Manufacturing Systems*, 63:381–391, 2022.

[98] Yu Ding, Shiyu Zhou, and Yong Chen. A comparison of process variation estimators for in-process dimensional measurements and control. *Journal of dynamic systems, measurement, and control*, 127(1):69–79, 2005.

[99] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.

[100] Didier Henrion, Jean-Bernard Lasserre, and Johan Löfberg. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.

[101] Jean B Lasserre. A semidefinite programming approach to the generalized problem of moments. *Mathematical Programming*, 112(1):65–92, 2008.

[102] J. Löfberg. Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[103] Giacomo Angione, Cristina Cristalli, José Barbosa, and Paulo Leitão. Integration challenges for the deployment of a multi-stage zero-defect manufacturing architecture. In *2019 IEEE 17th International Conference on Industrial Informatics (INDIN)*, volume 1, pages 1615–1620. IEEE, 2019.

[104] Mohammad Rezaei-Malek, Mehrdad Mohammadi, Jean-Yves Dantan, Ali Siadat, and Reza Tavakkoli-Moghaddam. A review on optimisation of part quality inspection planning in a multi-stage manufacturing system. *International Journal of Production Research*, 57(15-16):4880–4897, 2019.

[105] Kaveh Bastani, Zhenyu Kong, Wenzhen Huang, and Yingqing Zhou. Compressive sensing–based optimal sensor placement and fault diagnosis for multi-station assembly processes. *IIE Transactions*, 48(5):462–474, 2016.

[106] Zezhong Wang, Gareth Thomson, Yuchun Xu, and Xianghong Ma. Review of automated inspection, equipment monitoring and optimization of manufacturing.

[107] Shiyu Zhou, Yu Ding, Yong Chen, and Jianjun Shi. Diagnosability study of multistage manufacturing processes based on linear mixed-effects models. *Technometrics*, 45(4):312–325, 2003.

[108] Liming Xiang and Fugee Tsung. Statistical monitoring of multi-stage processes based on engineering models. *IIE transactions*, 40(10):957–970, 2008.

[109] Yanting Li and Fugee Tsung. Detecting and diagnosing covariance matrix changes in multistage processes. *Iie Transactions*, 43(4):259–274, 2011.

[110] Partha Protim Mondal, Placid Matthew Ferreira, Shiv Gopal Kapoor, and Patrick N Bless. Monitoring and diagnosis of multistage manufacturing processes using hierarchical bayesian networks. *Procedia Manufacturing*, 53:32–43, 2021.

[111] Dewen Yu, Junkang Guo, Qiangqiang Zhao, Dingtang Zhao, and Jun Hong. Fault diagnosis for underdetermined multistage assembly processes via an enhanced bayesian hierarchical model. *Journal of Manufacturing Systems*, 58:280–290, 2021.

[112] Kaveh Bastani, Babak Barazandeh, and Zhenyu James Kong. Fault diagnosis in multi-station assembly systems using spatially correlated bayesian learning algorithm. *Journal of Manufacturing Science and Engineering*, 140(3), 2018.

[113] Ruonan Liu, Boyuan Yang, Enrico Zio, and Xuefeng Chen. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, 108:33–47, 2018.

[114] Elpiniki I Papageorgiou, Theodosis Theodosiou, George Margetis, Nikolaos Dimitriou, Paschalis Charalampous, Dimitrios Tzovaras, and Ioannis Samakovlis. Short survey of artificial intelligent technologies for defect detection in manufacturing. In *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–7. IEEE, 2021.

[115] Moschos Papananias, Thomas E McLeay, Olusayo Obajemu, Mahdi Mahfouf, and Visakan Kadirkamanathan. Inspection by exception: A new machine learning-based approach for multistage manufacturing. *Applied Soft Computing*, 97:106787, 2020.

[116] Gerardo Beruvides, Alberto Villalonga, Pasquale Franciosa, Darek Ceglarek, and Rodolfo E Haber. Fault pattern identification in multi-stage assembly processes with non-ideal sheet-metal parts based on reinforcement learning architecture. *Procedia Cirp*, 67:601–606, 2018.

[117] Chatchawan Sornsiri and Chatchapol Chungchoo. Application of tolerance charting using rooted tree graph for allocating manufacturing specifications onto the precision machined part: Detailed explanation of manufacturing a standard weight of mass. *Int. J. Appl. Eng. Res*, 12:7838–7852, 2017.

[118] Douglas C Montgomery. *Introduction to statistical quality control*. John Wiley & Sons, 2020.

[119] Alberto Gonzalez-Sanchez, Eric Piel, Hans-Gerhard Gross, and Arjan JC van Gemund. Prioritizing tests for software fault localization. In *2010 10th International Conference on Quality Software*, pages 42–51. IEEE, 2010.

[120] Alberto Gonzalez-Sanchez, H Gross, and Arjan JC Van Gemund. Performance modeling of sequential diagnosis algorithms. In *Proceedings of the 22nd Int'l Workshop on the Principles of Diagnosis, Murnau am Staffelsee, Germany*, pages 4–7, 2011.

[121] Richard A Johnson. An information theory approach to diagnosis. *IRE Transactions on Reliability and Quality Control*, 1:35–35, 1960.

[122] D Yu Pimenov, Andrés Bustillo, and Tadeusz Mikolajczyk. Artificial intelligence for automatic prediction of required surface roughness by monitoring wear on face mill teeth. *Journal of Intelligent Manufacturing*, 29(5):1045–1061, 2018.

[123] Ricardo Coppel, Jose V Abellan-Nebot, Hector R Siller, Ciro A Rodriguez, and Federico Guedea. Adaptive control optimization in micro-milling of hardened steels—evaluation of optimization approaches. *The International Journal of Advanced Manufacturing Technology*, 84(9):2219–2238, 2016.

[124] E Garcia Plaza, PJ Nunez Lopez, and EM Beamud Gonzalez. Multi-sensor data fusion for real-time surface quality control in automated machining systems. *Sensors*, 18(12):4381, 2018.

[125] Jose Vicente Abellan-Nebot and Fernando Romero Subirón. A review of machining monitoring systems based on artificial intelligence process models. *The International Journal of Advanced Manufacturing Technology*, 47(1):237–257, 2010.

[126] José V Abellán-Nebot. A review of artificial intelligent approaches applied to part accuracy prediction. *International Journal of Machining and Machinability of Materials*, 8(1-2):6–37, 2010.

[127] Roberto Teti, Krzysztof Jemielniak, Garret O'Donnell, and David Dornfeld. Advanced monitoring of machining operations. *CIRP annals*, 59(2):717–739, 2010.

[128] HR Siller, C Vila, CA Rodríguez, and JV Abellán. Study of face milling of hardened aisi d3 steel with a special design of carbide tools. *The International Journal of Advanced Manufacturing Technology*, 40(1):12–25, 2009.

[129] Wit Grzesik. Influence of tool wear on surface roughness in hard turning using differently shaped ceramic tools. *Wear*, 265(3-4):327–335, 2008.

[130] Farbod Akhavan Niaki, Martin Michel, and Laine Mears. State of health monitoring in machining: Extended Kalman filter for tool wear assessment in turning of IN718 hard-to-machine alloy. *Journal of Manufacturing Processes*, 24:361–369, 2016.

[131] Srinivasan Purushothaman. Tool wear monitoring using artificial neural network based on extended kalman filter weight updation with transformed input patterns. *Journal of Intelligent Manufacturing*, 21(6):717–730, 2010.

[132] D Jaume, M Verge, A Rault, and A Moisan. A model-based diagnosis in machine tools: application to the milling cutting process. *CIRP annals*, 39(1):443–446, 1990.

[133] Jose V Abellan-Nebot and Miguel Ortells Rogero. Sustainable machining of molds for tile industry by minimum quantity lubrication. *Journal of Cleaner Production*, 240:118082, 2019.

[134] David A Stephenson and John S Agapiou. *Metal cutting theory and practice*. CRC press, 2018.

[135] S Kurada and C Bradley. A review of machine vision sensors for tool condition monitoring. *Computers in industry*, 34(1):55–72, 1997.

[136] J Paulo Davim. *Sustainable Machining*. Springer, 2017.

[137] Amit Kumar Jain and Bhupesh Kumar Lad. A novel integrated tool condition monitoring system. *Journal of Intelligent Manufacturing*, 30(3):1423–1436, 2019.

[138] David A Stephenson and John S Agapiou. *Metal cutting theory and practice*. CRC press, 2016.

[139] Changqing Liu, Jincheng Ni, and Peng Wan. An accurate prediction method of multiple deterioration forms of tool based on multitask learning with low rank tensor constraint. *Journal of Manufacturing Systems*, 58:193–204, 2021.

[140] Jinjiang Wang, Yilin Li, Rui Zhao, and Robert X Gao. Physics guided neural network for machining tool wear prediction. *Journal of Manufacturing Systems*, 57:298–310, 2020.

[141] Norman E Woldman and Robert C Gibbons. *Machinability and machining of metals*. McGraw-Hill, 1951.

[142] Yawei Hu, Shujie Liu, Huitian Lu, and Hongchao Zhang. Remaining useful life model and assessment of mechanical products: a brief review and a note on the state space model method. *Chinese Journal of Mechanical Engineering*, 32(1):15, 2019.

[143] Xiao-Sheng Si, Wenbin Wang, Chang-Hua Hu, and Dong-Hua Zhou. Remaining useful life estimation–a review on the statistical data driven approaches. *European journal of operational research*, 213(1):1–14, 2011.

[144] Hongkun Li, Yinhu Wang, Pengshi Zhao, Xiaowen Zhang, and Peilin Zhou. Cutting tool operational reliability prediction based on acoustic emission and logistic regression model. *Journal of Intelligent Manufacturing*, 26(5):923–931, 2015.

[145] Cyril Drouillet, Jaydeep Karandikar, Chandra Nath, Anne-Claire Journeaux, Mohamed El Mansori, and Thomas Kurfess. Tool life predictions in milling using spindle power with the neural network technique. *Journal of Manufacturing Processes*, 22:161–168, 2016.

[146] Tarak Benkedjouh, Kamal Medjaher, Noureddine Zerhouni, and Saïd Rechak. Health assessment and life prediction of cutting tools based on support vector regression. *Journal of Intelligent Manufacturing*, 26(2):213–223, 2015.

[147] Jun Wu, Yongheng Su, Yiwei Cheng, Xinyu Shao, Chao Deng, and Cheng Liu. Multi-sensor information fusion for remaining useful life prediction of machining tools by adaptive network based fuzzy inference system. *Applied Soft Computing*, 68:13–23, 2018.

[148] X Li, BS Lim, JH Zhou, S Huang, SJ Phua, KC Shaw, and M Er. Fuzzy neural network modelling for tool wear estimation in dry milling operation. In *Annual conference of the prognostics and health management society*, pages 1–11, 2009.

[149] Cheng-Geng Huang, Xianhui Yin, Hong-Zhong Huang, and Yan-Feng Li. An enhanced deep learning-based fusion prognostic method for RUL prediction. *IEEE Transactions on Reliability*, 69(3):1097–1109, 2019.

[150] Hai Li, Wei Wang, Ziwei Li, Liyi Dong, and Qingzhao Li. A novel approach for predicting tool remaining useful life using limited data. *Mechanical Systems and Signal Processing*, 143:106832, 2020.

[151] Ji-Yan Wu, Min Wu, Zhenghua Chen, Xiaoli Li, and Ruqiang Yan. A joint classification-regression method for multi-stage remaining useful life prediction. *Journal of Manufacturing Systems*, 58:109–119, 2021.

[152] Houman Hanachi, Wennian Yu, Il Yong Kim, Jie Liu, and Chris K Mechefske. Hybrid data-driven physics-based model fusion framework for tool wear prediction. *The International Journal of Advanced Manufacturing Technology*, 101(9-12):2861–2872, 2019.

[153] Jinjiang Wang, Peng Wang, and Robert X Gao. Enhanced particle filter for tool wear prediction. *Journal of Manufacturing Systems*, 36:35–45, 2015.

[154] Jianlei Zhang, Binil Starly, Yi Cai, Paul H Cohen, and Yuan-Shin Lee. Particle learning in online tool wear diagnosis and prognosis. *Journal of Manufacturing Processes*, 28:457–463, 2017.

[155] Pundarikaksha Baruah and Ratna B Chinnam*. HMMs for diagnostics and prognostics in machining processes. *International Journal of Production Research*, 43(6):1275–1293, 2005.

[156] Akhilesh Kumar, Ratna Babu Chinnam, and Finn Tseng. An HMM and polynomial regression based approach for remaining useful life and health state estimation of cutting tools. *Computers & Industrial Engineering*, 128:1008–1014, 2019.

[157] Kunal Tiwari, Ameer Shaik, and N Arunachalam. Tool wear prediction in end milling of Ti-6Al-4V through Kalman filter based fusion of texture features and cutting forces. *Procedia Manufacturing*, 26:1459–1470, 2018.

[158] Peng Wang and Robert X Gao. Adaptive resampling-based particle filtering for tool life prediction. *Journal of Manufacturing Systems*, 37:528–534, 2015.

[159] Jinjiang Wang, Yinghao Zheng, Peng Wang, and Robert X Gao. A virtual sensing based augmented particle filter for tool condition prognosis. *Journal of Manufacturing Processes*, 28:472–478, 2017.

[160] Huibin Sun, Dali Cao, Zidong Zhao, and Xia Kang. A hybrid approach to cutting tool remaining useful life prediction based on the Wiener process. *IEEE Transactions on Reliability*, 67(3):1294–1303, 2018.

[161] Hua An, Guofeng Wang, Yi Dong, Kai Yang, and Lingling Sang. Tool life prediction based on Gauss importance resampling particle filter. *The International Journal of Advanced Manufacturing Technology*, 103(9):4627–4634, 2019.

[162] Rui Liu, Achyuth Kothuru, and Shuhuan Zhang. Calibration-based tool condition monitoring for repetitive machining operations. *Journal of Manufacturing Systems*, 54:285–293, 2020.

[163] Farbod Akhavan Niaki, Durul Ulutan, and Laine Mears. In-process tool flank wear estimation in machining gamma-prime strengthened alloys using Kalman filter. *Procedia Manufacturing*, 1:696–707, 2015.

[164] Paul C Paris. A rational analytic theory of fatigue. *The trend in engineering*, 13:9, 1961.

[165] Yahui Wang, Lianyu Zheng, and Yiwei Wang. Event-driven tool condition monitoring methodology considering tool life prediction based on industrial internet. *Journal of Manufacturing Systems*, 58:205–222, 2021.

[166] Yung-Chang Yen, Jörg Söhner, Blaine Lilly, and Taylan Altan. Estimation of tool wear in orthogonal cutting using the finite element analysis. *Journal of materials processing technology*, 146(1):82–91, 2004.

[167] Z Pálmai. Proposal for a new theoretical model of the cutting tool's flank wear. *Wear*, 303(1-2):437–445, 2013.

[168] H. Takeyama and R. Murata. Basic investigation of tool wear. *Journal of Engineering for Industry*, 85(1):33–37, 02 1963.

[169] Amir Malakizadi, Hans Gruber, Ibrahim Sadik, and Lars Nyborg. An FEM-based approach for tool wear estimation in machining. *Wear*, 368:10–24, 2016.

[170] D Cuppini, G D'errico, and G Rutelli. Tool wear monitoring based on cutting power measurement. *Wear*, 139(2):303–311, 1990.

[171] Hae-Sung Yoon, Jang-Yeob Lee, Min-Soo Kim, and Sung-Hoon Ahn. Empirical power-consumption model for material removal in three-axis milling. *Journal of Cleaner Production*, 78:54–62, 2014.

[172] Tugrul Özel, Yigit Karpat, Luís Figueira, and J Paulo Davim. Modelling of surface finish and tool flank wear in turning of AISI D2 steel with ceramic wiper inserts. *Journal of materials processing technology*, 189(1-3):192–198, 2007.

[173] JG Lima, RF Avila, AM Abrao, M Faustino, and J Paulo Davim. Hard turning: AISI 4340 high strength low alloy steel and AISI d2 cold work tool steel. *Journal of Materials Processing Technology*, 169(3):388–395, 2005.

[174] Shayle R Searle and Andre I Khuri. *Matrix algebra useful for statistics*. John Wiley & Sons, 2017.