

## Article

# Monitoring of Tool and Component Wear for Self-Adaptive Digital Twins: A Multi-Stage Approach through Anomaly Detection and Wear Cycle Analysis

Robin Ströbel <sup>1</sup>, Alexander Bott <sup>1,\*</sup>, Andreas Wortmann <sup>2</sup> and Jürgen Fleischer <sup>1</sup>

<sup>1</sup> wbk Institute of Production Science, Karlsruhe Institute of Technology (KIT), Kaiserstraße 12, 76131 Karlsruhe, Germany; robin.stroebel@kit.edu (R.S.); juergen.fleischer@kit.edu (J.F.)

<sup>2</sup> Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW), University of Stuttgart, Seidenstrasse 36, 70174 Stuttgart, Germany; andreas.wortmann@isw.uni-stuttgart.de

\* Correspondence: alexander.bott@kit.edu; Tel.: +49-1523-950-2643

**Abstract:** In today's manufacturing landscape, Digital Twins play a pivotal role in optimising processes and deriving actionable insights that extend beyond on-site calculations. These dynamic representations of systems demand real-time data on the actual state of machinery, rather than static images depicting idealized configurations. This paper presents a novel approach for monitoring tool and component wear in CNC milling machines by segmenting and classifying individual machining cycles. The method assumes recurring sequences, even with a batch size of 1, and considers a progressive increase in tool wear between cycles. The algorithms effectively segment and classify cycles based on path length, spindle speed and cycle duration. The tool condition index for each cycle is determined by considering all axis signals, with upper and lower thresholds established for quantifying tool conditions. The same approach is adapted to predict component wear progression in machine tools, ensuring robust condition determination. A percentage-based component state description is achieved by comparing it to the corresponding Tool Condition Codes (TCC) range. This method provides a four-class estimation of the component state. The approach has demonstrated robustness in various validation cases.

**Keywords:** tool wear; component wear; digital twin; machine tools



**Citation:** Ströbel, R.; Bott, A.; Wortmann, A.; Fleischer, J. Monitoring of Tool and Component Wear for Self-Adaptive Digital Twins: A Multi-Stage Approach through Anomaly Detection and Wear Cycle Analysis. *Machines* **2023**, *11*, 1032. <https://doi.org/10.3390/machines11111032>

Academic Editor: Mark J. Jackson

Received: 30 October 2023

Revised: 13 November 2023

Accepted: 17 November 2023

Published: 19 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Machine tools are widely used in modern manufacturing industries to produce high-quality products with increased productivity and efficiency [1]. However, every machine tool is unique due to its individual characteristics (dynamics and tolerances) [2]. This individuality results from a complex interplay of different machine components, which may cause variations in the machine tool's performance over time. The increasing trend towards individualism of products leads to smaller batch sizes, which requires production processes that can adapt quickly and efficiently to changes in product requirements [3]. Digital Twins are being developed to meet these challenges, enabling real-time process optimisation and predictive maintenance. However, this requires accurate and up-to-date data. By monitoring the wear and condition of components and tools, the Digital Twin can predict potential failures and reduce downtime, thereby increasing productivity [4]. However, the wear and condition of machine tools are influenced by many factors, including the operating environment, cutting parameters and tool condition. This information must be transferred from the physical system to the Digital Twin environment. Data transfer can be a complex and challenging process that requires careful consideration because of various factors such as the type of data, the frequency of updates and the required level of accuracy [5]. This complexity demands a robust and adaptable condition monitoring approach to capture and process the machine's real-time status accurately.

In addressing the challenge mentioned, this paper introduces an innovative cycle-based two-stage tool and component condition index tailored for self-adapting Digital Twins of machine tools. The methodology uses wear cycles to identify anomalies in common machine signals, enabling the detection of wear conditions in tools and components. These calculations are executed on machine Industrial Personal Computers (IPCs) and transmitted as a condensed process wear matrix to a remote Digital Twin. To substantiate the efficacy of our approach, we conduct six experiments and a long-term wear test on a ball screw drive. The results demonstrate the robustness and effectiveness of our approach in detecting wear in cutting tools and wear mechanisms in ball screw drives. This innovative framework forms the basis for providing real-time insights into the machine's current capability, fostering adaptability and enhancing the decision-making capabilities of the Digital Twin.

## 2. State of the Art and Related Work

Process segmentation of machine tool data is essential for quality control and process improvement to differentiate varying operation states [6]. These varying operation states are characterised by sub-sequences which can result in variations in the same time series. This can influence the data processing or possible predictions. Because of this, the segmentation process divides the time series into sub-sequences to maximise comparability within a group [7]. Here, two cases have to be differentiated: online and offline pattern recognition. In offline pattern recognition, the whole time series already exists and is known to the algorithm. In online pattern recognition, the analysed time series is not fully known. Because of this, in the instance of recognition, not all possible sub-sequences and data points are known for comparison [8].

Process segmentation is needed in the first stage to differentiate a number of features that characterise the time series to recognise the starting point of new sub-sequences. In the next stage, the endpoint of the sub-sequence and the start point of the next sub-sequence, respectively, have to be recognised. The last step is the assignment of a group with maximum comparability. The change in the system behaviour which likely characterises a different operation state can be either a trigger or an anomaly [9,10]. Anomaly detection in time series is a well-researched area, with various methods being employed, and they can be classified into different categories. Ref. [6] presents a comprehensive overview of these methods, which include pattern matching, clustering and predictions, and distance-based, and statistical and probabilistic methods.

In their study, ref. [11] developed an unsupervised anomaly detection system for industrial control loops using an extreme learning machine classifier. They achieved this by mapping the data to a two-dimensional space and then setting limits in that space. To do this, they applied several approaches, including principal component analysis, beta Hebbian learning and curvilinear component analysis. Of these approaches, beta Hebbian learning was found to perform the best. Other researchers, such as [12], have developed approaches that detect point and collective anomalies using sliding windows and autoencoders, respectively. Predictive maintenance approaches also focus on feature extraction using methods like FRESH [13]. These features can be used as high-quality inputs to supervised Machine Learning models to achieve high accuracy. However, labelled training data is needed for supervised learning, which may not always be available. Ref. [14] presented an approach based on time series segmentation followed by anomaly detection, which detects anomalies using a combination of a Recurrent Neural Network for feature extraction and a Convolutional Neural Network-based autoencoder. However, this approach does not provide a clear division into actual processes, making the clear assignment of anomalies to subprocesses potentially difficult.

In conclusion, process segmentation of machine tool data is an important step for process analysis and comparability of operation states. The number of produced parts can influence the choice of algorithm for process segmentation, especially because most Machine Learning-based methods need a large training dataset for accurate estimations. Especially for small production runs, statistical process control methods may be more

appropriate due to the limited amount of data available [10]. Here, approaches that work without fixed thresholds to adapt themselves for autonomous usage can enable a broader usage. This motivates further research into the combination of statistical methods with self-adapting thresholds for an optimised process segmentation.

### 2.1. Tool Wear Detection

During machining, the cutting tool experiences different wear mechanisms, which lead to reduced tool life, poor surface finish and increased tool replacement costs. Therefore, the development of tool wear detection systems has become a focus point of the machining process to optimise tool usage and improve manufacturing efficiency [15]. Different types of wear mechanisms can occur in machining, including abrasive, adhesion, flank and crater wear. Among these, flank and crater wear are the most relevant wear mechanisms [16]. Flank wear occurs on the flank face of the cutting tool due to rubbing against the workpiece, while crater wear results from the deformation of the tool surface due to thermal and mechanical stresses during machining. Flank wear can cause an increase in cutting force and chatter vibrations, leading to poor surface finish and dimensional accuracy [17]. Crater wear can cause a reduction in tool strength and stability, leading to tool breakage [16]. Different sensor principles can be used to detect these wear mechanisms.

Ref. [18] used acoustic emission (AE) sensors in combination with a Machine Learning approach to detect the flank wear during machining processes in relation to the cutting speed and cutting time. The achieved quality in the prediction of the tool wear highly correlates with the usage of a feature for the cutting time. Through this feature and additional information on the cutting speed, a reduction in the mean classification error, too, in the worst case of 11.19%, could be achieved [18]. In opposition to this feature, the preprocessing-based approach from [19] used a dynamometer to continuously measure the process force and classify tool wear based on the raw force signal. For this classification, a Convolutional Neural Network (CNN) classifies three different phases of the tool's lifetime. Here, it is differentiated between a rapid initial wear phase, a uniform wear phase and the end of the useful life of the tool. Figure 1 illustrates the tool lifetime for different cutting speeds  $v_n$ . The different phases are characterised through the different areas in the gradient.

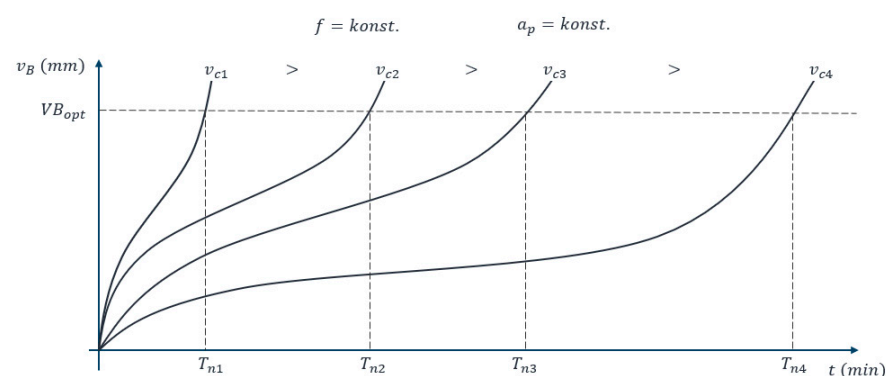


Figure 1. Lifetime curve for different cutting speeds after [20].

Ref. [19] achieved a classification accuracy of 90% for these classes. Ref. [21] uses vibration signals of the workpiece in combination with a support vector machine (SVM) as a classifier. The differentiated classes were also the same three phases as in [18]. Ref. [21] focused in their research on achievable accuracy through specific combinations of dimension reduction and varying kernels for the SVM. Through this, they could achieve, in the optimal case for their experiments, an accuracy of up to 96.13%. Ref. [22] uses a Deep Learning approach combined with images of different tools to classify in the first stage, the visible tool, and in the next step the degree of wear. The classification of the tool type [22] could achieve an accuracy of 95.6%. The results were 73% for the tool wear, but this could only be achieved on the test data. Unknown and slightly disturbed data could only achieve

a coefficient of 0.37%. Ref. [23] also uses an indirect approach for the detection of tool wear through the spindle power. Through the wear progress on the tool, the necessary power for a constant milling process increases. They acquire the power through an additional sensor which in the next step will be processed through a neuronal network for a curve fit to estimate the Remaining Useful Life (RUL). This enables prediction with a mean estimation error of <2 min for RUL.

The current approaches for tool wear detection have some limitations. AE sensors are non-invasive and can detect tool wear in real time. However, they can be affected by environmental noise and may require complex signal processing because of high sample rates [24]. Force sensors can provide accurate and reliable cutting force measurements, but they are costly and require contact with the workpiece, limiting their application to small sizes. Vibration sensors can detect wear in real time and are non-invasive. But they can be affected like AE sensors through external vibrations and also require complex signal processing. Optical sensors, in contrast, are non-contact and can provide accurate measurements of tool deformation, but they require a clear view of the tool surface and may be affected by ambient light [22]. Additionally, these sensors require frequent calibration and can be affected by tool geometry changes, which can lead to necessary retraining or possible modification steps in signal processing. The data processing methods depend on the application and machining process. Feature extraction and selection can reduce the dimensionality of the data and improve the accuracy of the wear detection system. However, they may require prior knowledge of the wear mechanism and the machining process [18]. Machine Learning algorithms can provide accurate and reliable wear detection, but they require a large amount of labelled data and may require frequent updates as the machining process changes.

In conclusion, tool wear detection is a critical component of the machining process to optimise tool usage and improve manufacturing efficiency. However, the addition of specialised sensors has downsides as an additional cost, and, more importantly, the addition of a new component increases the failing potential.

## 2.2. Current-Based Component Wear Detection

Condition monitoring of drive components is important in the industry, and several approaches have been developed for detecting and characterising wear mechanisms. Two different approaches are used to monitor these different effects. A distinction is made between detection by sensors and “sensorless” detection. For the usage of machine tools, additional sensors should always be avoided, as this results in costs and failure potential. In addition, there is the necessity of corresponding experts for the integration and evaluation of results [25]. Sensorless detection in this context describes using existing information, such as the motor current, position, etc. (soft sensors), that can be extracted from the plc [25]. There are many approaches for condition monitoring of components through investigations of the motor current [26]. These range from wear detection of motors and bearings to characterisation [14,27] and system property change monitoring [28]. The different system properties can result from geometrical differences in the components or different forms of wear [14,29]. Research results in condition monitoring by the motor current of ball screw drives (BSD) will be reviewed in depth. As an essential high-precision drive component, subject to high loads and thus wear due to the drive function, ball screws are particularly relevant for condition monitoring.

The different wear mechanisms in ball screws can be classified into four tribological categories: adhesion, surface disruption, abrasion and tribochemical reaction [30]. These mechanisms cause a decrease in ball diameter, which leads to a backlash and loss of preload, affecting the stiffness of the ball screw [31]. This phenomenon is the basis of condition-monitoring approaches. Ref. [32] is researching the estimation of ball screw wear based on the motor current of the actuating servo motor and other control variables such as feedback position and motor torque. For this purpose, ref. [32] uses quasi-sweep sine wave motions of varying. By relating the change in position and motor torque amplitudes the degree

of wear is estimated. However, this approach has optimisation potential due to a high estimator error of up to 27%. Ref. [33] also deals with the prediction of the wear of a BSD. The approach also addresses the prediction of wear based on the motor current but without any additional reference runs. This is achieved with additional compensation current. Identifying the rapid traverse in the first step and determining the resulting current from the three phases in the next step enables the evaluation and statistical characterisation of the signals. By progressively increasing the ratio of the specific band energy, ref. [33] successfully classify and characterise a normal lifetime phase, an accelerated wear phase and a strong wear phase. However, this requires the use of external, more expensive measurement systems. External sensors are also used by [34] to obtain the motor current signal. The aim of their work is fault diagnosis of BSD in industrial robots based on the motor current alone. For this purpose, a data processing pipeline using a short-time Fourier transform and wavelet decomposition is proposed. Statistical parameters are extracted and selected for further processing within the pipeline based on an evaluation measure. Using logistic regression, the wear behaviour is classified, thus enabling the diagnosis of faulty BSDs. Through this, they do not require any special reference runs. The disadvantage is that only defective conditions can be detected and not predicted. Ref. [28] deals with detecting preload losses, which can be traced back to wear. Here, too, external sensors are used, which, in contrast to [33], can be easily retrofitted. These are Hall sensors and acceleration sensors to detect changes in the natural frequency response of a dynamic feed axis model. The change in the natural frequency response can be used to infer the change in stiffness and, ultimately, the loss of preload. For this purpose [28], establish a dynamic model of the feed axis. Using the test bench, the natural frequencies of the feed axis are detected through defined motion sequences. The natural frequencies are validated using other types of excitation. This enables the preload to be classified as low, normal, or overly high. Thus, the positioning error of the carriage can be traced back to the change in the natural frequency.

In conclusion, component wear detection is a critical factor in predicting downtimes or changes in the kinematic characteristics of the drive train. However, the addition of specialised sensors for wear detection in ball screw drives has the same downside as tool wear monitoring.

### 2.3. Self-Adapting Digital Twins

Digital Twins are virtual replicas of physical systems that can be used to simulate, monitor and control their real-world counterparts. They are typically built by combining data from sensors, models and other sources to represent the physical system digitally [35]. One potential downside of Digital Twins is that they are typically designed with a fixed set of parameters and configurations that may not be optimal in all situations [36]. This is where self-adapting Digital Twins come in. Self-adapting Digital Twins, also known as adaptive Digital Twins or self-optimising Digital Twins, are designed to adjust their parameters and configurations in response to changes in their environments or objectives [37]. Using Machine Learning algorithms and optimisation techniques, self-adapting Digital Twins can analyse data from sensors and other sources in real time and predict the system's behaviour [38]. Without human intervention, they can learn from their experiences and optimise their performance over time [37]. This ability to adapt to changing conditions and objectives makes self-adapting Digital Twins a powerful tool for improving physical systems' efficiency, reliability and safety. Several frameworks exist to implement self-adapting Digital Twins, including model-based and data-driven approaches. Model-based frameworks use mathematical models to simulate the behaviour of physical systems and optimise their performance, while data-driven frameworks use Machine Learning algorithms to analyse data from sensors and other sources and make predictions about system behaviour [5]. Realising self-adapting Digital Twins requires a combination of hardware and software components, including sensors and actuators for collecting data and controlling physical systems, and Machine Learning algorithms and optimisation

techniques for analysing and adapting to changes in data and objectives. Advances in sensor technology, Machine Learning algorithms and Cloud Computing infrastructure have all contributed to the development of self-adapting Digital Twins [5]. Self-adapting Digital Twins can realise several positive effects, including improved system performance, reduced maintenance costs and increased operational efficiency. By adapting to changes in their environments and objectives, self-adapting Digital Twins can help ensure physical systems operate at peak efficiency and minimise the risk of failures or downtime.

#### 2.4. Data Transfer for Digital Twins

The direct data dependency shows the necessity of considering the data flow between the sensor and Digital Twin. Here, two different structures of data flow exist in state of the art [39]:

- (a) Direct data flow from the data source into the Digital Twin;
- (b) Indirect data flow from the data source through a processing step into the Digital Twin.

Due to substantial differences in the existing infrastructure, there are significant differences in the possible data rate between the data source and Digital Twin. Ref. [39] distinguishes between volatile and non-volatile data. Volatile data is data with a higher frequency that results, for example, from axis movement or material flow. Non-volatile data describes data that presents relatively static information, such as a product or machine list. The limited processing capacities of a Digital Twin require local (pre)processing for highly volatile data [5]. The pre-processing function is fulfilled by Edge Computing, which sits directly at the application through an Edge Device (ED). Edge Computing pursues the goal of processing data and executing applications closer to the source of the data, i.e., to the end devices and sensors, to avoid latencies or to realise time-critical processing in the first place. For this purpose, computing and storage resources are provided as micro data centres at the network's edges [40]. The disadvantage of an individualised and decentralised computing architecture using Edge Computing is, on the one hand, the efficiency and, on the other hand, the resource commitment [41]. Centralised instances or defined data centres enable a considerably more efficient operation of the corresponding computing resources [42]. Additional soft factors, such as the shortage of skilled personnel, continue to make the local operation of computing resources or an individualised processing structure more difficult [43]. On the other hand, there are centralised networks in the form of cloud infrastructure. This represents a global network in which IT services of varying complexity and for different user segments can be provided [42]. Infrastructure and platform providers offer basic services and the necessary application infrastructure, while application providers enable flexible and easy-to-maintain application operations. This enables low capital lockup, reduced complexity of IT operations and the ability to respond quickly and flexibly to changing capacity needs [44].

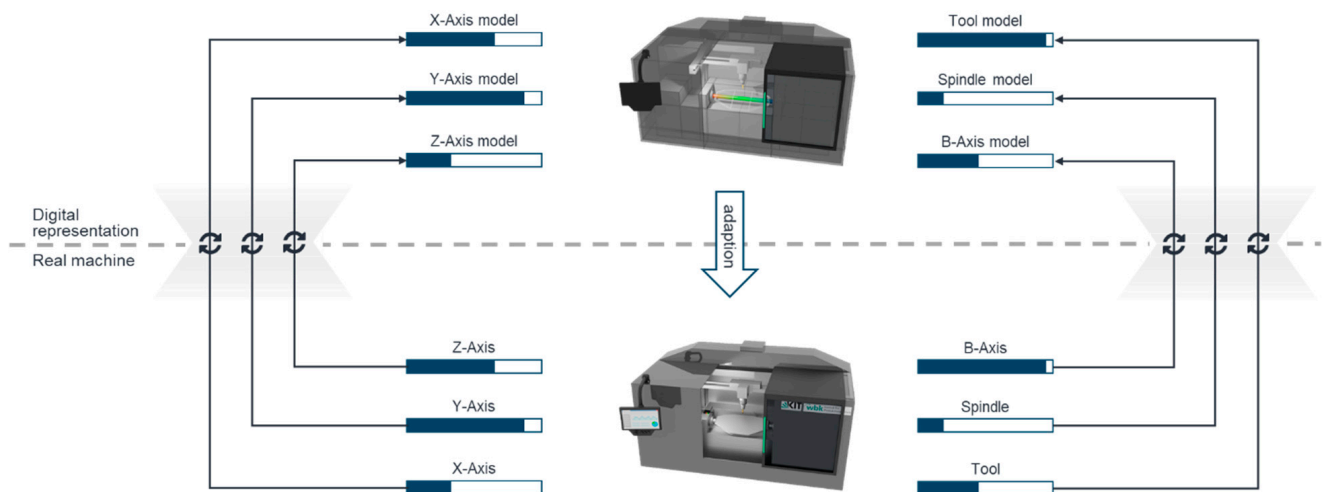
Digital Twins can be managed centrally in the company or across companies using a cloud infrastructure structure, for example, or integrated into an Edge Device like the pre-processing [44]. Ref. [45] argue that the centralisation of different Digital Twins of the same equipment opens the possibility to exchange information among each other and thus achieve optimisation of used models. Ref. [41] shows that a combination of edge and cloud computing can ideally exploit computational and transfer capacities when adequately distributed. This motivates a combined use of the technologies. In which the Digital Twins are managed centrally and highly volatile data is processed on edge, i.e., close to the process for latency reduction, and only the resulting indices can be transferred at a significantly lower frequency and in a smaller band.

### 3. Proposed Method

#### 3.1. Concept

For an adaptive digital representation of a machine, its components, and tools as shown in Figure 2, methods for the continuous synchronisation of the Digital Twin concerning the state of the actual machine are required. The data should be collected and processed by

the machine or close to the machine. This can be carried out by Edge Devices placed at the machine, which compute and provide indices regarding the machine's state. As a result, the relevant machine data will never be lost to the machine environment, and the operator retains control over the raw data. In the future, Digital Twins will increasingly be operated independently of the machine in cloud structures [41]. For example, the current status index can be transmitted to these models in the event of a change or on request. In this way, machines and model operators can be separated. Starting from this model, either in the form of a digital shadow, inquiries can be made over the current ability of the machine. By an adjustment of the production in the machine, such as by the initiation of a tool change, Digital Twins can be realised.

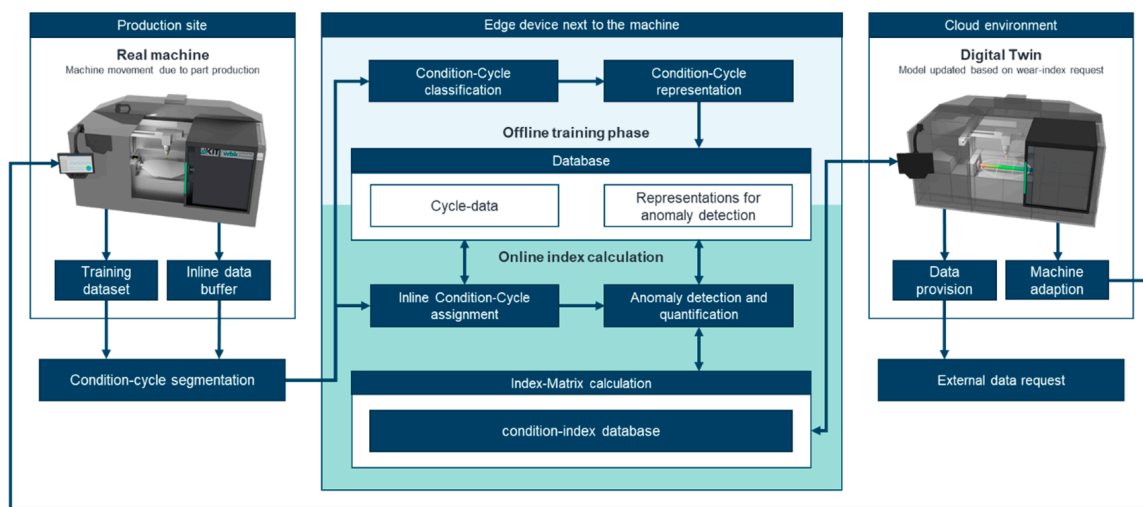


**Figure 2.** Approach of updated condition indices for components and tools for continuous mapping of the machine condition as well as the capability.

One basic assumption of the presented approach is that the production of lot size one of the highly individualised small batches can be understood as a series of individual repetitive motion sequences [10]. This will be exploited here by detecting and quantifying anomalies at a segment level. The models are trained during the offline training phase using a training dataset. During the online phase, the continuous index calculation and the update of these occur starting from the signal stored continuously with the respective machine movement. Common control signals such as position or spindle current are used for transferability. Starting from the time series, these must be sequenced in the first step. Based on this, the individual condition cycles are classified in the offline training phase. Since this involves a long time series and thus many different wear cycles and individual segments from ramp-up and transfer sections, representing noise in the classification, a robust algorithm must be used. Furthermore, in the offline training phase, the feature calculation for the inline classification, the determination of representative features and the cycle representations for calculating the wear indices are performed. At the end of the training phase, all relevant information is stored in a database accessed in the inline phase. The relevant indicators for registering machine and tool conditions depend on the used signals and can vary depending on these.

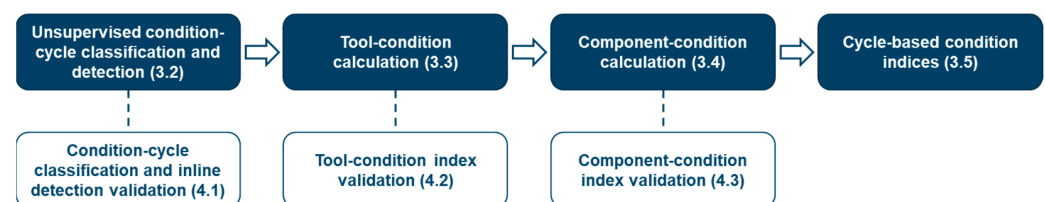
In the inline phase, it can be assumed that a time series consists of significantly fewer condition cycles. Thus, fewer have to be classified so faster algorithms can be used here. Following the classification, it is determined for each condition cycle found whether there is a deviation from the representatives stored in the database. If there is one, the deviation is quantified. Based on this quantification, the condition of the tool is determined for the respective cycles. The tool index calculation is updated once the tool has completed a life cycle. In the second stage, the index of all axes is calculated for the cycles. Through this continuous adaptation, a self-learning component is realised during the online phase. The calculations of the online and offline phases are to be performed on Edge Devices close to

the machine in a decoupled machine network to ensure no direct data access and a high level of cyber security. These calculations are executed on machine Industrial Personal Computers (IPCs) and transmitted as a condensed process wear matrix to a remote Digital Twin. This transfer to the cloud-based Digital Twin can be in the event of an index change or on request. The machine or production process can be adapted, or the data can be made available to external interested parties. This allows an early evaluation of the current machine's capabilities, or whether it is sufficient for a particular product. This modular approach as shown in Figure 3 enables also the usage of the developed algorithm without a digital twin. But for adaptive digital twins, a detection of tool and component wear is necessary. This interaction and the automated response to the generated predictions of a wear state will be addressed in future research.



**Figure 3.** General overview of the presented approach.

This paper presents the approach according to the steps shown in Figure 4. Thus, this section describes the time series segmentation procedure for obtaining the condition cycles and the offline classification. Based on this, an approach for a faster inline classification of the found segments is described. Furthermore, the calculation of the tool and component conditions based on them will be discussed, considering the condition cycles and the threshold values. The representative index matrix is derived from the condition cycles and the key indices of all components and tools. Subsequently, the condition-cycle detection, the tool condition index and the component condition index are validated using anonymous data. In the end, the overall approach is discussed.



**Figure 4.** Structure of the paper.

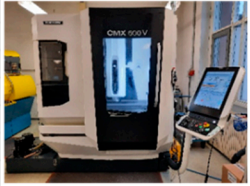
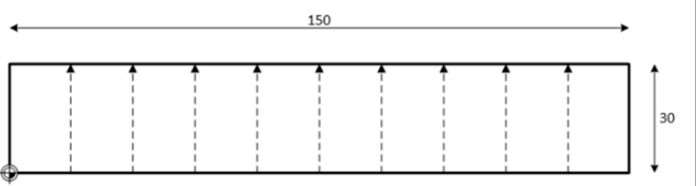
### 3.2. Unsupervised Condition-Cycle Classification and Detection

The basic assumption of the approach presented here is that individual sequences recur even in the case of a production run with a batch size of 1 [8,9]. It is further assumed that the state of the respective tool decreases steadily between the cycles [46]. To compare these cycles with each other over time, they must first be extracted using a suitable algorithm and then classified. To apply the approach to as many green and brownfield machines as possible, independently of the control system and the components installed, only common



control signals, such as the position of an axis or the spindle rotational speed, are to be used. In addition, the presented method should not require any additional force sensors to ensure inline capability. Therefore, the process forces, i.e., when the tool is engaged, are used.

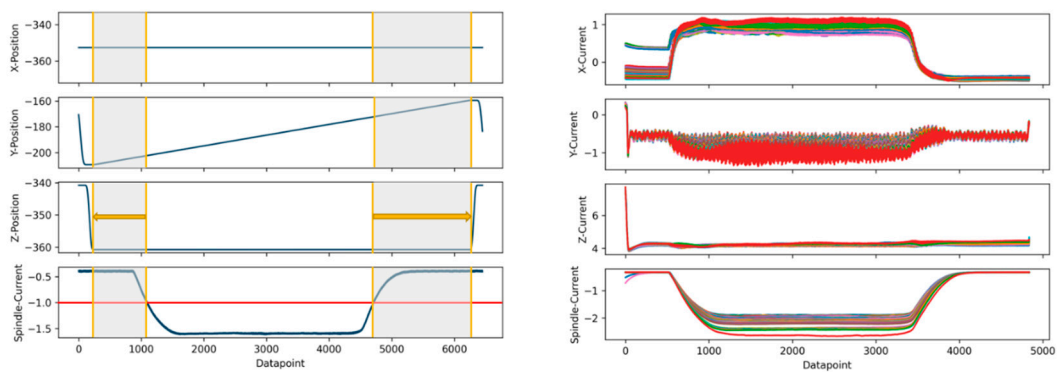
The algorithms were developed based on custom datasets, in which all tools were successively worn out based on recurring machining cycles until tool breakage occurred. The tests were conducted on a 3-axis CNC milling machine, CMD 600 V from DMG Mori, operated by a Sinumerik control system. In all the tests, an HSS Co 8 end mill with a diameter of 10 mm was used to machine C45 steel. The depth of cut is 5 mm in all experiments. The tool path and other experimental data can be seen in Figure 5. The recurring sequences are straight lines with a length of 30 mm along the Y-axis. The dataset consists of subgroups for which the technological values were adjusted. In the three tests of group 1, the machining was performed at a feed rate of 250 mm/min and a spindle speed of 3200 rpm until tool breakage. For the group two test, the feed rate of 312.5 mm/min and the spindle speed of 4000 rpm resulted from an overload of 25%. For the group three test, cooling lubricant was added. All signals were recorded with a sampling rate of 500 HZ.

test	feedrate [mm/min]	spindle speed [rpm]	lubricant	Test machine (CMX 600V)	tool path [mm]
1_1	250	3200	no		
1_2	250	3200	no		
1_3	250	3200	no		
2_1	312.5	4000	no		
2_2	312.5	4000	no		
2_3	312.5	4000	no		
3_1	312.5	4000	yes		

**Figure 5.** Design of experiments for unsupervised classification of condition cycles, inline detection and quantification.

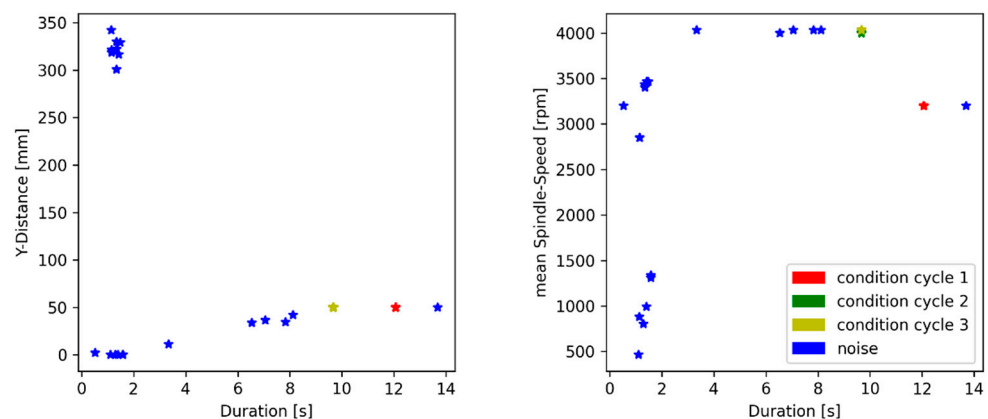
A two-stage approach is used for segmenting the individual sections. The starting point is the current of the main spindle motor. If the tool is not in use, only the movement of the spindle together with the chuck and the tool must be maintained. This results in a low and approximately constant current. Thus, by creating a threshold, it is possible to distinguish between areas outside and inside of a machining operation (see Figure 6 bottom left). For the proposed procedure, the time series of the spindle current is examined after smoothing. A potential segmentation candidate is found if this is in the range of a threshold value  $\varepsilon_\alpha$  (red line). If it is also ensured that only one candidate is considered per threshold value transition, the relevant range always results between two segmentation candidates.

As shown in Figure 6 on the bottom right, the position of the segmentation candidates shifts as the spindle current progresses through the condition of a tool. For this reason, only the rough position can be determined via the spindle current. However, this is not suitable for comparing the segments. For this reason, a constructive step follows. It is assumed that during machining, at least one of the axes will always move slowly and be repositioned at the end of the machining operation. This leads from a minimal movement to a rapid change of position in a rapid traverse. Since this repositioning connects the individual machining sequences, it is suitable for cutting out the condition cycles. For this, starting from two related segmentation candidates, a search is conducted for all axes for which the derivative of the position at the time of the cutting candidate is less than the threshold value  $\varepsilon_\beta$ . If the result exceeds the threshold, the start or end of the segment is determined. The yellow arrows illustrate this process in Figure 6, on the left side.



**Figure 6.** Procedure for the segmentation of the condition cycles (**left**) and progression of the signals during the wear process (**right**).

The cutting of the individual cycles takes place independently of the training and the inline application phase. Based on this, classification is carried out in the training phase by forming representatives of all underlying signals. The path length of the axes, the average spindle speed and the cycle duration are used to classify the cycles. These features were chosen because they describe the respective cycle and are position independent. The plots of individual characteristics can be seen in Figure 7. In the training phase, longer time series are evaluated, resulting in many sequences, whereby individual sections do not directly belong to a condition cycle. Accordingly, the method used for unsupervised learning must be robust concerning outliers. Furthermore, cycles that belong together form a region of particularly high density in the feature space. Furthermore, since duration plays a minor role in the learning phase, DBSCAN can be used [47]. As shown in [48], DBSCAN is well-fitted for anomaly detection and previous work also shows the capability in such use cases [49].



**Figure 7.** Feature space for segmenting the different condition cycles using the 1\_1, 2\_1 and 3\_1 datasets.

As shown in Figure 7, the tuples of individual cycles are arranged closely together and can therefore be grouped. The number of points per class is nearly equally distributed. This highlights the capability of the features for classification and cycle 3 differs from cycle 2 only by adding the cooling lubricant, which allows a slightly higher spindle speed. This allows a difference to be made via its average value (see Figure 6, right).

Finally, the calculation of the required representations for the inline index calculation is executed. In this case, the time series stored in the buffer is evaluated, which is shorter and contains fewer cycles. These cycles are now to be assigned to a class. For this purpose, the mean value and the standard deviation of the classes for each feature are calculated in the training phase. An assignment is made online if the respective mean lies within the sixfold standard deviation around the class mean.

Once the class has been determined, a deviation from the representative should be determined based on the change in the underlying condition-cycle signals. These must be determined during the training phase. Depending on the length of the cycle path, tool wear effects are not negligible, even within the first few cycles. For this reason, the first  $n_{Ref}$  cycles should always be used. In the context of this publication,  $n_{Ref}$  is chosen to be 3. Three approaches were examined to describe the deviation. Here, the mean value (a) and the average distance between all data points in a time series (b) are used. In addition, the reconstruction error of an autoencoder should be examined. For this purpose, the encoder consisting of a linear  $length_{cycle} \times 128$ , a ReLU layer, a linear  $128 \times 32$ , a ReLU and a linear  $32 \times 8$  layer, as well as an equivalent decoder is used. Due to the small amount of training data, it can be assumed that the reconstruction error approaches a distance measure. Table 1 shows the correlations between the anomaly dimensions when all signals assigned to a class are listed one after the other for the axes and the spindle. The high correlation shows that all distance measures can achieve similar results and are, interchangeable. However, it should be noted that due to the  $n_{Ref}$  and the associated small amount of training, the correlation with regard to the reconstruction error varies and is therefore only partially reproducible. Due to the high computational effort for the reconstruction error of the autoencoder, the average distance between the data points (b) is used in the following.

**Table 1.** Pearson Correlation coefficient for mean value (a), mean distance between the data points (b) and the autoencoder reconstruction error (c) for axis and spindle current signals. The coloring visualizes the correlation with white—no correlation and dark green—ideal correlation.

	Corr (a,b)	Corr (b,c)	Corr (c,b)
1 x-axis	-0.04939276	-0.16543531	-0.91010147
1 y-axis	-0.95581863	-0.98985663	0.94993983
1 z-axis	-0.70774908	0.97707274	-0.76128193
1 spindle	-0.99934485	-0.99784553	0.99735604
2 x-axis	-0.49271491	0.50516899	-0.85388119
2 y-axis	-0.76455602	-0.81543437	0.63313464
2 z-axis	-0.98536464	0.54104406	-0.53004049
2 spindle	-0.99961592	-0.9941404	0.99477251
3 x-axis	-0.75277514	0.59002364	-0.70386803
3 y-axis	-0.96254714	-0.96116497	0.93236389
3 z-axis	-0.99820867	-0.52270938	0.52980505
3 spindle	-0.99970386	0.97345975	-0.97130338

### 3.3. Tool Condition Calculation

The starting point for the determination of the tool condition is the course of the deviation over the identified cycles for the corresponding data signals. In the context of this paper, the approach is to be developed on the basis of the current signals of all axes, since here the tool condition has the greatest influence [10]. By lining up the datasets 1\_1, 1\_2 and 1\_3, as well as 2\_1, 2\_2 and 2\_3, and determining the course for the axes according to the deviation measure a, Figure 8 is obtained.

As can be seen, the course of the tool wear can be seen in both condition cycles. It should also be noted that, as expected, the tool wear has an increasing influence on the torque to be applied to the spindle. This results in an apparent increase in the spindle current and, thus, the deviation. Furthermore, it can be seen that especially in the second test series (cf. Figure 6 on the left), an apparent recurring deviation can be seen for the Y-axis. If this is compared with the deviations of the second dataset, it becomes clear that this does not necessarily have to be the case. Consequently, it can be stated that for the formation of the cycle-dependent tool condition index, all axes' signals should be considered. For

this reason, the index  $I_{C,Tool,i}$  for each cycle,  $C$  is calculated by the weighted sum of all deviations according to Equation (1).

$$I_{C,Tool,i} = \sum_N \alpha_{C,N,TCC} \cdot dev_{C,N,i} \quad N \in \{X - Axis, \dots, Spindle\} \quad (1)$$

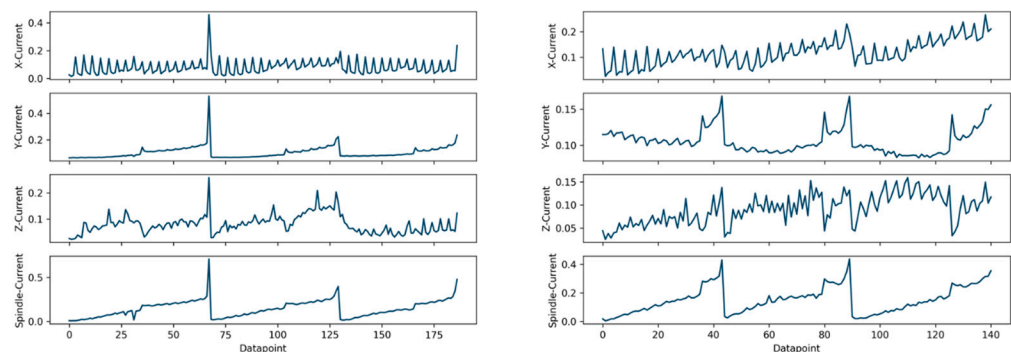
Here,  $\alpha_{C,N,TCC=1} = 1$  is selected as the initial value for the weighting.  $TCC$  represents the current run of the tool state and  $i$  the current cycle segment. Since an additional overlay with a deviation due to the component additions is assumed (cf. Section 2.3),  $\alpha_{C,N,TCC}$  must be continuously recalculated and adapted to the changed circumstances. This recalculation is always carried out when the tool is replaced by the operator or tool breakage  $i = i_{C,breake}$ . This is indicated by a rapid change in spindle deviation,  $dev_{C,N=Tool,i-1} - dev_{C,N=Tool,i} > \epsilon_{break}$ . When this occurs, the weighting is to be recalculated using the Spearman correlation of the deviations with an equivalent line  $\bar{w}_{C,TCC} = (0, \dots, length_{TCC-1} - 2)$ . This line represents the equivalent course of the tool wear in the stationary range [20]. According to this,  $\alpha_N$  in cycle  $c$  is cycle dependent according to Equation (2).

$$\begin{aligned} \alpha_{C,N,TCC} &= \alpha_{C,N,TCC-1} \cdot (1 - f_{N,TCC}) \\ &+ corr_{spear} \left( \overline{dev}_{C,N,TCC-1}, \overline{w}_{C,TCC} \right)^\beta \cdot f_{N,TCC} \end{aligned} \quad (2)$$

$$N \in [X - Axis, \dots, Spindle]$$

The cycle-specific weighting factor  $\alpha_{C,N,TCC}$  of the current tool condition cycle  $TCC$  for the axis or spindle  $N$  is calculated from the factor of the last cycle  $\alpha_{C,N,TCC-1}$  and the Spearman correlation between the deviations of the last cycle  $\overline{dev}_{C,N,TCC-1}$  and  $\bar{w}$  using the ratio  $f_{N,TCC}$ . Here, the values of the components, which show a high correlation, are to be strengthened by the exponent  $\beta$ . In the context of this paper,  $\beta = 3$  was chosen to realise a slight amplification. In the first calculation, the ratio  $f_{N,TCC=1} = 1$  is chosen. In the following new recalculations,  $f_{N,TCC} = 0.1$  is chosen in the context of this paper to consider the new cycle with a weighting of 10%.

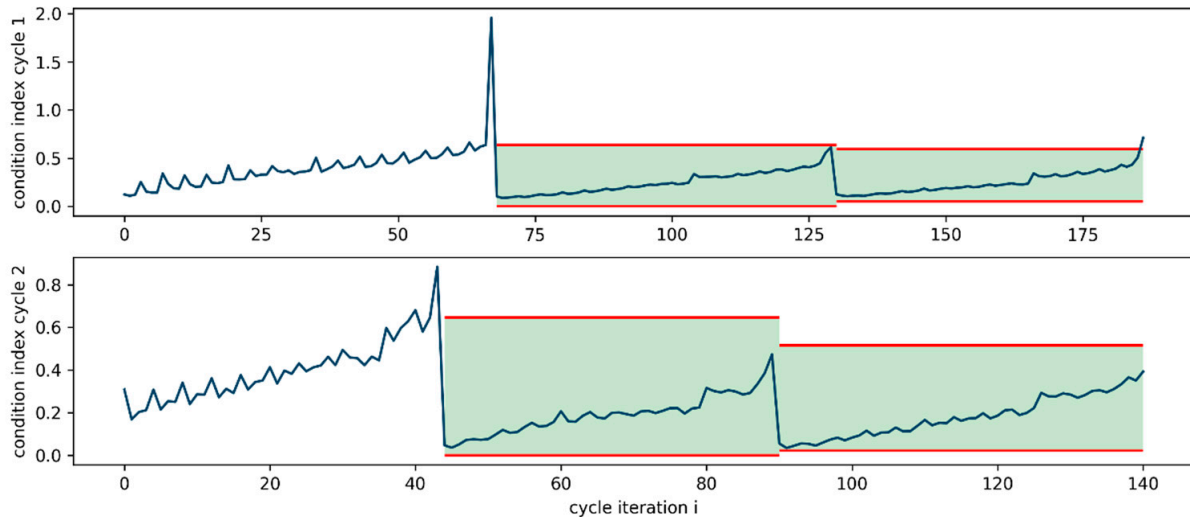
Furthermore, the current upper and lower thresholds are calculated for the quantification of the tool condition. In the initial calculation, the lower threshold  $\gamma_{Tool,C,lower} = 0$  and the upper threshold  $\gamma_{C,Tool,upper} = C_{Tool,C,i_{C,breake}-2}$  are used. In the recalculation, the values are calculated according to  $\gamma_{Tool,C,lower} = \gamma_{Tool,C,lower} \cdot (1 - f_{thrs}) + C_{Tool,C,i_{C,breake+1}} \cdot f_{thrs}$  where  $C_{Tool,C,i_{C,breake+1}}$  is the first value after the last recalculation. Further,  $\gamma_{C,Tool,upper} = \gamma_{C,Tool,upper} \cdot (1 - f_{thrs}) + C_{Tool,C,i_{C,breake}-2} \cdot f_{thrs}$  for the rotation of the threshold ratio  $f_{thrs}$ . Here,  $f_{thrs} = 0.5$  is chosen.



**Figure 8.** Deviation from the references of the classes for current signals of all axes and the spindle when using the 1\_1, 1\_2, 1\_3, 2\_1, 2\_2, 2\_3 and 3\_1 datasets.

Figure 9 shows the course of the tool condition index for cycle 1 (top) and cycle 2 (bottom). Here, the adjustment of the index value as a result of the respective recalculation can be seen after passing through  $TCC = 1$ . Furthermore, after the first run of the tools, the calculated upper and lower limit values (green areas) are shown. This shows an

overestimation of the limits or an underestimation of the tool condition as a result of the calculation of the values derived from the first tool using the initial index calculation. Accordingly, depending on the application, the question arises whether the limit values of the first cycle should be included in the calculation. As a result of the high value for  $f_{thrs} = 0.5$ , an adaptation to the real conditions takes place relatively quickly.



**Figure 9.** Tool condition index for the 1\_1, 1\_2, 1\_3, 2\_1, 2\_2, 2\_3 and 3\_1 datasets with updated upper and lower limits.

In the following, the algorithm for determining the cycle-specific tool condition index based on segmented and classified condition cycles and the component condition index is described as pseudo-code. The description of the component condition index calculation takes place in the following Section.

### 3.4. Component Condition Calculation

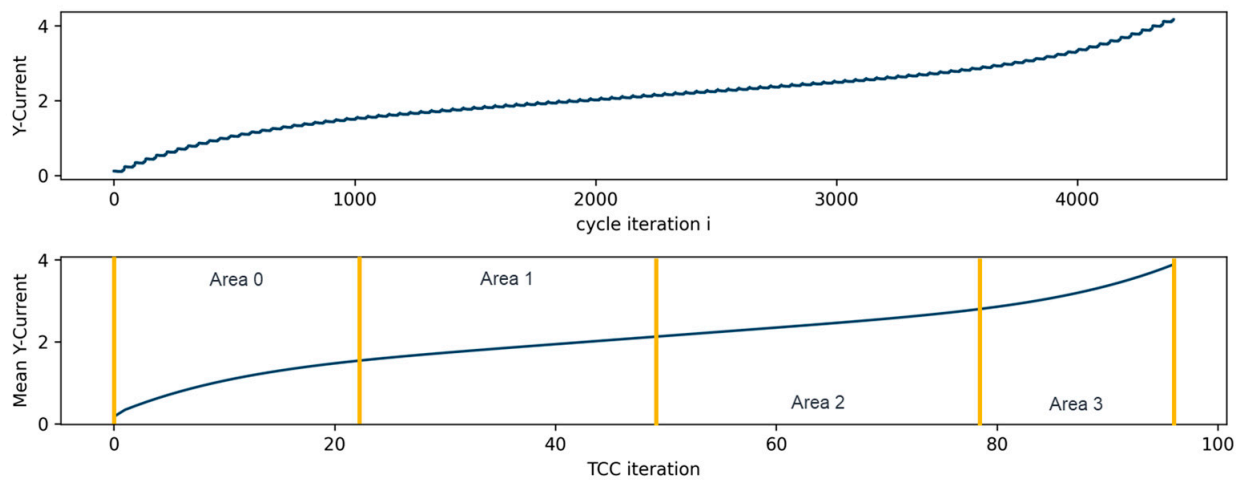
The effects of component wear, like tool wear, lead to a deviation from the respective cycle-specific reference signal. Accordingly, a superposition of the deviations occurs here. As shown in Section 2.3, several effects occur here. As a result of the loss of preload, there is a reduction in the input torque and, thus, in the current signal. The increase in friction, on the other hand, leads to an increased current signal. These effects superpose and form a combined deviation curve  $\tilde{dev}_{C,n,i}$ . It is assumed that the superposition of all wear  $w$  effects for  $w \rightarrow \infty$  will also result in a dominant curve in the form of a lifetime curve [50]. Since the wear of a component progresses much more slowly than a tool's, the respective component wear will become apparent over many cycles. Therefore, the component-specific deviation is overlaid with the function

$$\tilde{dev}_{C,n,i} = dev_{C,N,i} + \left( \frac{TCC}{50} - 1 \right)^5 + 1 + \frac{TCC}{50} \quad TCC \in [1, 100] \quad (3)$$

to simulate the deviation due to a superimposed lifetime curve. According to Equation (3), a superposition of the mean of the y-axis current signal deviation of datasets 3\_1, 3\_2 and 3\_3 will lead to the curve shown in Figure 10.

Since slower effects are present here as described, the mean value is calculated for the determination of the component state for all associated signals of the axes  $N$  after completion of a  $TCC$  in the context of the recalculation. Based on this, two possibilities are proposed here to determine the current state. The starting point is the nature of the service life curve. In this case, there is a steep slope at begin. Subsequently, a steady wear condition with minimum slope is established. Towards the end of the life cycle, however, the slope increases again. Since these gradient transitions are characteristic, states can be assigned to

them, as shown in Figure 10. This is, therefore, only a rough, qualitative additional estimate based on discrete stages. Quantification is possible if a state is assigned to the respective transitions (area 0 to 1, 1 to 2 and 2 to 3).



**Figure 10.** Superposed deviation of the y-axis current signal (**top**) and mean (**bottom**) based on TCC change for the 3\_1 dataset.

In order to ensure a robust determination of the condition, a pre-processing is required. For this purpose, the mean value over 15 *TCC* is determined based on the last 20 *TCC*. Based on this, the maximum of the change is determined in a neighbourhood of 5 *TCC*. For simplification, the auxiliary variables  $\ddot{d}ev_{C,N,TCC,max}$  and  $\ddot{d}ev_{C,N,TCC,min}$  are introduced.  $\ddot{d}ev_{C,N,TCC,max}$  is obtained from the mean value of the maximum derivative in a neighbourhood of 5 *TCC*.  $\ddot{d}ev_{C,N,TCC,min}$  represents the mean value of the minimum derivative in the same neighbourhood.

The rules for determining the respective transitions are shown in Table 2. Here, for the transition  $\epsilon_{dev,0\rightarrow1} = -0.001$ ,  $\epsilon_{dev,1\rightarrow2} = -0.001$  and  $\epsilon_{dev,2\rightarrow3} = -0.001$ . The quantification of the axis state  $C_{N,c,i}$  is performed by the comparison with the change of an area and the comparison with the corresponding *TCC*. Here,  $\partial_{dev,0\rightarrow1} = 0.2$ ,  $\partial_{dev,1\rightarrow2} = 0.6$  and  $\partial_{dev,2\rightarrow3} = 80$  were chosen. It should be noted that the percent depends on the superposition condition and therefore must be defined together.

**Table 2.** Component condition areas.

Area Transition	Condition	Percentage
0 to 1	$\ddot{d}ev_{C,N,TCC,min} > \epsilon_{dev,0\rightarrow1}$	$\partial_{dev,0\rightarrow1}$
1 to 2	$\ddot{d}ev_{C,N,TCC,max} > \epsilon_{dev,1\rightarrow2}$	$\partial_{dev,1\rightarrow2}$
2 to 3	$\ddot{d}ev_{C,N,TCC,max} > \epsilon_{dev,2\rightarrow3}$	$\partial_{dev,2\rightarrow3}$

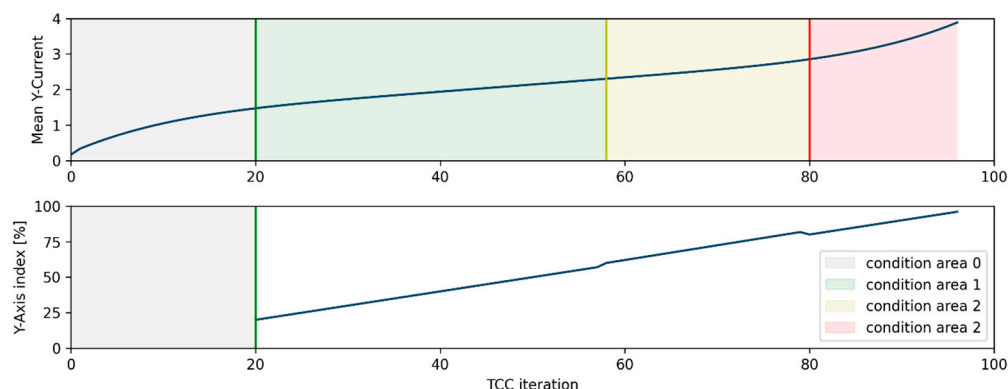
Figure 11 shows above the course of the averaged deviation of the *TCC* cycles for the current signal of the y-axis for dataset 3\_1. Based on these, the transitions of the ranges were determined as described above. The division into ranges represents a course four-step estimation of the component state. A percentage condition description is possible if the current value is compared with the transition values. As shown in Figure 11, no quantification is possible in condition area 0, but this is provided from the transition onwards.

To better understand the tool-state calculation (cf. Section 3.3) and the component-state calculation (cf. Section 3.4) described here, the two-step procedure is explained in the form of pseudo-code in the following. The condition cycles cut out and assigned according to Section 3.2 are the starting point for the calculations. The condition cycles extracted

from the current buffer data and assigned according to Section 1 are the starting point for the calculations. Based on the data now added to the database, the current index matrix is calculated.

If the condition for a recalculation and the associated component condition calculation is triggered by an operator-indicated tool change or tool breakage, the logic described in the following is executed for the condition cycle.

1. Repeat for Inline\_class  $c$  from 1 to  $C$  (*Inline classes*)
2.     Repeat for signal  $s$  from 1 to  $S$  (*data signals*)
3.         Repeat for timeseries  $t$  from 1 to  $T$  (*timeseries*)
4.             Calculate the current deviation for the specific signal using the representation (b)
5.             Update the Signal  $s$  Database based on timeseries  $t$  for Class  $c$   
update index matrix Load requirement matrix
6.     Repeat for  $m = 1, \dots, M$  (*samples*)
7.     Repeat for  $z = 1, \dots, Z$  (*teeth*)



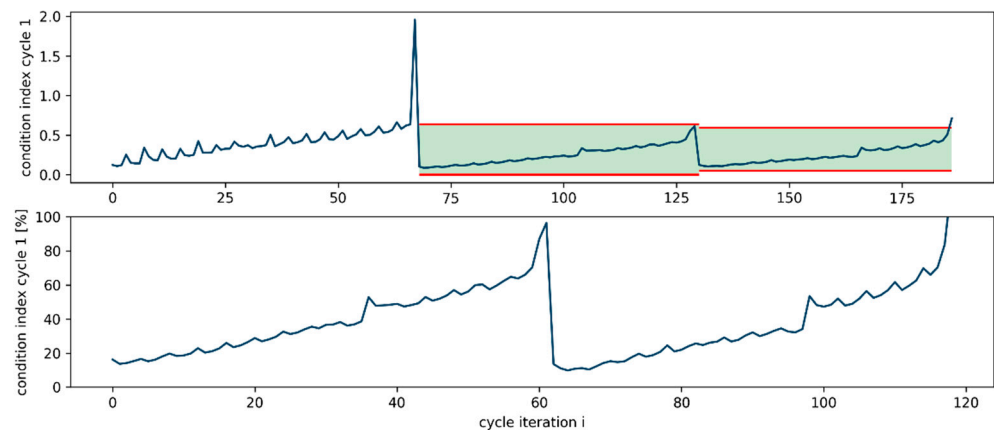
**Figure 11.** Superposed deviation mean of the y-axis current signal (**top**) with condition areas and derived condition index (**bottom**) based on TCC change for the 3\_1 dataset.

8. Repeat for class  $c$  from 1 to  $C$  (*classes*)
9.     Repeat for cycle iteration  $i$  from 1 to  $I$  (*added cycles*)
10.         If  $dev_{c,n=Tool,i} - dev_{c,n=Tool,i} > \epsilon_{break}$ .
11.             Recalculate Index basis
12.             Calculate the component condition index
13.         Else
14.             Calculate the current tool index  $C_{c,n=Tool,i}$  (*Equation (1)*)
15.             Calculate the Tool condition as percentage based on the
16.             upper and the lower threshold
17.     Return the current index matrix

18. If first calculation
19.     Override  $f_{N,TCC}$
20. Else use specific  $f_{N,TCC}$
21. Update  $\alpha_{C,N,TCC}$  (Equation (2))
22. Calculate the current tool index  $C_{c,n=Tool,i}$  (Equation (1))
23. Update the tool thresholds  $\gamma_{Tool,C,lower}$  and  $\gamma_{Tool,C,upper}$
24. Repeat for component  $n$  from 1 to  $N$  (Components)
25.     Calculate  $d\ddot{e}v_{C,N,TCC,min}$  and  $d\ddot{e}v_{C,N,TCC,max}$
26.     Determine the condition area and percentage basis (Table 3)

### 3.5. Cycle-Based Condition Indices

As described in Section 3.4, the cycle-specific component conditions are determined in the form of a discrete condition classification and as a percentage. According to Section 3.3, the cycle-specific tool condition  $C_{Tool,C,i}$  is characterised by the deviation index and the associated upper and lower threshold values  $\gamma_{Tool,C,lower}$  and  $\gamma_{Tool,C,upper}$ . For minimal effort data transfer according to Section 2.4, however, one index should always be sufficient for the condition description. If the current tool condition index is normalised by the use of  $\gamma_{Tool,C,lower}$  and  $\gamma_{Tool,C,upper}$ , a percentage value can be obtained. Figure 12 shows this conversion.



**Figure 12.** Tool condition index for the 1\_1, 1\_2 and 1\_3 datasets with updated upper and lower limit (**top**) and as percentage (**bottom**).

At this point, the machine's tools and components have specific conditions for each cycle, represented by a percentage value. This can be represented in the combined form of a matrix, as shown in Figure 13.

The state of the tool for processing the cycle  $c$  for the cycle iteration  $i$  is given by the tool state index  $C_{Tool,C,i}$ . Further, the index of the component  $n$  is given by  $C_{c,n,i}$ . Accordingly, the component-specific index can be given cycle independently as a column vector by outputting a column; if, on the other hand, the current capability is required for the execution of a specific state cycle, this is given by row  $c$  of the matrix. Here, due to the features representing a condition cycle, it is not bound to a specific axis position to be able to map a general and pose-independent capability. The cycle-specific capability is then in the form of a row vector. In addition, each index can be output individually. For each cycle iteration, the update of the index matrix is performed.



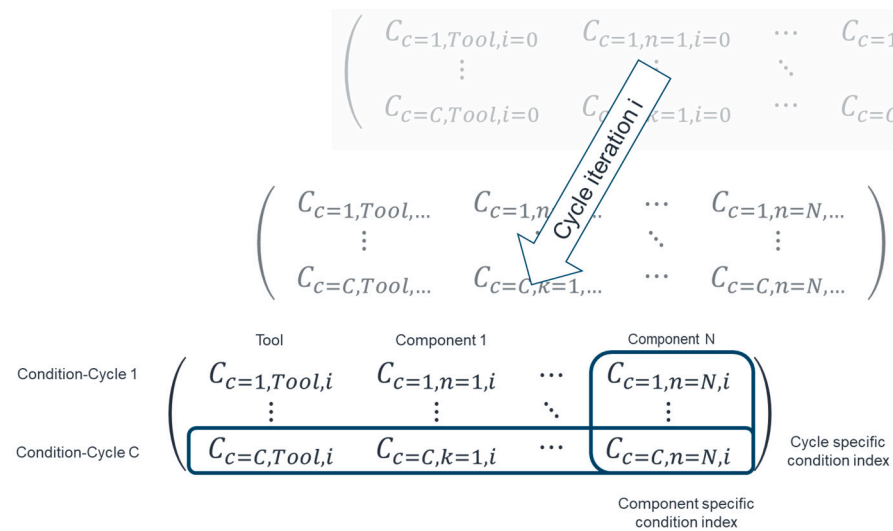


Figure 13. Proposed condition index format.

As described in Section 3.1, index monitoring is to be performed at IPCs close to the machine to ensure data sovereignty. The IPC continuously provides the current condition matrix. Its transmission can be carried out in several ways, depending on the requirements of the Digital Twin. On the one hand, a transmission can be made to the Digital Twin in case of an index change. On the other hand, the Digital Twin can make a request if necessary. On the one hand, the entire matrix can be transmitted. However, depending on the subscriber and its needs, transmitting only the cycle-specific condition vector, the composition-specific condition vector or specific indices is possible. This way, the amount of data to be transmitted can be minimised according to the specific application.

#### 4. Validation

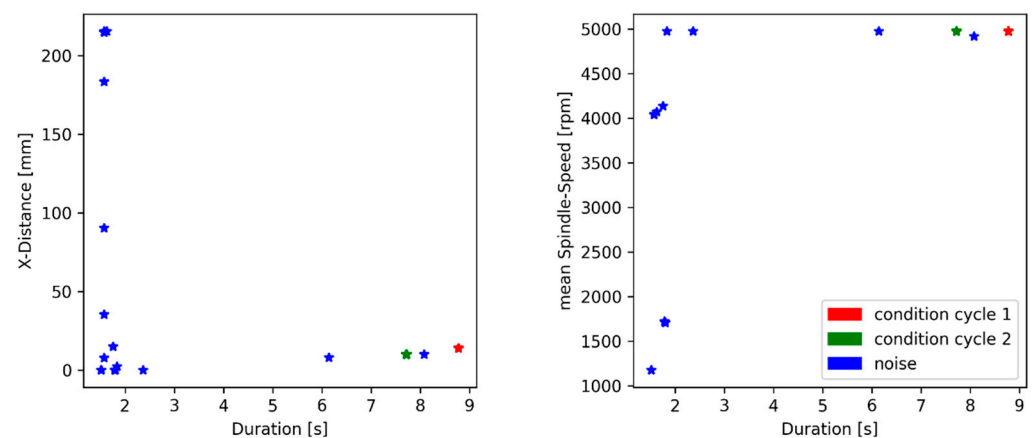
##### 4.1. Condition Cycle Classification and Inline Detection

New validation datasets were created for the validation. For this purpose, HSS Co 8 end mills with a diameter of 8 mm were worn by the recurrent machining of C45. The tests were conducted on the 3-axis CNC milling machine type CMD 600 V from DMG Mori. Two series of tests were carried out with three tests each. Machining was performed at a 400 mm/min feed rate and a spindle speed of 4975 rpm. In tests 1\_1, 1\_2 and 1\_3, the 30 mm thick steel was machined by moving the Y-axis, followed by the X-axis and then by moving both axes together. No cooling lubricant was used in this process. For datasets 2\_1, 2\_2 and 2\_3, machining was performed by moving the X and Y axes. At this moment, a cooling lubricant was used. The test data, as well as sketches of the movement, can be seen in Figure 14.

test	feedrate [mm/min]	spindle speed [rpm]	lubricant	Test machine (CMX 600V)	tool path [mm]
V1_1	400	4975	no		
V1_2	400	4975	no		
V1_3	400	4975	no		
V2_1	400	4975	yes		
V2_2	400	4975	yes		
V2_3	400	4975	yes		

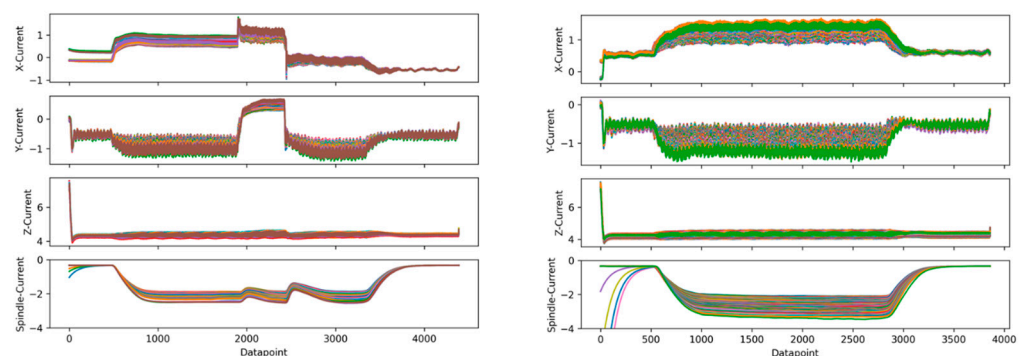
Figure 14. Design of experiments for the validation datasets.

If the datasets V1\_1 and V2\_1 are used for the training of the presented approach, and the extracted features of the travelled distance in the X-direction, as well as the average rotational speed of the spindle, are plotted over the duration of the sections, a clear clustering of the relevant cycles can be seen (cf. Figure 15). Hereby, it was not necessary to adjust the cutting parameters  $\epsilon_\alpha$  and  $\epsilon_\beta$ . Thus, the transferability of the approach for cutting and the assignment to other condition cycles could be demonstrated. Since no threshold value adjustment was necessary, it is further shown that a transfer can be carried out with minimal effort. This is possible as long as the inertia of the spindle drive train does not deviate significantly. Thus, the selected characteristics are sufficient in this case. In particular, the cycle duration and the differences in the distance covered in the Y-direction represent the relevant features. Furthermore, it is shown that the segmentation algorithm can reliably find the corresponding segments. If the 6 simple features reach their limits, they can easily be extended. For example, derivatives or tool parameters can be used for additional differentiation.



**Figure 15.** Feature space for the segmentation of the different condition cycles using the V1\_1, V2\_1 dataset.

The current signal curves belonging to the extracted condition cycles can be taken from Figure 16. Here, clear differences can be seen. In the recordings of datasets V1\_1, drops in the spindle current intensity as a result of the change in direction can be seen. There are also rapid changes in the X- and Y-axis curves. In the case of dataset V2\_1, approximately constant current curves are formed.

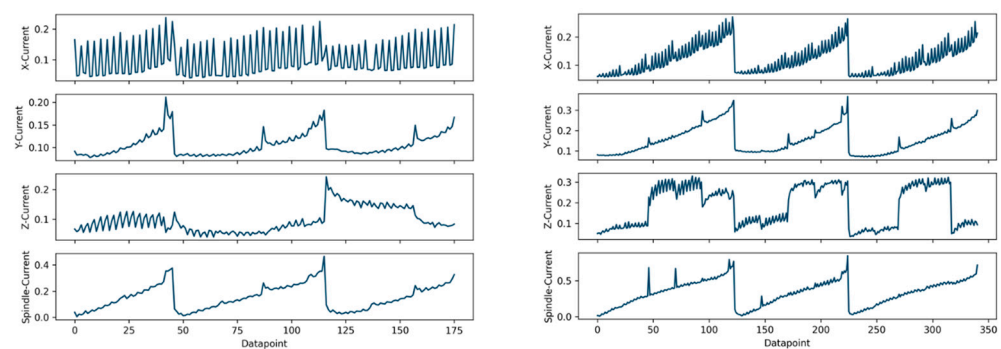


**Figure 16.** Current signal for the extracted condition cycles of the V1\_1, V2\_1 dataset.

#### 4.2. Tool Condition Index

For the validation of the approach to determine the tool condition, all evaluation series (V1\_1, V1\_2, V1\_3, V2\_1, V2\_2 and V2\_3) are evaluated together. Analogous to Section 3.3, the deviation from the references is determined using the distance measure (b). Figure 17 plots the deviations for the axes' and spindle current signals over the extracted state cycles.

For the evaluation of the deviation measure, a strong correlation to the tool wear condition is shown for the spindle and the deviation of the Y-axis. For all three tools, a continuous increase in the deviation can be seen. If breakage occurs, these deviations decrease abruptly. In the data of the second series of tests, the tool wear effect can be observed throughout. In the course of the deviations for the spindle current, the course of the tool life curve can be recognised. On the other hand, there is a slightly exponential course for the deviation of the Y-axis. This can also be recognised in the course of the deviation of the X-axis. Furthermore, the effect of an increasing deviation can be seen for the Z-axis. Here, however, the levels are rather discrete, probably caused by the change in the raw material blocks to be processed. Nevertheless, a clear tendency can be recognised over a tool's lifetime. In general, it can be seen that the progression for the second validation series is more detailed than the first one. This can be explained by the significantly increased lifetime of the tool as a result of the added cooling lubricant.



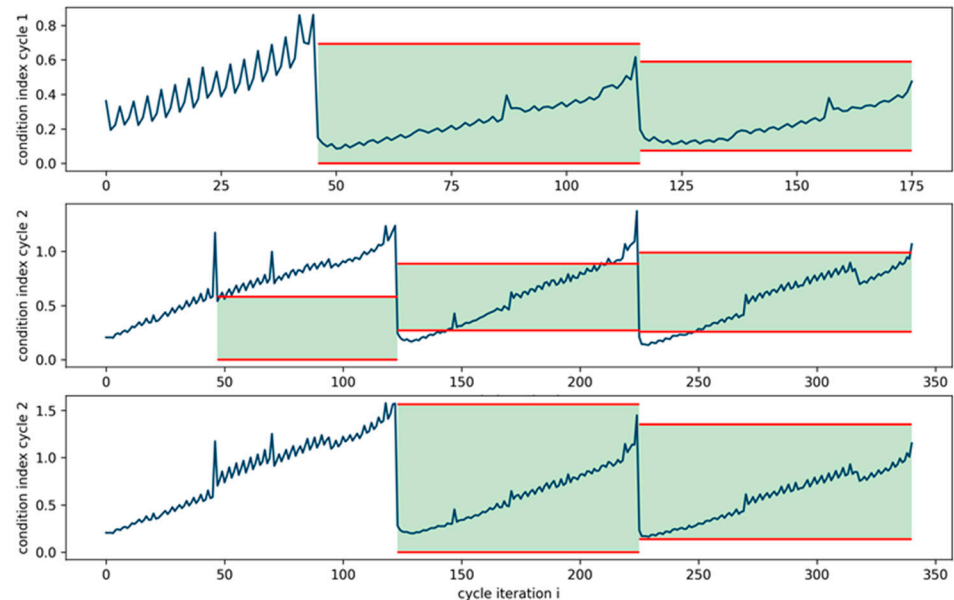
**Figure 17.** Deviation from the references of the classes for current signals of all axes and the spindle when using the V1\_1, V1\_2, V1\_3 on the left side and V2\_1, V2\_2, V2\_3 as validation datasets on the right.

If the tool index is calculated for the first validation dataset using Equations (1) and (2), the curve shown in Figure 18 at the top is obtained. None of the introduced parameters had to be adjusted here. Nevertheless, it can be seen that the changes in the used tools were recognised correctly. The adjusted curves show reduced noise, which significantly improves the derivation of the tool condition. A strongly linear mapping of the condition emerges for the first validation dataset. The formation of the threshold values is also successive. Thus, for this dataset, it is shown that the approach can lead to a suitable mapping of the condition state without further adjustment.

If the condition of the tool is determined for the second validation dataset without an adjustment of the parameters, the course and the threshold values from the middle plot are drawn. It can be seen that the peaks in the deviation, which are caused by the change of the workpiece, lead to an incorrect evaluation. In this case, the occurring deviation is interpreted as a tool change. As a result, the first re-evaluation is carried out. This can be suitably compensated for in calculating the index by the subsequent re-evaluations. In the case of the thresholds, however, this erroneous recalculation leads to problems, since the upper threshold  $\gamma_{Tool,C,upper}$  and the lower threshold  $\gamma_{Tool,C,lower}$  of the next TCC are calculated incorrectly. This error can be compensated for by the next TCCs. Thus, erroneous recalculations can be compensated by continuously updating the thresholds. However, it remains to be checked whether better results can be achieved by adjusting the threshold  $\epsilon_{break}$  of the recalculation. If  $\epsilon_{break} = -0.3$  is changed to  $\epsilon_{break} = -0.4$ , the plot in Figure 18 at the bottom can be obtained. Here, the change could be detected correctly. As a result, the recalculation and the thresholds are calculated at the correct cycle, which already leads to more suitable thresholds from the beginning.

In summary, it can be stated that the approach to the validation datasets with adjustment of the threshold value  $\epsilon_{break}$  was possible in both cases. It should be ensured that the tool break or its change always causes the difference. Thus, this must be aligned with the

maximum change. Under the assumption of superimposed axis effects, however, absolute values must be assumed here, since the total deviation level increases, while the change remains approximately constant.



**Figure 18.** Tool condition index for the V1\_1, V1\_2, V1\_3, V2\_1, V2\_2 and V2\_3 datasets with updated upper and lower limits (**top, middle**) and using an adapted  $\epsilon_{break}$  for the V2\_1, V2\_2 and V2\_3 datasets (**bottom**).

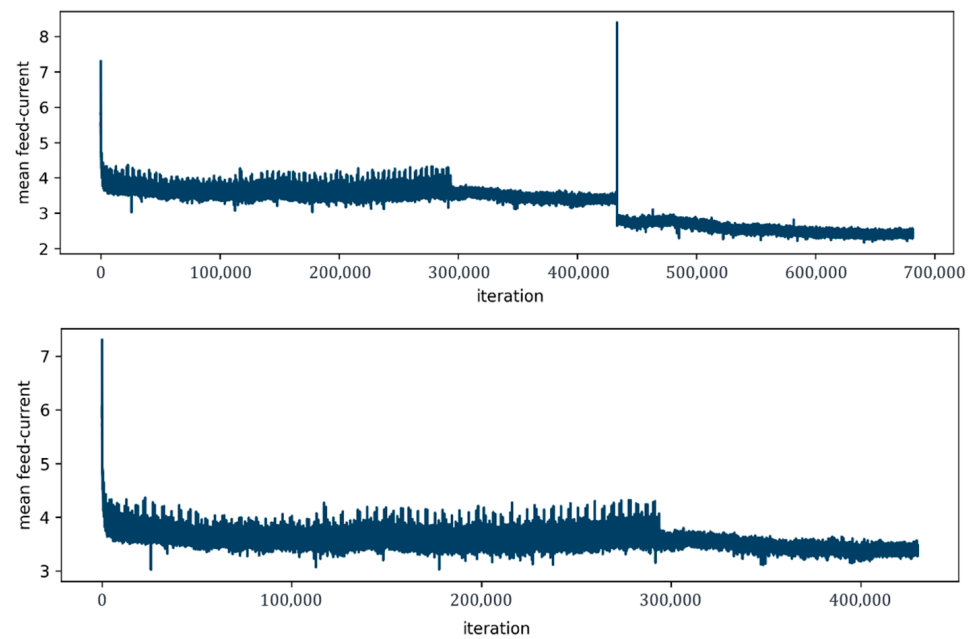
#### 4.3. Component Condition Index

Ball screws were worn out in long-term tests to validate the component indices. For this purpose, a ball screw of type BASA  $32 \times 20 R \times 3.969 - 3$  was clamped into a test stand. In the test scenario, a Force  $F_m = 13.5$  KN was applied on the components which represents a relative load of  $\sim 40\%$  of  $C_0 = 33,700$  KN. The axis was periodically moved over a distance of 650 mm with a rotational speed of  $n_m = 400$ rpm, so various wear effects became apparent in this area. These test conditions lead to an analytical lifetime estimation through equations 4 and 5 of 259.6 h. An iteration in Figure 19 correlates to a duration of  $1.35 \times 10^{-3}$  h through the linear distance, rotational speed and incline. Therefore, the analytical end of the component life is expected after 192,296 iterations.

$$L_{1,2} = \left( \frac{C_0}{F_m} \right)^2 \cdot 10^6 \quad (4)$$

$$L_h = \frac{L_{1,2}}{n_m \cdot 60} \quad (5)$$

The most relevant effect, in the beginning, is the relatively higher loss of preload, which initially leads to a sharp drop in the required torque. This effect decreases in the early phase of the component life. Further, friction increases and damage occurs on the spindle surface. All three effects affect the spindle current to be applied by the drive. Figure 19 on the left shows the average current over each period, which must be applied within a period. The strong peak in the middle of the time series, where the nut ruptured, is particularly noticeable here. Additionally, in the early phase of the curve from Figure 19, there is a larger noise that results from lubrication cycles executed every 4000 iterations. After  $\sim 215,000$  iterations first microparticles could be seen in the lubricant through a finger sample. The number and size increased until the end of the useful component life was classified through a manual inspection after 290,000 iterations.

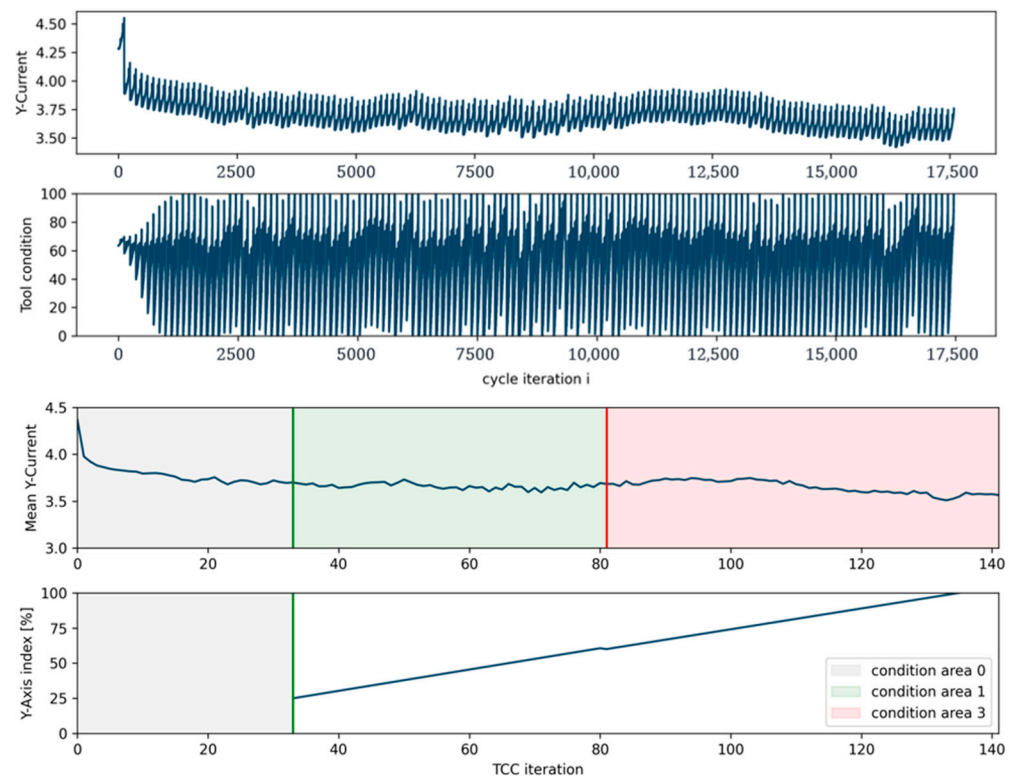


**Figure 19.** Mean feed current of the ball screw drive during the wear experiments.

After the end of the useful component life, no further lubrication cycles were performed, resulting in the significantly less noisy signal in Figure 19. The buildup of damage in the contact area of the balls (pittings) and a corresponding degradation of the balls led to the destruction of the rolling elements. This destruction significantly influenced the rolling capacity of other balls, which in the next step led to the rupturing of the nut. This also explains the subsequent drop in current. In the following, the occurrence of the component failure is defined as the end of the component life, and the time series is thus shortened. Furthermore, by forming the mean value over 3000 periods, the current signal is smoothed in order to depict the long-term course of the effects.

If the signal curves of the deviations are superimposed with the influence of the components as described in Section 3.4, the curve shown in Figure 3 above results for the Y-axis in test X, for example. It should be noted that datasets V2\_1, V2\_2 and V2\_3 were used iteratively. It can be clearly seen that the cycles of the tool and the global component influence lead to separate effects. It has to be checked whether the tool condition can still be mapped and the component condition can be extracted. The third and fourth plots in Figure 20 show the recurring status of the tool. In the first cycle, the percentages mapping starts around 60% but widens to the entire 100%. Corrective actions can be over the course of the process.

When considering the course of the component-related signal change in Figure 20, it can be seen that the first characteristic transition, which represents the initiation of the constant progression of the wear process, is pronounced. However, based on this, a clearly more linear phase cannot be detected. Accordingly, only the transition between the areas 0 to 1 and 2 to 3 is suitable. The transitions detected according to Section 3 without adjustments can be seen in the third image in Figure 20. The transition between the 0 to 1 range occurs here after 23.239% and therefore corresponds to the expectation. The transition between ranges 2 to 3 is detected at the first significant slope change. This corresponds to a local increase after 57.042% which is equivalent to ~239,576 iterations in Figure 20, interceding the end of useful component life. If, for example,  $\partial_{dev, 0 \rightarrow 1} = 25\%$  and  $\partial_{dev, 2 \rightarrow 3} = 60\%$  are specified, then the index progression is obtained in the image below in Figure 20. In summary, it can be stated that the transition from 0 to 1 is recognised somewhat too late, whereas the phase transition between 2 to 3 is recognised correctly.



**Figure 20.** Superimposed deviations of the Y-axis current signal with the course of the current signal of the long-term test, the mapped component index, the reconstructed component course with the detected areas and the index course derived from it using the V2\_1, V2\_2 and V2\_3 validation datasets.

## 5. Discussion

### 5.1. Wear-Cycle Detection

The algorithm for detecting wear cycles needs a minimum cutting depth for thresholding. This results in a dependence on the milling tool and machine. However, this threshold can be adjusted through the training phase on the machine. Here, adaptive thresholding can be used to generalise this approach. These intelligent approaches for adaptive threshold selection would show significant potential in optimisation. Without this, the proposed approach is very static, with a minimum cutting depth, which makes it robust in these cases. The static threshold selection of the approach only works in instances where the tool passes through a component and not during changes in milling depth. Furthermore, there is no segmentation for complex multi-axis shapes. However, adaptive thresholding approaches can address this issue and expand the algorithm's scope. The segmentation algorithm is stable and promising in a possible generalisation to other paths. However, further experiments are needed to analyse the interaction between the path and threshold, and approaches for transfer to other dynamics are required. Nevertheless, for the given and investigated cases, the segmentation performed well.

### 5.2. Tool Wear Detection

The parameter selection for tool wear detection is crucial in functional performance. It must be tested for generalisability, as demonstrated by the need for adaptive parameter selection in  $\epsilon_{break}$ . The interactions between parameters and their effects on prognostic errors and interactions between segmentation and tool wear detection have not been considered, leaving optimisation potentials. However, the change in  $\epsilon_{break}$  shows the possibility of reacting stably to trajectory changes with adapted parameters. This leads to the conclusion that special approaches for intelligent parameter selection should be further focused on. As previously mentioned, there is also the question of whether the first cycle should be considered, as it can strongly shift weights. Nevertheless, it can be concluded

that the approach presented in this paper enables reliable detection of the wear state during operation solely through control signals.

### 5.3. Component Wear Detection

The test for detecting component wear was validated through a test rung on a test bench. This reduces the complexity of the super-positioned wear progress. Here, averaging was performed over several iterations to reduce the noise caused by lubrication cycles. This raises the question of the influence of the averaging and the calculation over  $n$  TCC and the resulting effects. Further experiments are recommended to quantify or at least estimate the relationship. The same applies to the test wench for the wear of ball screws.

The existing results show a stable classification from the second to the third range. Still, the timing of the transition between 0–1 occurred later than expected, which raises the question of the validity of a single experiment. In addition, the superposition between component and tool is a new approach that can close the existing gap. Still, questions remain regarding the possible interaction of the wear mechanisms between the tool and components and between the different components within the machine. But for the given experiments and the corresponding data, the algorithm for detecting superpositioned tool and component wear classifies the end of the useful component life as reliable.

### 5.4. Cycle-Based Two-Stage Tool and Component Condition Index

This paper proposed an approach that is a promising solution to detect and quantify the condition of components and tools without sensors during operation. This can combine with edge-based calculation and pure index transmission to cloud-based Digital Twins, providing data security. While the results are promising, there are still questions about the robustness of the approach in different machines and the impact of parameter selection in TCC on component wear and segmentation quality. Additionally, the interactions between individual modules have not yet been investigated, which offers further research potential.

## 6. Conclusions and Outlook

In conclusion, this paper presents a novel approach for segmenting and classifying individual machining cycles using common control signals to monitor tool and component wear in CNC milling machines. The proposed method assumes that individual sequences recur even with a batch size of 1 and that the state of the tool decreases steadily between cycles. The algorithms were developed based on custom datasets obtained from a 3-axis CNC milling machine, and the approach was shown to be effective in segmenting individual cycles and classifying them based on path length, average spindle speed and cycle duration. To form a cycle-dependent tool condition index, all signals of the axes should be considered, and the index is calculated using a weighted sum of all deviations. The upper and lower thresholds are calculated to quantify the tool condition. The same algorithm was adapted to also predict the component wear progress for a superimposed case as seen in machine tools. A pre-processing method is used to equalise tool wear influence on the underlying component wear curve to ensure the robust determination of the condition. The component state is quantified by comparing it with the change of a range of the corresponding TCC, and a percentage condition description is possible based on this. The proposed method provides a four-class estimation of the component state. In summary, the proposed method can estimate the wear in green and brownfield machines and does not require additional sensors to ensure inline capability. The approach has been tested and demonstrated robustness in various validation cases. The modular approach allows for the application of the developed algorithm even in the absence of a digital twin. However, to facilitate adaptive digital twins, it is crucial to identify tool and component wear. Future research will delve into addressing the interaction and automated response to the predicted wear states.

As it stands, the presented approach still has optimisation potential to further increase the wear estimation's precision. Future work will focus on incorporating Machine Learning

techniques to determine threshold values for the proposed approach, which could help improve the estimations' accuracy. Through this, it can be expected to see a significant improvement in the proposed approach's precision, making it an even more valuable tool for optimising processes with Digital Twins. Furthermore, incorporating additional signal inputs in the presented approach is another perspective. Here, the control error between the motor encoder and the direct position is a promising factor in supporting the estimation of component wear. This additional position information could also enable the consideration of position-specific conditions. Integrating the approach into adaptive Digital Twins should also be explored to automate data mapping and transmission. This will allow for seamless integration with different machines and systems and enable real-time optimisation of processes. Finally, a mechanism that only triggers a recalculation of wear metrics in case of a failure or replacement of tools or components is considered, not when initiated by the Digital Twin. This will minimise unnecessary calculations and improve the efficiency of our approach. We believe that this integration will be a significant step towards achieving autonomous production in Industry 4.0.

**Author Contributions:** Conceptualization, R.S. and A.B.; methodology, R.S.; software, R.S.; validation, R.S. and A.B.; formal analysis, R.S.; investigation, R.S. and A.B.; resources, J.F. and A.W.; data curation, R.S. and A.B.; writing—original draft preparation, R.S. and A.B.; writing—review and editing, R.S., A.B., A.W. and J.F.; visualization, R.S.; supervision, J.F. and A.W.; project administration, R.S. and J.F.; funding acquisition, J.F. and A.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This publication is based on the research results of the project “SDMFlex-Flexible SDM through Continuously Quality-aware Digital Twins”. The authors would like to thank the Ministry of Science, Research and Arts of the Federal State of Baden-Württemberg for the financial support of the projects within the InnovationCampus Future Mobility (ICM).

**Data Availability Statement:** Data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy issues.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

Parameter	Value
IPC	Industrial PC
$\epsilon_\alpha$	Spindle current threshold for relevant sections
$\epsilon_\beta$	Axis position derivative threshold for a relevant section
$n_{Ref}$	Number of condition cycles used for the reference
$length_{cycle}$	Length of a condition cycle
deviation measure a	Mean value
deviation measure b	The mean distance between the data points
deviation measure c	Autoencoder reconstruction error
$I_{c,Tool,i}$	$i^{th}$ condition index calculation for the tool in condition cycle $c$
$i$	$i^{th}$ condition index calculation
$C$	Condition-cycle number
$N$	Axis or spindle reference, $N \in [X - Axis, \dots, Spindle]$
$\alpha_{C,N,TCC}$	Weight for the deviation of component $C$ during tool run $TCC$
$dev_{C,N,i}$	Current deviation for component $N$ in cycle $C$ during the $i^{th}$ condition index calculation
$TCC$	Lifecycle of a tool
$\epsilon_{break}$	Threshold for tool replacement indicating a recalculation
$f_{N,TCC}$	Ratio factor for the weight recalculation
$corr_{spear}(\dots)$	Spearman correlation
$\beta$	Correlation exponent
$\gamma_{Tool,C,lower}$	Lower tool condition threshold
$\gamma_{C,Tool,upper}$	Upper tool condition threshold



$\ddot{dev}_{C,N,TCC,min}$	Second derivatives for component $N$ in condition cycle $C$ during TCC using the local minimum
$\ddot{dev}_{C,N,TCC,max}$	Second derivatives for component $N$ in condition cycle $C$ TCC using the local maximum
$\epsilon_{dev,0\rightarrow1}$	Threshold for a change from stage 0 to 1
$\epsilon_{dev,1\rightarrow2}$	Threshold for a change from stage 1 to 2
$\epsilon_{dev,2\rightarrow3}$	Threshold for a change from stage 2 to 3
$\partial_{dev,0\rightarrow1}$	Percentage for a change from stage 0 to 1
$\partial_{dev,1\rightarrow2}$	Percentage for a change from stage 1 to 2
$\partial_{dev,2\rightarrow3}$	Percentage for a change from stage 2 to 3

## References

- Xu, X. Machine Tool 4.0 for the new era of manufacturing. *Int. J. Adv. Manuf. Technol.* **2017**, *92*, 1893–1900. [CrossRef]
- Iglesias, A.; Taner Tunç, L.; Özşahin, O.; Franco, O.; Munoa, J.; Budak, E. Alternative experimental methods for machine tool dynamics identification: A review. *Mech. Syst. Signal Process.* **2022**, *170*, 108837. [CrossRef]
- Rüßmann, M.; Lorenz, M.; Gerbert, P.; Waldner, M.; Justus, J.; Engel, P.; Harnisch, M. Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries. 2015. Available online: [https://inovasyon.org/images/Haberler/bcgperspectives\\_Industry40\\_2015.pdf](https://inovasyon.org/images/Haberler/bcgperspectives_Industry40_2015.pdf) (accessed on 16 February 2023).
- Chabanet, S.; El-Haouzi, H.B.; Thomas, P. Toward a self-adaptive Digital Twin based Active learning method: An application to the lumber industry. *IFAC-PapersOnLine* **2022**, *55*, 378–383. [CrossRef]
- Dalibor, M.; Michael, J.; Rumpe, B.; Varga, S.; Wortmann, A. Towards a Model-Driven Architecture for Interactive Digital Twin Cockpits. In Proceedings of the International Conference on Conceptual Modeling, Vienna, Austria, 3–6 November 2020; Springer: Cham, Switzerland, 2020; pp. 377–387.
- Cook, A.A.; Misirli, G.; Fan, Z. Anomaly Detection for IoT Time-Series Data: A Survey. *IEEE Internet Things J.* **2020**, *7*, 6481–6494. [CrossRef]
- Seevers, J.-P.; Johst, J.; Weiß, T.; Meschede, H.; Hesselbach, J. Automatic Time Series Segmentation as the Basis for Unsupervised, Non-Intrusive Load Monitoring of Machine Tools. *Procedia CIRP* **2019**, *81*, 695–700. [CrossRef]
- Netzer, M.; Palenga, Y.; Goennheimer, P.; Fleischer, J. Offline-Online pattern recognition for enabling time series anomaly detection on older NC machine tools. *J. Mach. Eng.* **2021**, *21*, 98–108. [CrossRef]
- Netzer, M.; Palenga, Y.; Fleischer, J. Machine tool process monitoring by segmented timeseries anomaly detection using subprocess-specific thresholds. *Prod. Eng. Res. Devel.* **2022**, *16*, 597–606. [CrossRef]
- Putz, M.; Frieß, U.; Wabner, M.; Friedrich, A.; Zander, A.; Schlegel, H. State-based and Self-adapting Algorithm for Condition Monitoring. *Procedia CIRP* **2017**, *62*, 311–316. [CrossRef]
- Jove, E.; Casteleiro-Roca, J.-L.; Quintián, H.; Méndez-Pérez, J.A.; Calvo-Rolle, J.L. A fault detection system based on unsupervised techniques for industrial control loops. *Expert Syst.* **2019**, *36*, e12395. [CrossRef]
- Theumer, P.; Zeiser, R.; Trauner, L.; Reinhart, G. Anomaly detection on industrial time series for retaining energy efficiency. *Procedia CIRP* **2021**, *99*, 33–38. [CrossRef]
- Christ, M.; Kempa-Liehr, A.W.; Feindt, M. Distributed and parallel time series feature extraction for industrial big data applications. *Neurocomputing* **2017**, *307*, 72–77. [CrossRef]
- Zhang, Z.; Wu, X.; Liu, T.; Liu, X. Fault diagnosis of planetary gear backlash based on motor current and Fisher criterion optimized sparse autoencoder. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2022**, *236*, 7529–7545. [CrossRef]
- Teti, R.; Jemielniak, K.; O'Donnell, G.; Dornfeld, D. Advanced monitoring of machining operations. *CIRP Ann.* **2010**, *59*, 717–739. [CrossRef]
- Hatt, O.; Crawforth, P.; Jackson, M. On the mechanism of tool crater wear during titanium alloy machining. *Wear* **2017**, *374–375*, 15–20. [CrossRef]
- Siddhpura, A.; Paurobally, R. A review of flank wear prediction methods for tool condition monitoring in a turning process. *Int. J. Adv. Manuf. Technol.* **2013**, *65*, 371–393. [CrossRef]
- Twardowski, P.; Tabaszewski, M.; Wiciak-Pikuła, M.; Felusiak-Czyryca, A. Identification of tool wear using acoustic emission signal and Machine Learning methods. *Precis. Eng.* **2021**, *72*, 738–744. [CrossRef]
- Gouarir, A.; Martínez-Arellano, G.; Terrazas, G.; Benardos, P.; Ratchev, S. In-process Tool Wear Prediction System Based on Machine Learning Techniques and Force Analysis. *Procedia CIRP* **2018**, *77*, 501–504. [CrossRef]
- Neslušan, M.; Turek, S.; Brychta, J.; Čep, R.; Tabaček, M. Experimental methods in splinter machining. *EDIS ŽU Žilina* **2007**, *343*. Available online: <https://scholar.google.com/citations?user=ipt-r1qaaaaj&hl=de&oi=sra> (accessed on 15 February 2023).
- Wang, G.F.; Yang, Y.W.; Zhang, Y.C.; Xie, Q.L. Vibration sensor based tool condition monitoring using v support vector machine and locality preserving projection. *Sens. Actuators A Phys.* **2014**, *209*, 24–32. [CrossRef]
- Bergs, T.; Holst, C.; Gupta, P.; Augspurger, T. Digital image processing with deep learning for automated cutting tool wear detection. *Procedia Manuf.* **2020**, *48*, 947–958. [CrossRef]
- 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*, 161–168. [CrossRef]

24. Zhou, Y.; Sun, W. Tool Wear Condition Monitoring in Milling Process Based on Current Sensors. *IEEE Access* **2020**, *8*, 95491–95502. [[CrossRef](#)]
25. Walther, M. Antriebsbasierte Zustandsdiagnose von Vorschubantrieben. Ph.D. Thesis, University of Stuttgart, Stuttgart, Germany, 2011. Heimsheim: Jost-Jetter (ISW/IPA Forschung und Praxis, 183).
26. Han, Y.; Song, Y.H. Condition monitoring techniques for electrical equipment—a literature survey. *IEEE Trans. Power Deliv.* **2003**, *18*, 4–13. [[CrossRef](#)]
27. Corne, B.; Knockaert, J.; Desmet, J. Misalignment and unbalance fault severity estimation using stator current measurements. In Proceedings of the IEEE 11th International Symposium on Diagnostics for Electrical Machines, Power Electronics and Drives (SDEMPED), Tinos, Greece, 29 August–1 September 2017.
28. Nguyen, T.L.; Ro, S.-K.; Park, J.-K. Study of ball screw system preload monitoring during operation based on the motor current and screw-nut vibration. *Mech. Syst. Signal Process.* **2019**, *131*, 18–32. [[CrossRef](#)]
29. Jamshidi, M.; Chatelain, J.-F.; Rimpault, X.; Balazinski, M. Tool condition monitoring based on the fractal analysis of current and cutting force signals during CFRP trimming. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 8127–8142. [[CrossRef](#)]
30. Czichos, H. *Tribologie-Handbuch. Tribometrie, Tribomaterialien, Tribotechnik*; überarbeitete und erweiterte Auflage; Vieweg+Teubner: Wiesbaden, Germany, 2010.
31. Zhao, J.; Lin, M.; Song, X.; Guo, Q. Analysis of the precision sustainability of the preload double-nut ball screw with consideration of the raceway wear. *Proc. Inst. Mech. Eng. Part J J. Eng. Tribol.* **2020**, *234*, 1530–1546. [[CrossRef](#)]
32. Sato, R. Wear Estimation of Ball Screw and Support Bearing Based on Servo Signals in Feed Drive System. In Proceedings of the International Conference on Leading Edge Manufacturing in 21st Century, Nagoya, Japan, 19–22 October 2011; Volume 6, p. \_3233-1\_.
33. Liu, X.; Mao, X.; He, Y.; Liu, H.; Fan, W.; Li, B. A new approach to identify the ball screw wear based on feed motor current. In Proceedings of the International Conference on Artificial Intelligence and Robotics and the International Conference on Automation, Control and Robotics Engineering, Kitakyushu, Japan, 13–15 July 2016; ACM: New York, NY, USA, 2016.
34. Yang, Q.; Li, X.; Wang, Y.; Ainapure, A.; Lee, J. Fault Diagnosis of Ball Screw in Industrial Robots Using Non-Stationary Motor Current Signals. *Procedia Manuf.* **2020**, *48*, 1102–1108. [[CrossRef](#)]
35. Kritzinger, W.; Karner, M.; Traar, G.; Henjes, J.; Sihn, W. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* **2018**, *51*, 1016–1022. [[CrossRef](#)]
36. Bibow, P.; Dalibor, M.; Hopmann, C.; Mainz, B.; Rumpe, B.; Schmalzing, D. Model-Driven Development of a Digital Twin for Injection Molding. In Proceedings of the International Conference on Advanced Information Systems Engineering, Grenoble, France, 8–12 June 2020; Springer: Cham, Switzerland, 2020; pp. 85–100. Available online: [https://link.springer.com/chapter/10.1007/978-3-030-49435-3\\_6](https://link.springer.com/chapter/10.1007/978-3-030-49435-3_6) (accessed on 1 March 2023).
37. Bolender, T.; Burvenich, G.; Dalibor, M.; Rumpe, B.; Wortmann, A. Self-Adaptive Manufacturing with Digital Twins. In Proceedings of the International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), Madrid, Spain, 18–24 May 2021.
38. Hribernik, K.; Cabri, G.; Mandreoli, F.; Mentzas, G. Autonomous, context-aware, adaptive Digital Twins—State of the art and roadmap. *Comput. Ind.* **2021**, *133*, 103508. [[CrossRef](#)]
39. Uhlemann, T.H.-J.; Schock, C.; Lehmann, C.; Freiburger, S.; Steinhilper, R. The Digital Twin: Demonstrating the Potential of Real Time Data Acquisition in Production Systems. *Procedia Manuf.* **2017**, *9*, 113–120. [[CrossRef](#)]
40. Costantini, A.; Di Modica, G.; Ahouangonou, J.C.; Duma, D.C.; Martelli, B.; Galletti, M. IoTwins: Toward Implementation of Distributed Digital Twins in Industry 4.0 Settings. *Computers* **2022**, *11*, 67. [[CrossRef](#)]
41. D’Agostino, D.; Morganti, L.; Corni, E.; Cesini, D.; Merelli, I. Combining Edge and Cloud computing for low-power, cost-effective metagenomics analysis. *Future Gener. Comput. Syst.* **2019**, *90*, 79–85. [[CrossRef](#)]
42. Asim, M.; Wang, Y.; Wang, K.; Huang, P.-Q. A Review on Computational Intelligence Techniques in Cloud and Edge Computing. *IEEE Trans. Emerg. Top. Comput. Intell.* **2020**, *4*, 742–763. [[CrossRef](#)]
43. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [[CrossRef](#)]
44. Hu, L.; Nguyen, N.-T.; Tao, W.; Leu, M.C.; Liu, X.F.; Shahriar, M.R.; Al Sunny, S.M.N. Modeling of Cloud-Based Digital Twins for Smart Manufacturing with MT Connect. *Procedia Manuf.* **2018**, *26*, 1193–1203. [[CrossRef](#)]
45. Jiang, Y.; Yin, S.; Li, K.; Luo, H.; Kaynak, O. Industrial applications of Digital Twins. *Philos. Trans. Ser. A Math. Phys. Eng. Sci.* **2021**, *379*, 20200360. [[CrossRef](#)] [[PubMed](#)]
46. Salgado, D.R.; Alonso, F.J. An approach based on current and sound signals for in-process tool wear monitoring. *Int. J. Mach. Tools Manuf.* **2007**, *47*, 2140–2152. [[CrossRef](#)]
47. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN Revisited. *ACM Trans. Database Syst.* **2017**, *42*, 19. [[CrossRef](#)]
48. Zhang, C.; Liu, J.; Chen, W.; Shi, J.; Yao, M.; Yan, X.; Xu, N.; Chen, D. Unsupervised anomaly detection based on deep autoencoding and clustering. *Secur. Commun. Netw.* **2021**, *2021*, 7389943. [[CrossRef](#)]

49. Netzer, M. Intelligente Anomalieerkennung für Hochflexible Produktionsmaschinen: Prozessüberwachung in der Brownfield Produktion. Ph.D. Thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany, 2022.
50. Zhang, L.; Gao, H.; Dong, D.; Fu, G.; Liu, Q. Wear Calculation-Based Degradation Analysis and Modeling for Remaining Useful Life Prediction of Ball Screw. *Math. Probl. Eng.* **2018**, *2018*, 2969854. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.