Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

Technical Paper

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

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

Departament of Industrial Systems Engineering and Design, Universitat Jaume I, Castellón de la Plana, Spain

ARTICLE INFO

Keywords:
Cutting tool condition prognosis
Power consumption
Surface roughness
Adaptive recursive least squares
Remaining useful life prediction

ABSTRACT

In machining processes, underusing and overusing cutting tools directly affect part quality, entailing economic and environmental impacts. In this paper, 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.

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 [1], cutting tool failures may represent about 20% of the downtime of a machining system and it is estimated that the expense of cutting tools and their maintenance grosses about 3–12% of overall manufacturing cost [2,3]. 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 [4]. 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 [5]. 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 [6]. Additionally, in [7], physics guided neural networks have been developed to predict the state of the tool wear using deep learning techniques supported by physical

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

https://doi.org/10.1016/j.jmsy.2021.09.001
Received 8 October 2020; Received in revised form 3 August 2021; Accepted 6 September 2021
Available online 24 September 2021

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 [8]. Physics-based approaches directly use formulae from theory, such as Taylor's tool life equation [9] or other more sophisticated ones [10], to estimate the remaining useful life of the cutting tool. Data-based approaches can mainly be classified into statistical methods and artificial intelligence methods [11]. 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 [12]. Autoregressive moving average models (ARMA) and logistic regressions are common techniques applied in this field [13]. In artificial intelligence methods, techniques such as artificial neural networks (ANN) [14], support vector regression (SVR) [15], adaptive neuro-fuzzy inference systems (ANFIS) [16] or fuzzy systems [17] have been investigated. The tendency during the latest years is the research of deep learning techniques [18–20].

According to [11], 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 [21] 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 [22].

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 [21]. 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 [23]. Depending on system type and noise assumption, different approaches have been investigated such as hidden Markov models (HMM) [24,25], state-space models (SSM) with Kalman filters [26], SSM with particle filters [27] and SVR applied to a physics-based tool condition degradation model [28]. Additionally, other types of hybrid approaches have been investigated, such as fusing ANN with Wiener processes [29] or Gauss importance resampling particle filters [30], or using multiple-scenario calibration methods [31].

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 [26] 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 [32], 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 [21], 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 [27] used Paris' law [33] 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 [23], 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 [22], which uses a similar model approach as [27]. 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 [34] 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 paper 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: (i) a continuous measurement of power consumption at the machine-tool level and, (ii) 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 paper 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 paper is organized as follows: Section 2 states the problem and the direct strategies, Section 3 presents the proposed model-based approaches, and Section 4 develops the data that will be used to validate the proposed approaches. In Section 5, a simulation using the data is used to select the most fitting models. Section 6 explains the ARLS updating algorithm, Section 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 8 concludes the paper.

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 [35]. 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 straightforward strategies that will be use later to validate our proposed strategies.

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

$$P_c(k) \propto F_c(k) \propto V_b(k) \propto k, \tag{1}$$

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

$$R_a(k) \propto V_b(k) \propto k. \tag{2}$$

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.

2.2. Available measurements

In this paper, 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 paper 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 (Fig. 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

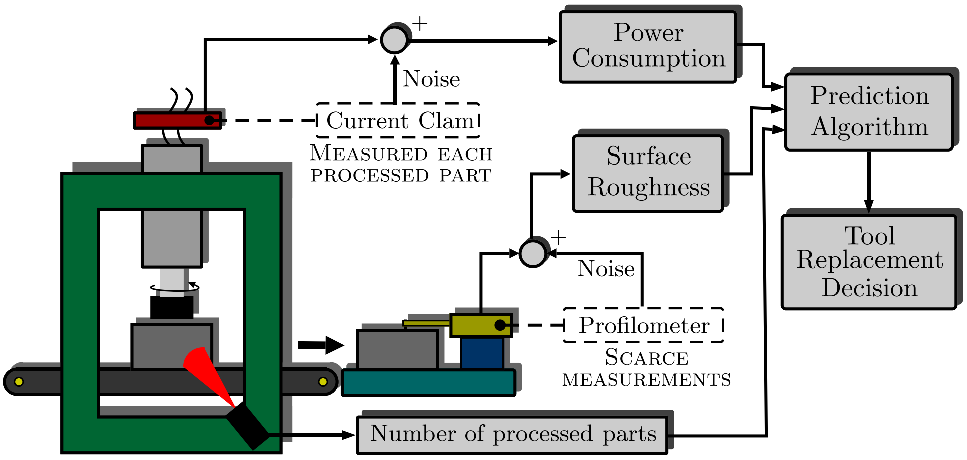


Fig. 1. Problem case description.

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.

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

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 machin ing 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 (Fig. 2).

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.

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 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 2). We wonder

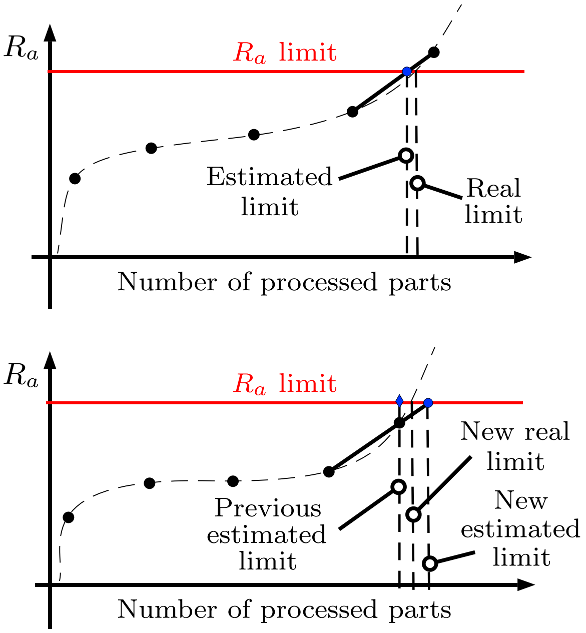


Fig. 2. Interpolation-based approach. Its precision depends on the measurement frequency.

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 4 we first obtain a set of data to explore possible models able to represent several conditions.
- In Section 5 we obtain general models that can fit the previous dataset.
- In Section 6 we present the updating procedure that allows us to perform any initial experimentation and update the models to adapt the changing cutting and material conditions.

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.

4.1. Tool flank wear dataset generation

Previous research on the evolution of tool flank wear has been compiled in [38,39]. Studies like [40] state that the tool flank wear V_b has an evolution on time t given by

$$V_b(t) = Alog(\mathrm{B}\ \mathrm{t} + 1) + \mathrm{Ct}^3, \tag{3}$$

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 [38,41,42] and references therein state that the evolution may follow a differential equation that depends on the temperature T

$$\frac{\mathrm{d}V_b(t)}{\mathrm{dt}} = A + Be^{\frac{-C}{T(t)}}, \tag{4}$$

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 1. The evolutions are shown in Fig. 3a.

4.2. Power consumption dataset generation

In order to monitor the tool condition, the authors in this paper [43] related the cutting power with the tool flank wear with a linear relationship function

$$P_c = \alpha + \beta V_b, \tag{5}$$

Table 1
Tool flank wear (V_b) dataset generation parameters.

#	Eq.	A	B	C	$-C/T$
Dataset V1	(3)	13.06	149.5	0.00506	–
Dataset V2	(3)	14.891	34.7	0.008526	–
Dataset V3	(4)	0	0.0375	–	10.39
Dataset V4	(4)	0	0.0750	–	10.39

where parameters α and β are empirical constants. These empirical constants were developed as empirical functions depending on cutting conditions in [44]. Further research [32] 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 α and β with real machining settings.

To generate the power consumption dataset, we applied several variants of the linear equation (5) to the previously shown tool flank wear datasets, using parameters from Table 2. These equation systems are shown as follows:

$$\begin{cases} P_c = \alpha + \beta V_b, \\ V_b(t) = A log(\mathrm{Bt}+1) + \mathrm{Ct}^3. \end{cases} \qquad \begin{cases} P_c = \alpha + \beta V_b, \\ V_b(t) = \displaystyle\int_{\tau=t_0}^{\tau=t} A + \mathrm{B}e^{\frac{-C}{T(\tau)}} d\tau. \end{cases}$$

Their evolution is shown in Fig. 3b. In order to model uncertainty related to the measurement of characteristic power consumption, zero-mean Gaussian noise has been added.

4.3. Surface roughness dataset generation

Surface roughness of the processed parts has frequently been related w.r.t. cutting time in the literature [45,46], along with several cutting conditions, using empirical equations. The relationship between surface roughness and tool flank wear has been researched, though. In this paper [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

$$R_a = \delta + \epsilon V_b^{\gamma}. \tag{6}$$

The surface roughness dataset has been generated using the previously shown tool flank wear dataset and Eq. (6), employing parameters from Table 3. These equation systems are shown as follows:

$$\begin{cases} R_a = \delta + \epsilon V_b^{\gamma}, \\ V_b(t) = A log(\mathrm{Bt}+1) + \mathrm{Ct}^3. \end{cases} \qquad \begin{cases} R_a = \delta + \epsilon V_b^{\gamma}, \\ V_b(t) = \displaystyle\int_{\tau=t_0}^{\tau=t} A + Be^{\frac{-C}{T(\tau)}} d\tau. \end{cases}$$

Their evolution is shown in Fig. 4a. Additionally, in Fig. 4b it is shown the relationship between the surface roughness dataset w.r.t the power consumption dataset.

Table 2
Power consumption (P_c) dataset generation parameters.

#	Eq.	α	β
Dataset P1	(5)	6000	0.5
Dataset P2	(5)	4000	2

Table 3
Surface roughness (R_a) dataset generation parameters.

#	Eq.	δ	ϵ	γ
Dataset R1	(6)	0.1	5.5	0.455
Dataset R2	(6)	0.1	5	0.7
Dataset R3	(6)	0.1	6.5	0.8
Dataset R4	(6)	0.1	4.5	0.6

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.

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

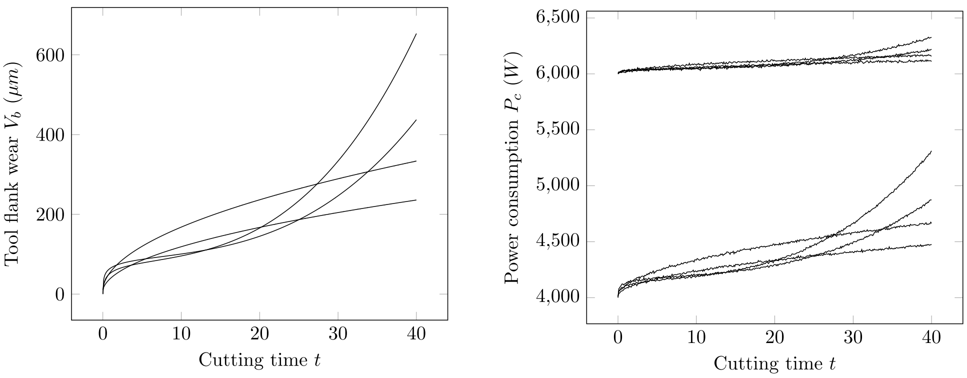
$$z(y) = \phi(x)\theta + v, \tag{7}$$

where $z(y)$ is a function of the observed value y, ϕ is the regression vector with independent variables x, and θ 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

$$p_n(x) = c_0 + c_1x + c_2x^2 + \cdots + c_nx^n = \underbrace{\begin{bmatrix} 1 & x & x^2 & \cdots & x^n \end{bmatrix}}_{\phi_n(x)} \underbrace{\begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}}_{\theta}, \tag{8}$$

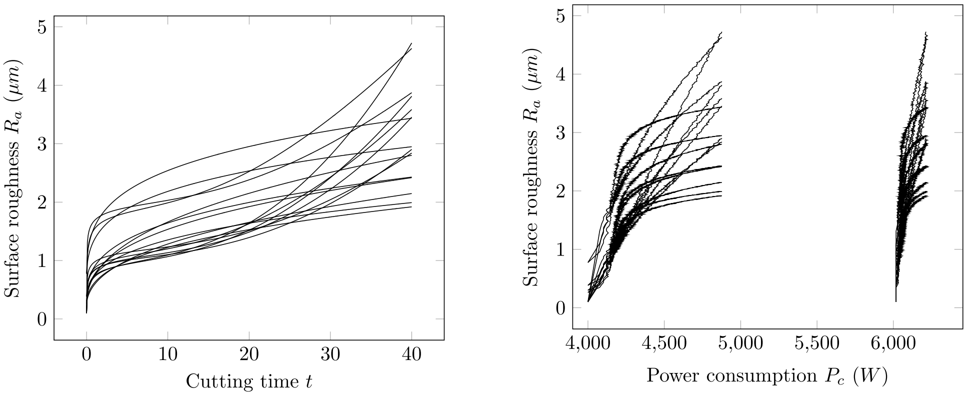
and we will use notation $\phi_n(x) = \begin{bmatrix} 1 & x & x^2 & \cdots & x^n \end{bmatrix}$ to express the



(a) Tool flank wear (V_b) dataset.

(b) Power consumption (P_c) dataset.

Fig. 3. Tool flank wear and power consumption vs. cutting time.



(a) Surface roughness (R_a) dataset.

(b) Surface roughness (R_a) dataset w.r.t Power consumption (P_c) dataset.

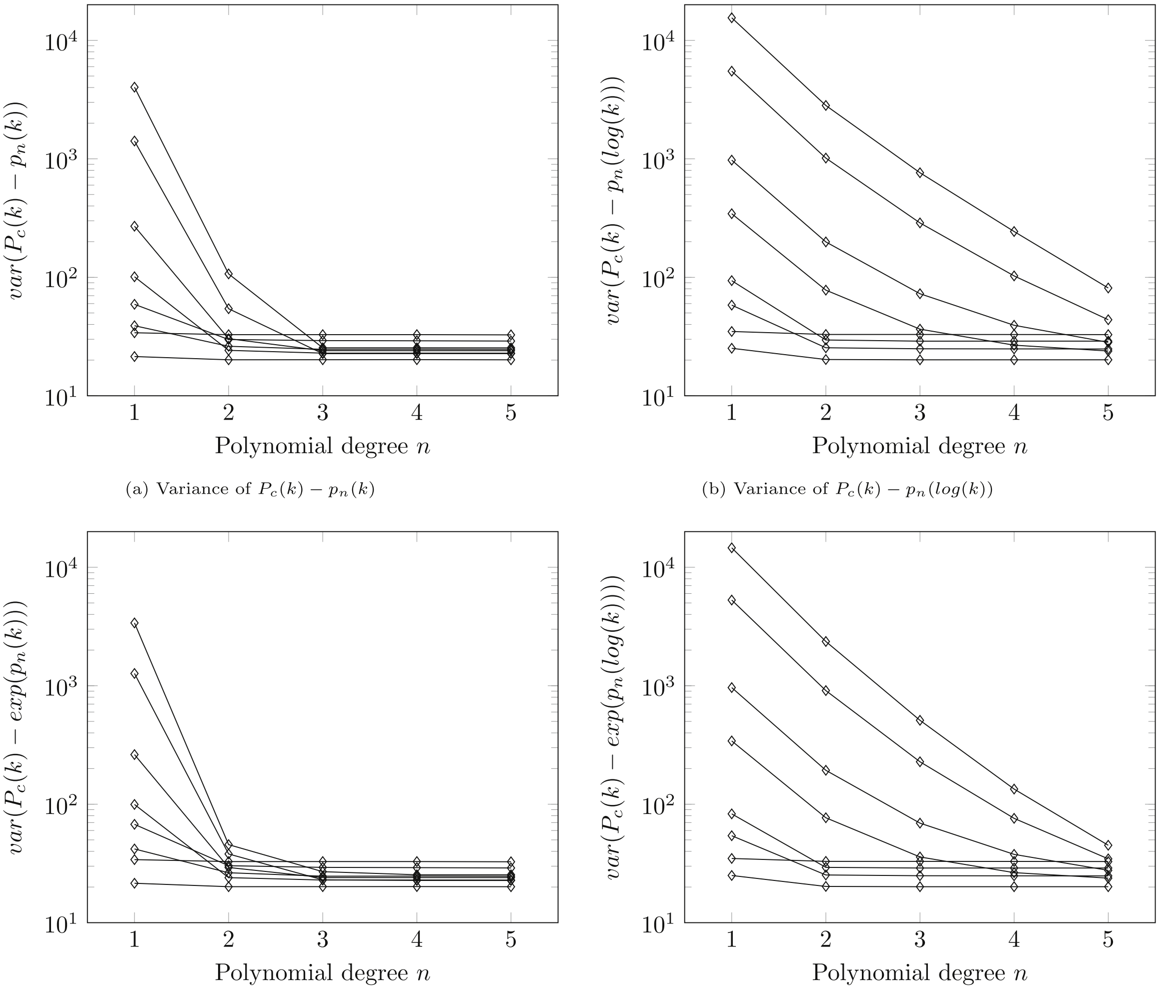
Fig. 4. Surface roughness vs. cutting time and vs. power consumption.

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.,

$$z(y) = y, \quad z(y) = log(y),$$

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 $\widehat{y} = \phi(x)\theta$ or $\widehat{y} = exp(\phi(x)\theta)$ 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 θ for both desired models using the least squares (LS) method. The obtaining of parameter vector θ has been



(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 $P_c(k) - exp(p_n(log(k)))$

Fig. 5. Validation of the $P_c(k)$ models.

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,

$$\sigma^2 = \mathrm{var}(z(y) - \phi_n(x)\theta)$$

Each model has been developed using the fittest parameter vector θ 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 (σ^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

$$\begin{aligned} &P_c(k) = p_n(k) + v^{P_c}(k); \quad P_c(k) = p_n(\log(k)) + v^{P_c}(k); \\ &log(P_c(k)) = p_n(k) + v^{P_c}(k); \quad log(P_c(k)) = p_n(\log(k)) + v^{P_c}(k); \end{aligned} \tag{9}$$

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 Fig. 5. Same equations in the dataset appear as part of the same line.

Options 5 b and d are discarded due to the general high estimation error variance. 5 a and 5 c 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 5 a is chosen before 5 c because it requires less computational costs, i.e.,

$$\begin{aligned} P_c(k) &= c_0 + c_1k + c_2k^2 + c_3k^3 + v^{P_c}(k) = p_3(k) + v^{P_c}(k) \\ &= \phi_3(k)\theta^{P_c} + v^{P_c}(k). \end{aligned} \tag{10}$$

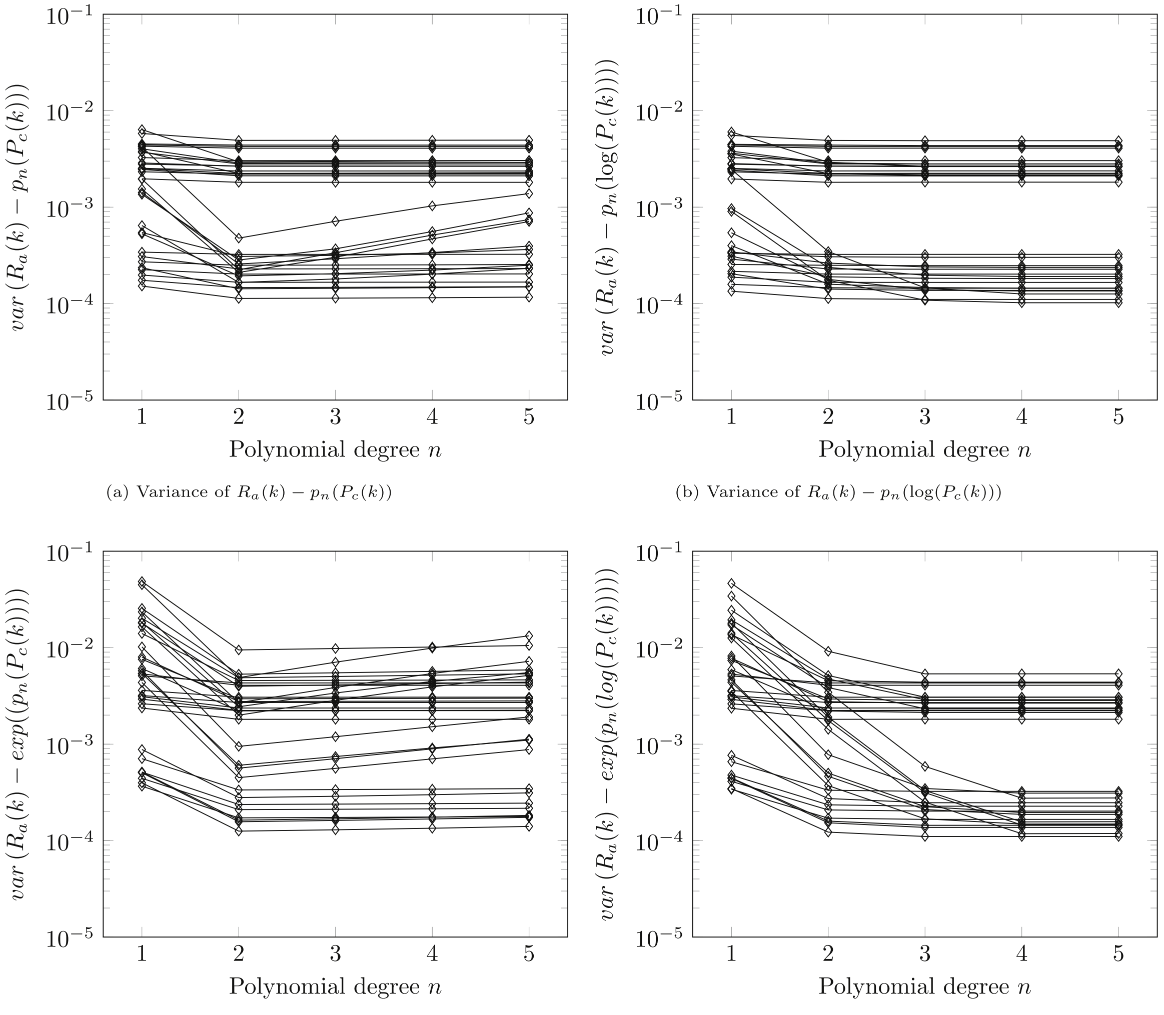
The surface roughness model is a function of the power consumption. The logarithmic modes that will be compared are

$$\begin{aligned} &R_a(k) = p_n(P_c(k)) + v^{R_a}(k); \quad R_a(k) = p_n(log(P_c(k))) + v^{R_a}(k); \\ &log(R_a(k)) = p_n(P_c(k)) + v^{R_a}(k); \quad log(R_a(k)) = p_n(\log(P_c(k))) + v^{R_a}(k); \end{aligned} \tag{11}$$

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 Fig. 6. Same equations in the dataset appear as part of the same line.

Options 6 c and d are discarded due to the general high estimation error variance. 6 a and b 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 6 a is chosen before 6 b because it requires less computational costs, i.e.,

$$\begin{aligned} R_a(P_c(k)) &= d_0 + d_1P_c(k) + d_2P_c(k)^2 + v^{R_a}(k) = p_2(P_c(k)) + v^{R_a}(k) \\ &= \phi_2(P_c(k))\theta^{R_a} + v^{R_a}(k). \end{aligned} \tag{12}$$



(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) $R_a(k) - exp(p_n(log(P_c(k))))$

Fig. 6. Validation of the $R_a(k)$ models.

Parameters from θ have been labeled here as $d_0, d_1, \dots$ to differentiate them from their equivalents in the P_c function.

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., $\widehat{P}_c = f(k)$. 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., $\widehat{R}_a = f(\widehat{P}_c)$. 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 ϕ.

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.

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

$$\underbrace{y}_{P_c} = \underbrace{\begin{bmatrix} 1 & k & k^2 & k^3 \end{bmatrix}}_{\phi_3(k)} \underbrace{\begin{bmatrix} c_0 \\ \vdots \\ c_3 \end{bmatrix}}_{\theta^{P_c}} + v^{P_c}, \tag{13}$$

with θ^{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

$$\widehat{\theta}_N^{P_c} = (X^\top X)^{-1} X^\top Y, \tag{14}$$

where

$$X = \begin{bmatrix} \phi_3(k(1)) \\ \phi_3(k(2)) \\ \vdots \\ \phi_3(k(N)) \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} P_c(1) \\ P_c(2) \\ \vdots \\ P_c(N) \end{bmatrix}, \tag{15}$$

being $\phi(k(i)) = \begin{bmatrix} 1 & k(i) & k(i)^2 & k(i)^3 \end{bmatrix}$. The product $X^\top X$ 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

$$\Sigma_N^{P_c} = \mathcal{P}_N^{P_c} V^{P_c}, \tag{16}$$

with

$$\mathcal{P}_N^{P_c} = (X^\top X)^{-1}, \tag{17}$$

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

$$V^{P_c} = \frac{1}{N-n}\left(Y - X\widehat{\theta}_N^{P_c}\right)^\top \left(Y - X\widehat{\theta}_N^{P_c}\right).$$

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.

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

$$\widehat{P}_c(i|i-1) = \phi_3(k(i))\widehat{\theta}_{i-1}^{P_c}, \tag{18}$$

where $\widehat{\theta}_{i-1}^{P_c}$ 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:

$$\phi_3(k(i)) = \begin{bmatrix} 1 & k(i) & k(i)^2 & k(i)^3 \end{bmatrix}, \tag{19}$$

whose general structure was defined in Eq. (8), its degree and logarithmic mode were selected after testing in Section 5, Eq. (10), and appeared within the generator function for P_c in Eq. (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 $\widehat{P}_c(i|i-1)$ and the direct measurements of the consumed power during the machining process of the current part, expressed as $P_c(i)$:

$$e_i^{P_c} = P_c(i) - \underbrace{\phi_3(k(i))\widehat{\theta}_{i-1}^{P_c}}_{\widehat{P}_c(i|i-1)}. \tag{20a}$$

Then, we calculate a confidence interval in which the a priori error should be contained in stable conditions.

$$J_i^{P_c} = t_{\alpha_{P_c}} \sqrt{V^{P_c}\left(1 + \phi_3(k(i))\mathcal{P}_{i-1}^{P_c}\phi_3(k(i))^\top\right)}. \tag{20b}$$

Here, $J_i^{P_c}$ represents the confidence interval with the actual model and α_{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:

$$\lambda_i^{P_c} = \begin{cases} \lambda_H^{P_c}, & |e_i^{P_c}| < J_i^{P_c}, \\ \lambda_L^{P_c}, & |e_i^{P_c}| \geq J_i^{P_c}. \end{cases} \tag{20c}$$

We calculate the gain vector using the forgetting factor and the inverse of the information matrix:

$$L_i^{P_c} = \frac{1}{\lambda_i^{P_c} + \phi_3(k(i))\mathcal{P}_{i-1}^{P_c}\phi_3(k(i))^\top}\mathcal{P}_{i-1}^{P_c}\phi_3(k(i))^\top. \tag{20d}$$

$L_i^{P_c}$ is the gain vector, which depends on the forgetting factor $\lambda_i^{P_c}$. We update the parameter vector $\widehat{\theta}_i^{P_c}$ using the calculated gain vector and the a priori estimation error:

$$\widehat{\theta}_i^{P_c} = \widehat{\theta}_{i-1}^{P_c} + L_i^{P_c} e_i^{P_c}. \tag{20e}$$

Finally, the inverse of the information matrix $\mathcal{P}^{P_c}$ is updated with the gain vector and the forgetting factor:

$$\mathcal{P}_i^{P_c} = \frac{1}{\lambda_i^{P_c}}(I - L_i^{P_c}\phi_3(k(i)))\mathcal{P}_{i-1}^{P_c}. \tag{20f}$$

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

$$\widehat{P}_c(l|i) = \phi_3(k(l))\widehat{\theta}_i^{P_c}, \tag{21}$$

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 λ^{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 λ^{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 α^{P_c} reduces the confidence interval width, i.e., $t_{\alpha^{P_c}}$, and assigning λ^{P_c} to $\lambda_L^{P_c}$ (the lower value) occurs more frequently, which causes big changes on $\widehat{\theta}^{P_c}$. Contrarily, a high value of α_{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.

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 $\widehat{P}_c(i|i)$ 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

$$y = R_a, \quad x = \widehat{P}_c(i|i), \quad \phi(x) = \phi_2(\widehat{P}_c(i|i)).$$

Therefore, we write the generation function for surface roughness as

$$\underbrace{y}_{R_a} = \underbrace{\begin{bmatrix} 1 & \widehat{P}_c(i|i) & \widehat{P}_c(i|i)^2 \end{bmatrix}}_{\phi_2(\widehat{P}_c(i|i))} \underbrace{\begin{bmatrix} d_0 \\ d_1 \\ d_2 \end{bmatrix}}_{\theta^{R_a}} + v^{R_a}. \tag{22}$$

From now on, we use the compact notation $\widehat{P}_c(i)$ to denote $\widehat{P}_c(i|i)$. 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 on 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

$$\widehat{\theta}_N^{R_a} = (X^\top X)^{-1}X^\top Y, \tag{23}$$

where

$$X = \begin{bmatrix} \phi_2\left(\widehat{P}_c\left(i_1\right)\right) \\ \phi_2\left(\widehat{P}_c\left(i_2\right)\right) \\ \vdots \\ \phi_2\left(\widehat{P}_c\left(i_N\right)\right) \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} R_a\ (i_1) \\ R_a\ (i_2) \\ \vdots \\ R_a\ (i_N) \end{bmatrix}, \tag{24}$$

being $\phi_2(\widehat{P}_c(i_j)) = \begin{bmatrix} 1 & \widehat{P}_c(i_j) & \widehat{P}_c(i_j)^2 \end{bmatrix}$. Note that $R_a(i_j)$ refers to the jth measurement of the roughness, not to the jth processed part. The product $X^\top X$ 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 $\widehat{P}_c$ 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

$$\Sigma_N^{R_a} = \mathcal{P}_N^{R_a}V^{R_a}, \tag{25}$$

with

$$\mathcal{P}_N^{R_a} = (X^\top X)^{-1}, \tag{26}$$

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.

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

$$\widehat{R}_a(i_j|i_{j-1}) = \phi_2(\widehat{P}_c(i_j))\widehat{\theta}_{j-1}^{R_a}. \tag{27}$$

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, $\widehat{\theta}_{j-1}^{R_a}$ contains the parameters that were estimated in the last time this algorithm was activated, and $\phi_2(\widehat{P}_c(i_j))$ is the regression vector, which uses the values of $\widehat{P}_c$ that were calculated using the previous algorithm in Section 6.2. As a reminder, this regression vector takes this form:

$$\phi_2(\widehat{P}_c(i_j)) = \left[1 \quad \widehat{P}_c(i_j) \quad \widehat{P}_c(i_j)^2\right], \tag{28}$$

whose general structure was defined in Eq. (8), its degree and logarithmic mode were selected after testing with the datasets in Section 5, Eq. (12) and appeared within the generator function for R_a in Eq. (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 $\widehat{R}_a(i_j|i_{j-1})$ and the direct measurements of the surface roughness during the instant i_j (i.e., at the jth surface roughness measurement, which takes place on the ith processed part), expressed as $R_a(i_j)$:

$$e_j^{R_a} = R_a(i_j) - \underbrace{\phi_2(\widehat{P}_c(i_j))\widehat{\theta}_{j-1}^{R_a}}_{\widehat{R}_a(i_j|i_{j-1})}. \tag{29a}$$

We calculate a confidence interval where the a priori error should remain in stable conditions.

$$J_j^{R_a} = t_{\alpha_{R_a}}\sqrt{V^{R_a}(1 + \phi_2(\widehat{P}_c(i_j))\mathcal{P}_{j-1}^{R_a}\phi_2(\widehat{P}_c(i_j))^\top)}. \tag{29b}$$

Here, $J_j^{R_a}$ represents the confidence interval with the actual model and α_{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 λ^{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:

$$\lambda_j^{R_a} = \begin{cases} \lambda_H^{R_a}, & |e_j^{R_a}| < J_j^{R_a}, \\ \lambda_L^{R_a}, & |e_j^{R_a}| \geq J_j^{R_a}. \end{cases} \tag{29c}$$

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 $\lambda_j^{R_a}$:

$$L_j^{R_a} = \frac{1}{\lambda_j^{R_a} + \phi_2(\widehat{P}_c(i_j))\mathcal{P}_{j-1}^{R_a}\phi_2(\widehat{P}_c(i_j))^\top.}\mathcal{P}_{i-1}^{R_a}\phi_2(\widehat{P}_c(i_j))^\top \tag{29d}$$

Then, we update the parameter vector $\widehat{\theta}^{R_a}$ using the calculated gain vector and the a priori estimation error:

$$\widehat{\theta}_j^{R_a} = \widehat{\theta}_{j-1}^{R_a} + L_j^{R_a}e_j^{R_a}. \tag{29e}$$

Finally, the inverse of the information matrix is updated with the gain vector and the forgetting factor:

$$\mathcal{P}_j^{R_a} = \frac{1}{\lambda_j^{R_a}}(I - L_j^{R_a}\phi_2(\widehat{P}_c(i_j)))\mathcal{P}_{j-1}^{R_a}. \tag{29f}$$

We use the following expression at any instant i to estimate a future value of the surface roughness at instant l

$$\widehat{R}_a(l|i) = \phi(\widehat{P}_c(l|i))\widehat{\theta}_j^{R_a} = \phi_2\left(\phi_3(k(l))\widehat{\theta}_i^{P_c}\right)\widehat{\theta}_j^{R_a}, \tag{30}$$

where i represents the instant of time for the most updated model for power predictions (i.e., at the ith processed part), and $\widehat{\theta}_j^{R_a}$ is the value of $\widehat{\theta}^{R_a}$ that was calculated in the last instant i_j. The values of λ^{R_a} and α_{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.

Fig. 7 summarizes the internal steps of the algorithm.

7. Simulation results

In this section, we will validate the performance of the proposed approaches from Section 3 and will compare them to the direct approaches from Section 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 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.

7.1. 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 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 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 $\text{var}(\text{meas}_{\text{noise}}) = (m/3)^2$, where m is the uncertainty of each instrument, as shown in Table 5. Measurement frequency of surface roughness is also included in the aforementioned table.

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 of 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 6.

Firstly, the accuracy of the updating algorithm is checked. In this case, the selected approach does not affect the updating performance. Fig. 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. Fig. 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.

The following step is the validation of the internal stability of vector parameters θ in order to ensure appropriate predictions. Fig. 9a and b shows the evolution of θ 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.

7.3. Performance indexes and settings

The performance of the approaches from Section 2.3 and 3 is evaluated with a simulated experiment. In these simulations, which use all three cases from Section 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**

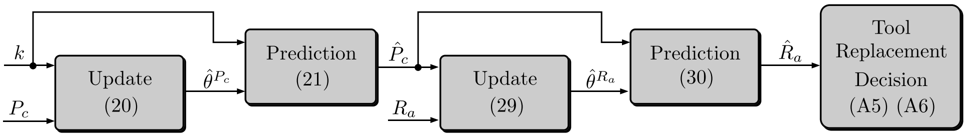


Fig. 7. Operating diagram of the ARLS algorithm.

Table 4
Multiplying factors (benchmark).

	Tool ≤ 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

Table 5
Benchmark parameters.

Name	Value	Units	Definition
m^{P_c}	±300	W	P_c uncertainty noise
m^{R_a}	±0.2	µm	R_a uncertainty noise
freq	20	(parts/meas.)	R_a measurement frequency

Table 6
Simulation parameters.

Name	Value	Units	Definition
$R_{a,lim}$	3.6	µm	R_a threshold
$P_{c,lim}$	7400	W	P_c threshold (**A3**)
$\lambda_L^{P_c}$	0.7	–	Lower λ^{P_c}
$\lambda_H^{P_c}$	1	–	Higher λ^{P_c}
$\lambda_L^{R_a}$	0.8	–	Lower λ^{R_a}
$\lambda_H^{R_a}$	1	–	Higher λ^{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 $t_{\alpha_{R_a}}$

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 6. Approaches **A1**, **A4**, **A5** and **A6** require the surface roughness limit $R_{a,\text{lim}}$. 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 λ and t_α.

7.4. Discussion

The results of the simulations are shown in Table 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**.

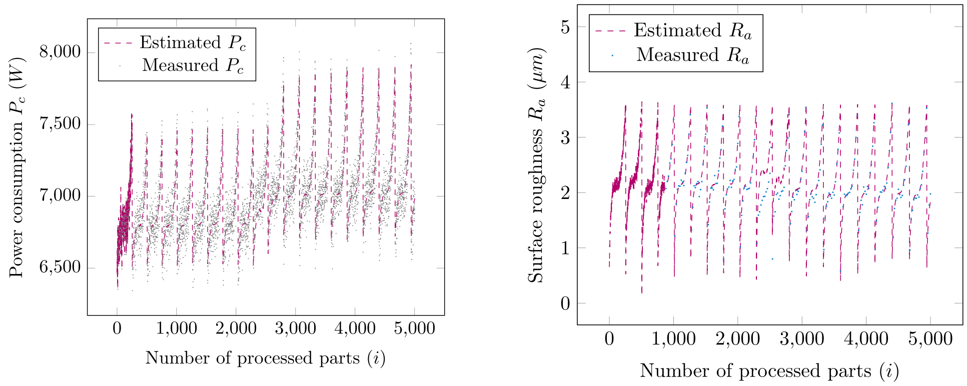
Approach **A3** performs well in **Case 1**, which is stable, but uses a high amount of tools in **Case 2** and processes an excessive amount of parts out of specifications in **Case 3**. This is due to the fact that **A3** does not react to those internal changes. Approach **A2** behaves in a similar 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 ($\lambda_H^{P_c}, \lambda_L^{P_c}, \lambda_H^{R_a}, \lambda_L^{R_a}, \ldots$). 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 6.

8. Conclusions

In this paper, 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.

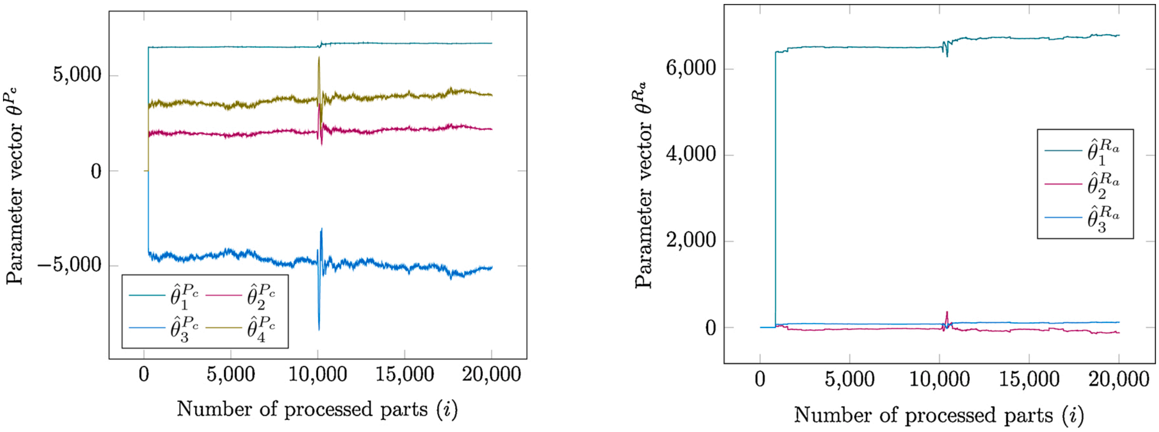
The idea behind these tool replacement strategies is to assure that the processed parts fulfill certain quality criteria based on surface roughness



(a) Estimating power consumption (P_c) with the ARLS algorithm.

(b) Estimating surface roughness (R_a) with the ARLS algorithm.

Fig. 8. Validation of the updating algorithm accuracy.



(a) Stability of parameter vector θ^{P_c}.

(b) Stability of parameter vector θ^{R_a}.

Fig. 9. Stability of parameter vectors.

Table 7
Simulation results.

Approach	Case 1			Case 2			Case 3		
	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

thresholds. The tool replacement strategies we have presented in this paper 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 theoretical 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.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgements

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

Conflict of interest: None declared.

References

[1] Kurada S, Bradley C. A review of machine vision sensors for tool condition monitoring. Comput Ind 1997;34(1):55–72.
[2] Davim JP. Sustainable machining. Springer; 2017.
[3] Jain AK, Lad BK. A novel integrated tool condition monitoring system. J Intell Manuf 2019;30(3):1423–36.
[4] Stephenson DA, Agapiou JS. Metal cutting theory and practice. CRC Press; 2016.
[5] Mohanraj T, Shankar S, Rajasekar R, Sakthivel N, Pramanik A. Tool condition monitoring techniques in milling process – a review. J Mater Res Technol 2020;9(1):1032–42.
[6] Liu C, Ni J, Wan P. An accurate prediction method of multiple deterioration forms of tool based on multitask learning with low rank tensor constraint. J Manuf Syst 2021;58:193–204.
[7] Wang J, Li Y, Zhao R, Gao RX. Physics guided neural network for machining tool wear prediction. J Manuf Syst 2020;57:298–310.
[8] Gao R, Wang L, Teti R, Dornfeld D, Kumara S, Mori M, et al. Cloud-enabled prognosis for manufacturing. CIRP Ann 2015;64(2):749–72.
[9] Taylor FW. On the art of cutting metals…, vol. 23. American Society of Mechanical Engineers; 1906.
[10] Woldman NE, Gibbons RC. Machinability and machining of metals. McGraw-Hill; 1951.
[11] Hu Y, Liu S, Lu H, Zhang H. Remaining useful life model and assessment of mechanical products: a brief review and a note on the state space model method. Chin J Mech Eng 2019;32(1):15.

[12] Si X-S, Wang W, Hu C-H, Zhou D-H. Remaining useful life estimation – a review on the statistical data driven approaches. Eur J Oper Res 2011;213(1):1–14.

[13] Li H, Wang Y, Zhao P, Zhang X, Zhou P. Cutting tool operational reliability prediction based on acoustic emission and logistic regression model. J Intell Manuf 2015;26(5):923–31.

[14] Drouillet C, Karandikar J, Nath C, Journeaux A-C, El Mansori M, Kurfess T. Tool life predictions in milling using spindle power with the neural network technique. J Manuf Process 2016;22:161–8.

[15] Benkedjouh T, Medjaher K, Zerhouni N, Rechak S. Health assessment and life prediction of cutting tools based on support vector regression. J Intell Manuf 2015; 26(2):213–23.

[16] Wu J, Su Y, Cheng Y, Shao X, Deng C, Liu C. Multi-sensor information fusion for remaining useful life prediction of machining tools by adaptive network based fuzzy inference system. Appl Soft Comput 2018;68:13–23.

[17] Li X, Lim B, Zhou J, Huang S, Phua S, Shaw K, et al. Fuzzy neural network modelling for tool wear estimation in dry milling operation. Annual conference of the prognostics and health management society 2009:1–11.

[18] Huang C-G, Yin X, Huang H-Z, Li Y-F. An enhanced deep learning-based fusion prognostic method for RUL prediction. IEEE Trans Reliab 2019;69(3):1097–109.

[19] Li H, Wang W, Li Z, Dong L, Li Q. A novel approach for predicting tool remaining useful life using limited data. Mech Syst Signal Process 2020;143:106832.

[20] Wu J-Y, Wu M, Chen Z, Li X, Yan R. A joint classification-regression method for multi-stage remaining useful life prediction. J Manuf Syst 2021;58:109–19.

[21] Hanachi H, Yu W, Kim IY, Liu J, Mechefske CK. Hybrid data-driven physics-based model fusion framework for tool wear prediction. Int J Adv Manuf Technol 2019; 101(9–12):2861–72.

[22] Wang J, Wang P, Gao RX. Enhanced particle filter for tool wear prediction. J Manuf Syst 2015;36:35–45.

[23] Zhang J, Starly B, Cai Y, Cohen PH, Lee Y-S. Particle learning in online tool wear diagnosis and prognosis. J Manuf Process 2017;28:457–63.

[24] Baruah P, Chinnam RB. HMMs for diagnostics and prognostics in machining processes. Int J Prod Res 2005;43(6):1275–93.

[25] Kumar A, Chinnam RB, Tseng F. An HMM and polynomial regression based approach for remaining useful life and health state estimation of cutting tools. Comput Ind Eng 2019;128:1008–14.

[26] Tiwari K, Shaik A, Arunachalam N. Tool wear prediction in end milling of Ti-6Al-4V through Kalman filter based fusion of texture features and cutting forces. Proc Manuf 2018;26:1459–70.

[27] Wang P, Gao RX. Adaptive resampling-based particle filtering for tool life prediction. J Manuf Syst 2015;37:528–34.

[28] Wang J, Zheng Y, Wang P, Gao RX. A virtual sensing based augmented particle filter for tool condition prognosis. J Manuf Process 2017;28:472–8.

[29] Sun H, Cao D, Zhao Z, Kang X. A hybrid approach to cutting tool remaining useful life prediction based on the Wiener process. IEEE Trans Reliab 2018;67(3): 1294–303.

[30] An H, Wang G, Dong Y, Yang K, Sang L. Tool life prediction based on Gauss importance resampling particle filter. Int J Adv Manuf Technol 2019;103(9): 4627–34.

[31] Liu R, Kothuru A, Zhang S. Calibration-based tool condition monitoring for repetitive machining operations. J Manuf Syst 2020;54:285–93.

[32] Niaki FA, Ulutan D, Mears L. In-process tool flank wear estimation in machining gamma-prime strengthened alloys using Kalman filter. Proc Manuf 2015;1: 696–707.

[33] Paris PC. A rational analytic theory of fatigue. Trend Eng 1961;13:9.

[34] Wang Y, Zheng L, Wang Y. Event-driven tool condition monitoring methodology considering tool life prediction based on industrial internet. J Manuf Syst 2021;58: 205–22.

[35] Standard I. 8688-2, Tool life testing in milling – part 2. 1989.

[36] Shao H, Wang H, Zhao X. A cutting power model for tool wear monitoring in milling. Int J Mach Tools Manuf 2004;44(14):1503–9.

[37] Kovac P, Rodic D, Pucovsky V, Savkovic B, Gostimirovic M. Application of fuzzy logic and regression analysis for modeling surface roughness in face milling. J Intell Manuf 2013;24(4):755–62.

[38] Yen Y-C, Söhner J, Lilly B, Altan T. Estimation of tool wear in orthogonal cutting using the finite element analysis. J Mater Process Technol 2004;146(1):82–91.

[39] Pálmai Z. Proposal for a new theoretical model of the cutting tool's flank wear. Wear 2013;303(1–2):437–45.

[40] Zhu K, Zhang Y. A generic tool wear model and its application to force modeling and wear monitoring in high speed milling. Mech Syst Signal Process 2019;115: 147–61.

[41] Takeyama H, Murata R. Basic investigation of tool wear. J Eng Ind 1963;85(1): 33–7.

[42] Malakizadi A, Gruber H, Sadik I, Nyborg L. An FEM-based approach for tool wear estimation in machining. Wear 2016;368:10–24.

[43] Cuppini D, D`errico G, Rutelli G. Tool wear monitoring based on cutting power measurement. Wear 1990;139(2):303–11.

[44] Yoon H-S, Lee J-Y, Kim M-S, Ahn S-H. Empirical power-consumption model for material removal in three-axis milling. J Clean Prod 2014;78:54–62.

[45] Özel T, Karpat Y, Figueira L, Davim JP. Modelling of surface finish and tool flank wear in turning of AISI D2 steel with ceramic wiper inserts. J Mater Process Technol 2007;189(1–3):192–8.

[46] Lima J, Avila R, Abrao A, Faustino M, Davim JP. Hard turning: AISI 4340 high strength low alloy steel and AISI d2 cold work tool steel. J Mater Process Technol 2005;169(3):388–95.

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

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

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