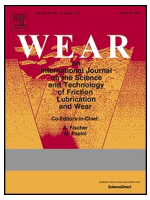
Wear 582-583 (2025) 206348

Contents lists available at ScienceDirect

Wear

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



A novel data concept for cutting processes through comprehensive experimental setup enabling grey-box models

Sebastian Schibsdat [a,*], Ya-Jing Wu [b], Martin Keunecke [c], Sarah Baron [c], Daniel Höche [b], Jens-Peter Zemke [d], Sebastian Götschel [e], Christoph Herrmann [c], Jan Hendrik Dege [a]

[a] Hamburg University of Technology, Institute of Production Management and Technology, Denickestraße 15, 21073, Hamburg, Germany
[b] Helmholtz-Zentrum Hereon, Max-Planck-Straße 1, 21502, Geesthacht, Germany
[c] Fraunhofer-Institute for Surface Engineering and Thin Films IST, Riedenkamp 2, 38108, Braunschweig, Germany
[d] Numerical Mathematics, Institute of Mathematics, Hamburg University of Technology, 21073, Hamburg, Germany
[e] Computational Mathematics, Institute of Mathematics, Hamburg University of Technology, 21073, Hamburg, Germany

A B S T R A C T

This paper presents a novel experimental setup for recording the cutting process of external longitudinal turning in situ, capturing data such as cutting force, vibration, acoustic emission (AE), rake face temperature, surface quality, and tool wear. The primary objective of this setup is to generate reliable, repeatable data for developing a comprehensive grey-box model. To achieve this, the measurements were partly automated. Additionally, a complete post-processing pipeline has been introduced to combine all relevant data. To validate the setup, external longitudinal turning experiments were conducted and two of these were analysed for sensitivity to tool wear. This was achieved by analysing the mean and variance of the resultant force F_Z, resultant vibrations a_Z and AE signal, in order to demonstrate the impact of tool wear on the measurement. The experiments were also used to train an autoencoder to analyse the process data.

1. Introduction

Machining is a fundamental manufacturing process that involves shaping materials into complex components with great precision. Despite significant technological advancements in tool coating and cutting edge geometry, tool wear remains a persistent challenge in machining. The prediction of wear on uncoated tools has been extensively researched in the past to optimise tool life and surface quality. However, the high mechanical and thermal loads mean that uncoated tools no longer achieve an acceptable service life. To enhance tool life and performance, coatings have been developed. Protection against abrasive wear, increased thermal and chemical resistance, and reduced friction are all offered by these. This allows them to reach longer tool life in modern applications. Predicting the wear progression of coated tools is especially difficult. Models have been developed to monitor and predict tool life under various specific conditions [1–4]. However, reliably predicting wear progression and tool life specifically with analytical models (white-box) remains challenging. This is due to the multi-layered process-dependent effects that appear during the cutting process and which are hard to model analytical due to the high complexity.

In recent years, machine learning (ML) techniques have emerged as a promising tool for predicting tool wear, thanks to their ability to model complex, non-linear relationships from data [5–9]. However, this research has often been based on small datasets created using a limited number of sensors [10]. As these datasets have not been published, further research requires the acquisition of new datasets, making comparison and development of machine learning algorithms for tool wear prediction more difficult.

Moreover, while powerful within the bounds of their training data, purely data-driven ML models (black-box) lack interpretability and often generalisability. To address these limitations, grey-box models, which combine physical knowledge with data-driven models, have gained attention [11,12]. These models aim to improve robustness and extrapolation capability while providing insights into the underlying physical mechanisms.

This paper presents a novel experimental setup that records the cutting process in external longitudinal turning in situ, including cutting

This article is part of a special issue entitled: Grey Box Wear Prediction_invited only published in Wear.

* Corresponding author.
E-mail addresses: sebastian.schibsdat@tuhh.de (S. Schibsdat), ya-jing.wu@hereon.de (Y.-J. Wu), martin.keunecke@ist.fraunhofer.de (M. Keunecke), sarah.baron@ist.fraunhofer.de (S. Baron), daniel.hoeche@hereon.de (D. Höche), zemke@tuhh.de (J.-P. Zemke), sebastian.goetschel@tuhh.de (S. Götschel), Christoph.Herrmann@ist.fraunhofer.de (C. Herrmann), jan.dege@tuhh.de (J.H. Dege).

https://doi.org/10.1016/j.wear.2025.206348
Received 30 May 2025; Received in revised form 12 September 2025; Accepted 16 September 2025
Available online 17 September 2025

Table 1
Process parameter range.

Process Parameter	Symbol	Values
Cutting Depth	a_p	1.2–1.6 mm
Cutting Speed	v_c	80–320 m/min
Cutting Feed	f	0.15–0.35 mm
Coating Thickness	$t_{Coating}$	3.2–5.4 µm
Coating Hardness	$HV_{Coating}$	HV 2100
Workpiece Material		C45 (1.0503) + N, X5CrNi18-10 (1.4301)

force, vibration, acoustic emission (AE), rake face temperature, surface quality, and tool wear. The aim is to enable the reliable generation of repeatable data in order to create a comprehensive grey box model. To validate the setup, some external longitudinal turning experiments were carried out. Two of these experiments are then investigated with regard to their sensitivity to tool wear.

2. Experimental setup

The planned experiments are designed to be conducted in a series of repeated timed cutting intervals, with each interval lasting between t_c = 30 s. Tool failure will be defined as reaching a maximum flank wear width of VB_{max} = 300 µm, tool breakage, or a significant change in chip formation. Each cutting edge will be tested under fixed cutting parameters. The setup aims to vary cutting depth a_p, cutting speed v_c and feed rate f, as well as coating thickness $t_{Coating}$ and hardness $HV_{Coating}$. The workpiece material will also vary. Table 1 outlines the proposed variations to these process parameters.

The planned machining process involves repeatedly performing external cylindrical turning on the shafts up to a length of l_S = 500 mm. The shafts start with a diameter of d_S = 150 mm and will be used until a diameter of d_S = 80 mm is reached. Additionally, once a shaft is mounted in the lathe, an initial turning pass will be performed to compensate for any misalignment.

2.1. Cemented carbide cutting insert and TiAlN cutting tool coatings

As tools cemented carbides cutting inserts, fine grained tungsten carbide (WC grain sizes ≤1 µm) with 10 wt% Co binder and a geometry CNMG160608-MM5 were used. In Fig. 1 the used cemented carbide cutting inserts are displayed using light microscopy. A relative complex shape and a rough surface topography of the cutting edge is visible.

For a wear resistant tool coating well investigated and established TiAlN coatings deposited with flexible PVD technologies was selected. to have the possibility to modify and adapt coating properties and structure to requirements [13–15]. The PVD TiAlN coatings were prepared using a CemeCon CC800/9 sinox ML sputtering system. The process chamber has a volume of about 0.8 m^3 and is equipped with magnetron sources (dimensions 500 mm × 88 mm). The residual pressure before starting the process was below 10^{-3} Pa. Before deposition the PVD TiAlN coating the substrates were cleaned using an industrial ultrasonically sustained water-based cleaning line for cemented carbide substrates. After cleaning the tools are transferred to the coating machine and mounted at sample holders like shown in Fig. 2.

The deposition processes for TiAlN coatings consisted of four main steps: (I) heating (II) Ar ion etching for substrate cleaning; (III) sputter deposition by increasing of the reactive gas flow gradually from zero to approximately 200 sccm for TiAlN coatings and increasing the target power up to maximum values and gradually increasing the substrate bias starting from −30 V d.c. up to −60 V d.c.; and finally (IV) the deposition of the TiAlN coating under constant process conditions. The deposition rate is around 1 µm/h. The main coating parameters are summarized as.

- Targets: four TiAl (50:50 composition) targets
- Target power: TiAl 4 kW, pulsed d.c. (50 kHz, duty cycle 50 %)

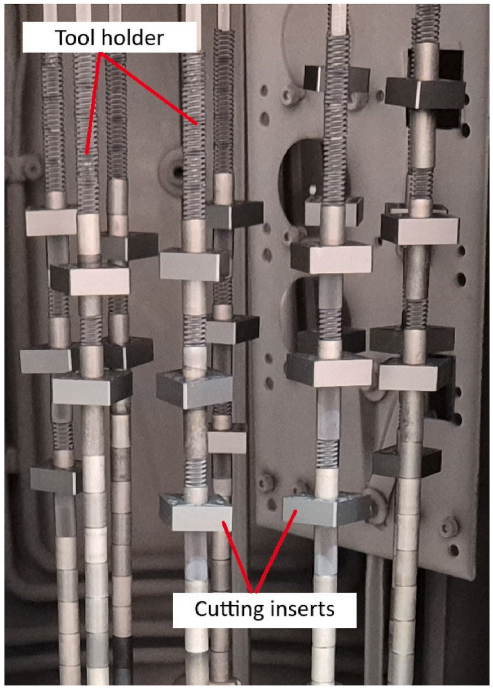


Fig. 2. Cutting inserts and samples for characterization inside the vacuum chamber.

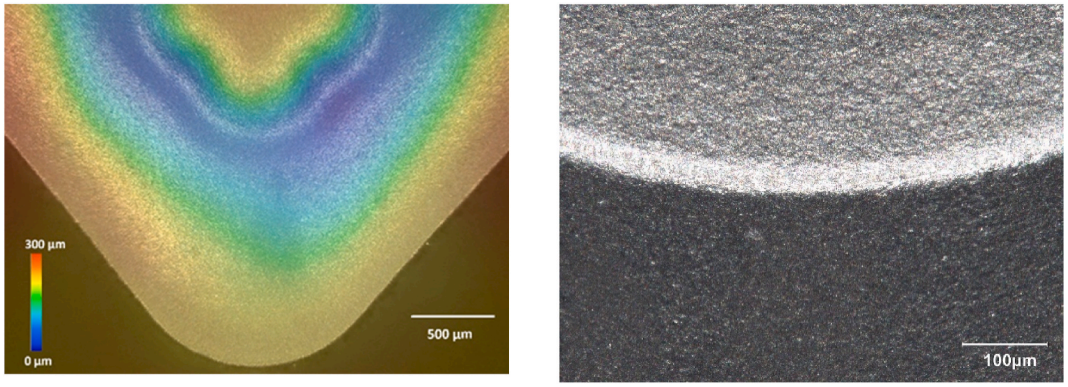


Fig. 1. 3D light microscope images of an uncoated cemented carbide cutting insert. left:100x magnification with colors indicated height profile, right: uncoated cutting edge with 500x magnification. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

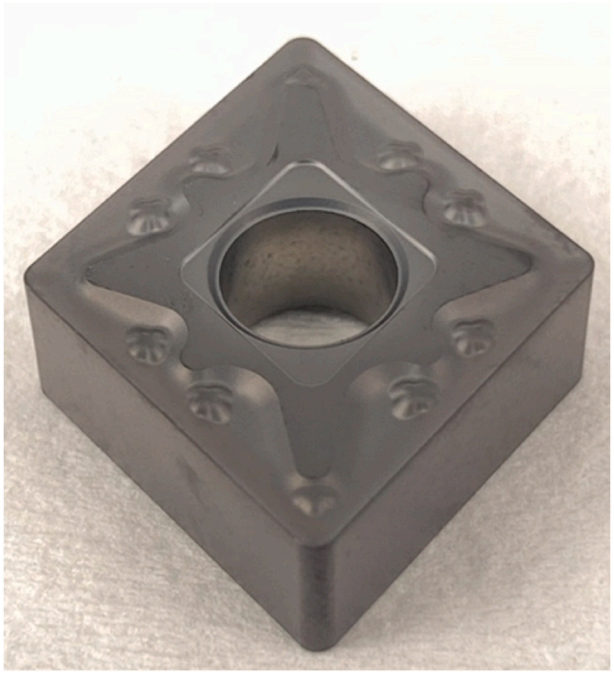


Fig. 3. Photo of a TiAlN coated cutting insert.

- Reactive gas: nitrogen (N2); Gas flows: Ar: 200–300 sccm; N2: 200 sccm
- Process pressure: approximately 0.4 Pa
- process temperature approx. 250 °C
- Substrate rotation: 1 rpm of planetary and additionally two-fold rotation on satellites. This means a three-fold rotation of cemented carbide cutting inserts

After the coating process the coated tools and substrates were demounted. As substrate material for characterization polished steel substrates, e.g., HSS (1.3343) as well as pieces of Si wafers and the above-described cemented carbide cutting inserts were used. The substrates were fixed at sample holders and coated in two-fold rotation. Fig. 2 shows an image of different samples inside the process chamber. A photo of a TiAlN coated cemented carbide insert is displayed in Fig. 3.

The indentation hardness and the indentation modulus of the coatings on polished HSS (1.3343) substrates were measured by a Fischerscope H100 commercial instrument. The hardness and indentation modulus were derived from load - indentation depth curves. The indentation depth was around 300 nm. Hardness measurements of TiAlN coatings on different substrates were performed. The rough surface of the cemented carbide cutting inserts significantly affects the hardness measurement and lead to a large variance of and consequently makes a determination of coating hardness inaccurate. In contrast the coated polished steel substrate. The reproducible run of load-indentation curves allows a precise measurement of coating hardness. From load indentation curves different hardness values were derived. For the TiAlN coating on polished HSS substrates Vickers hardness values around 2100 HV were determined, which means only a moderate hardness for TiAlN coatings but a significant higher hardness compared to uncoated cemented carbides with measured Vickers hardness of around 1580 HV.

Table 2
Hardness and mechanical properties of TiAlN coating on polished steel.

TiAlN coating properties	Values	Unit
Martens hardness HM	11.6 ± 0.7	GPa
Plastical hardness HU_{pl}	26 ± 2	GPa
Plastical Vickers hardness HV_{pl}	2200 ± 190	HV_{pl} 0.005
Vickers hardness HV	2100 ± 168	HV 0.005
Indentation modulus EIT	265 ± 12	GPa
Abrasive wear rate	3.1 ± 0.4	10^{-15} m^3/Nm

Abrasive wear rates of the coatings were determined with the ball cratering test operating with an alumina suspension (alumina glycerin suspension, 25 % by weight Al_2O_3, particle size 1 µm). To quantify the results, the volume of the crater ground into the coating was divided by the normal load and the track length of the rotating ball. With a value of 3.1 10^{-15} m^3/Nm for TiAlN coating the resistance against abrasive wear is more than 10 times higher as for uncoated tool substrates. The coating adhesion was determined by Rockwell (HRC) indenter tests. After DIN 4856 - 2018-02 an adhesion class of two for the TiAlN coatings was detected and thus a good coating adhesion to the substrate was generated. Furthermore, the coating quality at the cutting insert as well as at the cutting edge was verified by light microscopy. The TiAlN coating was deposited with a thickness of 3.5 µm ± 0,3 µm onto cutting inserts. Microscope images of a TiAlN coated cutting edge are displayed in Fig. 4. The cutting edge topography of a TiAlN coated cutting insert is comparable to an uncoated cutting edge (Fig. 1). In combination with coating thickness the roundness of the cutting edge slightly increases. The radius was measured by light microscopic investigation. The uncoated cemented carbide cutting insert exhibit a cutting-edge radius in the range of 38 µm whereas TiAlN coated cutting inserts exhibit cutting-edge radius values in the range of 42 µm. The coating quality and properties (see Table 2) are promising and are suitable for cutting experiments and further investigations.

2.2. Machine setup

The experiments were carried out on a Max Mueller Gildemeister

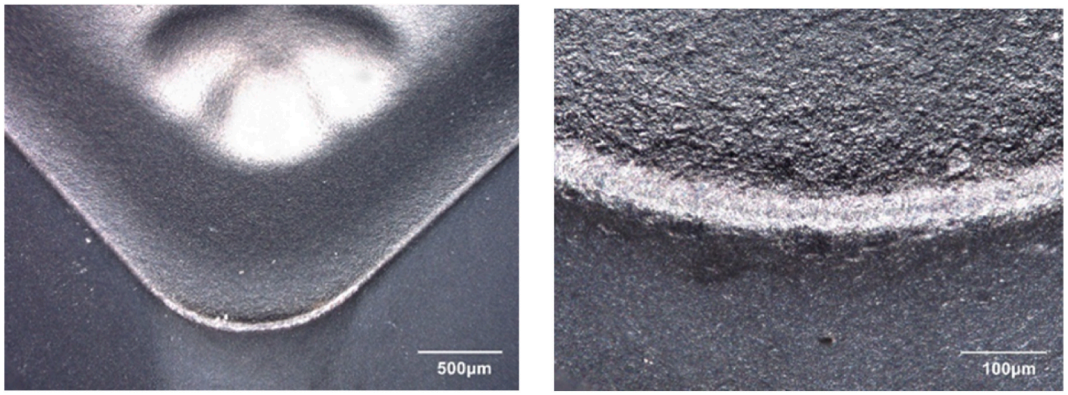


Fig. 4. Light microscope images of a TiAlN coated insert cutting edge; left: 100x magnification; right: 500x magnification.

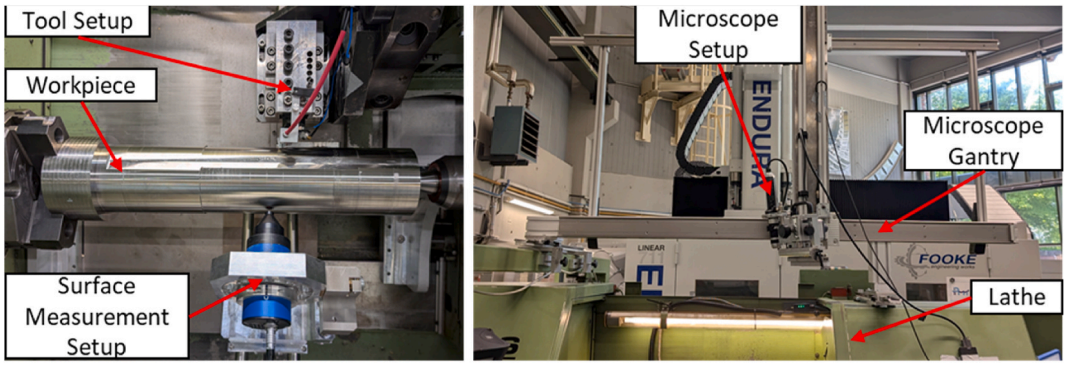


Fig. 5. Overview of the machine metup (left: inside the lathe chamber, right: on top the lathe).

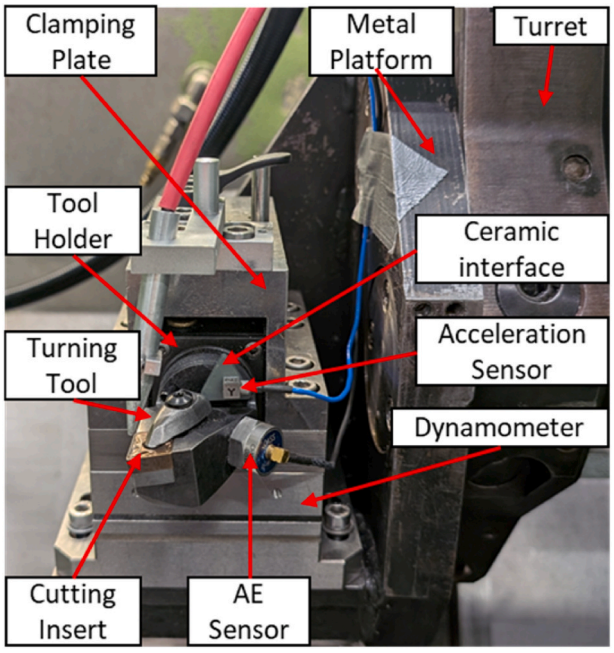


Fig. 6. Setup of tool, dynamometer, acceleration sensor and AE sensor.

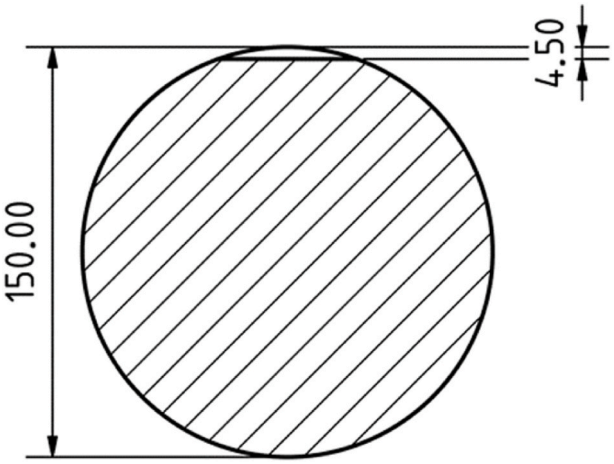


Fig. 7. Flattened shaft at starting diameter $d_S = 150$ mm.

MD5S CNC lathe using CNMG160608-MM5 indexable inserts with a custom TiAlN coating. The Walter C4-DCLNL-27055-16 turning tool was used in conjunction with the Walter C4-LC2085-5048 tool holder. This

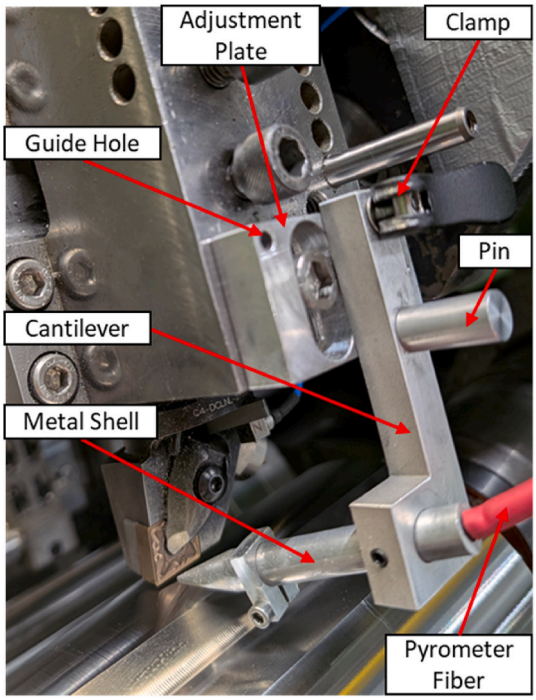


Fig. 8. Setup of the temperature measurement.

setup produced the following cutting parameters: tool orthogonal clearance $\alpha_0 = 16°$, tool orthogonal rake angle $\gamma_0 = -6°$, wedge angle $\beta_0 = 80°$, tool cutting edge inclination $\lambda_s = -6°$ and corner radius $r_\varepsilon = 0.8$ mm.

The workpieces were clamped in the lathe using a three-jaw chuck and a tailstock. Fig. 5 provides an overview of the entire setup, while Fig. 6 shows a detailed depiction of how the tool is setup.

To measure the rake face temperature in situ with the 2-colour pyrometer, an interruption to the turning process is necessary. To achieve this, the workpieces are flattened. Initially, the flattening is $t = 4.5$ mm deep and will be machined again if it does not interrupt the next external longitudinal rotation. To minimise the impact of interrupted cutting on tool wear, flattening is applied across the entire secant line width, as illustrated in Fig. 7.

2.2.1. Force measurement, acceleration and acoustic emission

To measure the cutting forces a 3-component dynamometer type 9257B, Kistler Instrumente GmbH, is mounted between the tool holder and the machine turret, Fig. 6. To achieve this, a metal platform is directly fixed to the turret. The design of the metal platform allows still the rotation of the turret to move the cutting tool out of the cutting

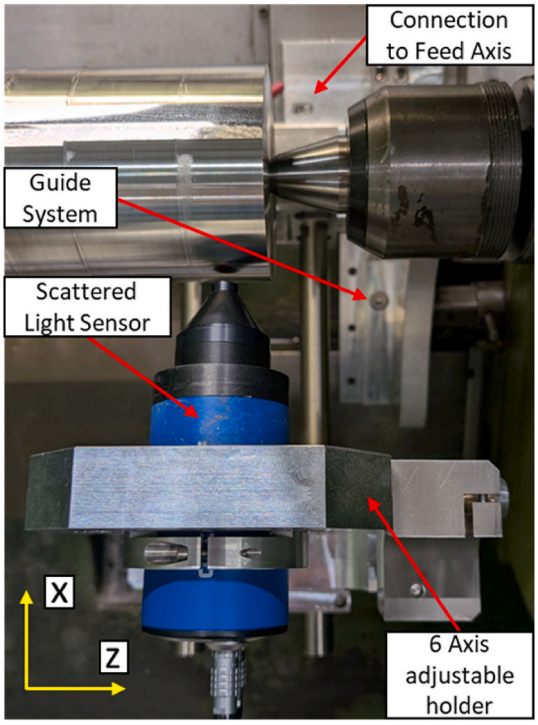


Fig. 9. Setup of the workpiece surface measurement.

plane. To fix the tool holder on to the dynamometer a clamp is used. The acceleration sensor W356A03 NC, PCB Piezotronics, is glued to the turning tool. This is not done directly, because conductivity of the setup introduced noise to the acceleration measurement. To combat this, a ceramic isolator was used as an interface to the turning tool. The AE Sensor of the Optimizer4D measurement system, QASS GmbH, is directly glued to the turning tool.

2.2.2. Temperature measurement

To measure the temperature of the rake face in the interrupted part of the cutting process, a two-colour pyrometer Fire-3, en2Aix GmbH, is used, Fig. 8. The measuring unit is placed outside the machine. An optical fibre is used to direct the light emitted from the rake face to the sensor. The optical fibre is fixed to the top of the clamping plate for the tool holder using an adjustment plate and a cantilever. The adjustment plate enables fine alignment of the fibre optic with the rake surface to be achieved. The cantilever can be used to rotate the fibre optic out of the way when the cutting edge needs to be changed or images of tool wear need to be taken. To reliably return the fibre optic to the measurement position, an indexation and a clamp has been introduced. To increase the measurement accuracy, the fibre optic must be as close as possible to the surface. In this case, a distance of $d_{Fibre} = 7$ mm was achieved. To protect the fibre from chips, a metal shell was placed around it.

2.2.3. Workpiece surface measurement

The surface measurement is realised using a scattered light sensor OS 500, OptoSurf GmbH, which measures the surface along the direction of rotation. To produce comparable, high-resolution measurements for different cutting parameters, the workpiece surface is measured in a second pass over the freshly turned surface, at a constant rotation velocity for all experiments. To achieve this the measurement will be done by a fixed rotation velocity of $v_{OS} = 76$ m/min.

To enable movement of the scattered light sensor, a guide system is installed on the tailstock's guide rails, Fig. 9. The sled on the guide rails is connected to the lathe's feed axis so that it moves in z-direction at a fixed distance from the workpiece. The sensor itself is placed inside a six-axis adjustable holder to enable perpendicular alignment of the sensor with the workpiece, as well as setting the gap of $d_{OS} = 5$ mm between the sensor and the shaft for varying shaft diameters d_S.

2.2.4. Wear measurement

To measure tool wear without removing the tool from the holder, a gantry with a microscope VHX-7020, Keyence Deutschland GmbH, and an adjustable magnification lens VH-Z20T, Keyence Deutschland GmbH, is placed on top of the lathe, as shown in Fig. 5. The microscope can be lowered into the processing area, where it is fixed in place to take pictures The gantry is necessary to be able to lift the heavy shafts into the machining area using a crane. A hinge connecting the camera, motor, and portal allows the camera to rotate up to 90° around the cutting edge

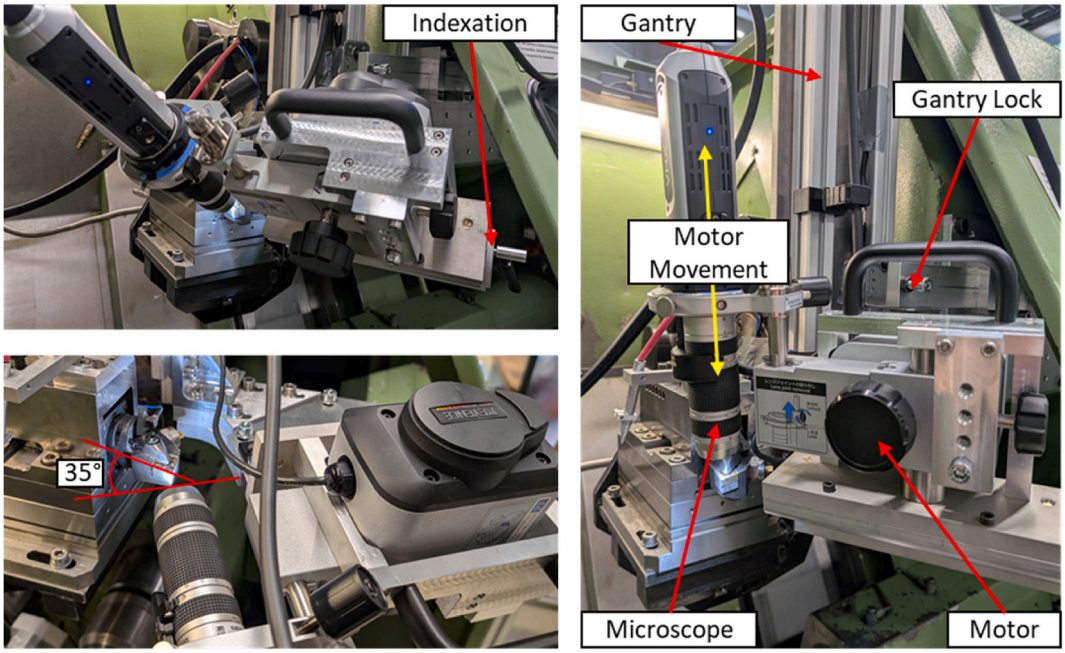


Fig. 10. Microscope setup at different angels (top left: aligned 45° between rake face and flank face, bottom left: aligned to flank face, right: aligned to rake face).

of the tool, enabling images of the rake and flank faces to be captured, as shown in Fig. 10. The hinge is fixed at a 35° angle to the primary cutting edge. This ensures that both the primary and secondary cutting edges are visible when images of the free flank are taken. To improve repeatability, an indexation is used to fix the camera at the different angles. The camera motor VHX-S90F, Keyence Deutschland GmbH, in combination with the microscope controller VHX-X1F, Keyence Deutschland GmbH, enables the tool to be brought into focus, as well as enabling focus variation for greater depth of field and 3D surface mapping.

Between cutting experiment intervals, a microscope image with 150x magnification is taken of the rake face, flank face and the area at a 45° angle between them. Additionally, an image of the flank face is captured at a lower magnification of 100x. This makes it easier for the evaluation to identify the original cutting edge in its worn state. Images are taken with focus variation to ensure that large areas of the tool are in focus, despite the curvature and angle of the surface. A height map of the surface is also captured using focus variation. To reduce reflections, the microscope takes the same image multiple times with different lighting settings and then combines the best parts of each to create the final image.

2.3. Sensor setup

The signal from the 3-component dynamometer is amplified using a 3-channel signal amplifier 5019B130, Kistler Instrumente AG. The three amplified signals are connected to an analogue-digital converter (ADC) NI-9215, National Instruments Corporation. The ADC is plugged in an Ethernet connector chassis cDAQ-9184, National Instruments Corporation. Additionally, an Arduino Uno R3, Arduino SA, 5V Digital Output is connected to the fourth input to the ADC NI-9215. From here on out, the Arduino Uno R3 will be referred to as the 'Uno Trigger'. The 2-colour pyrometer has two outputs, which are already amplified by the measurement system itself. These are connected directly connected to a second ADC NI-9215, National Instruments Corporation, module, which is also plugged into the Ethernet connector chassis. The position signal from the rotary axis of the lathe is accessed via a self-made signal splitter directly from the rotation encoder inside the machine. To reduce the frequency required for capturing the position signal, a ripple-carry binary counter CD4024B CMOS, Texas Instruments, is employed to divide the TTL signal by four. Both the reduced and original signal of the encoder are connected to the second ADC module. The accelerations sensor is connected to a ADC NI-9234, National Instruments Corporation, module. This ADC supports integrated electronic piezoelelectric (IEPE), which is required by the acceleration sensor. The ADC NI-9234 is also plugged into the Ethernet connector chassis. The Ethernet connector chassis establishes synchronisation between the different ADC modules connected to it. As measure frequency $f_{NI} = 51.2$ kHz was selected, which is the maximum frequency supported by the ADC NI-9234 module. The Ethernet connector chassis is accessed via the local network using a computer. The connection is established with the software NI Max, while data acquisition is done with the software MATLAB R2024b. The scattered light sensor is directly connected to the computer. To access the sensor and save the measurement, the manufacturer software for the sensor was used. The measure frequency is determined through an external trigger signal. For this the Uno Trigger 5V digital output connected to the ADC module was spliced and connected to the scattered light sensor. The digital output produces a 5 V TTL signal with the a frequency $f_{SL} = 2$ kHz. The Optimzer4D measurement system, which measures the AE, is a closed system, that uses it's own amplifier and computer. Here, the measurement frequency f_{AE} was set to 200 kHz. This frequency was selected as preliminary tests showed no significant effects above 100 kHz.

To automate the experiment as much as possible, a second Arduino Uno R3, Arduino SA, is used, which will be referred to as the 'Uno Controller' from here on out. A mechanical switch is connected to the Uno Controller, which allows to change the state of a digital input channel. The Uno Controller is connected to the computer and directly

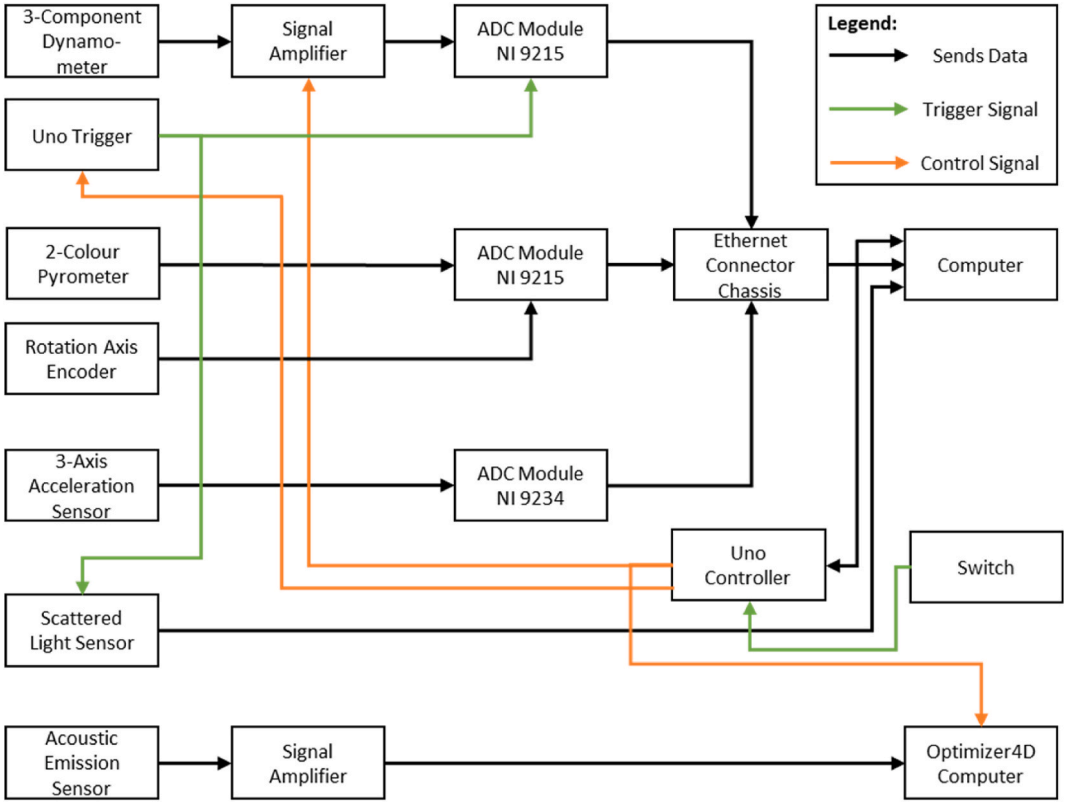


Fig. 11. Schematic representation of the installed sensors and measuring devices.

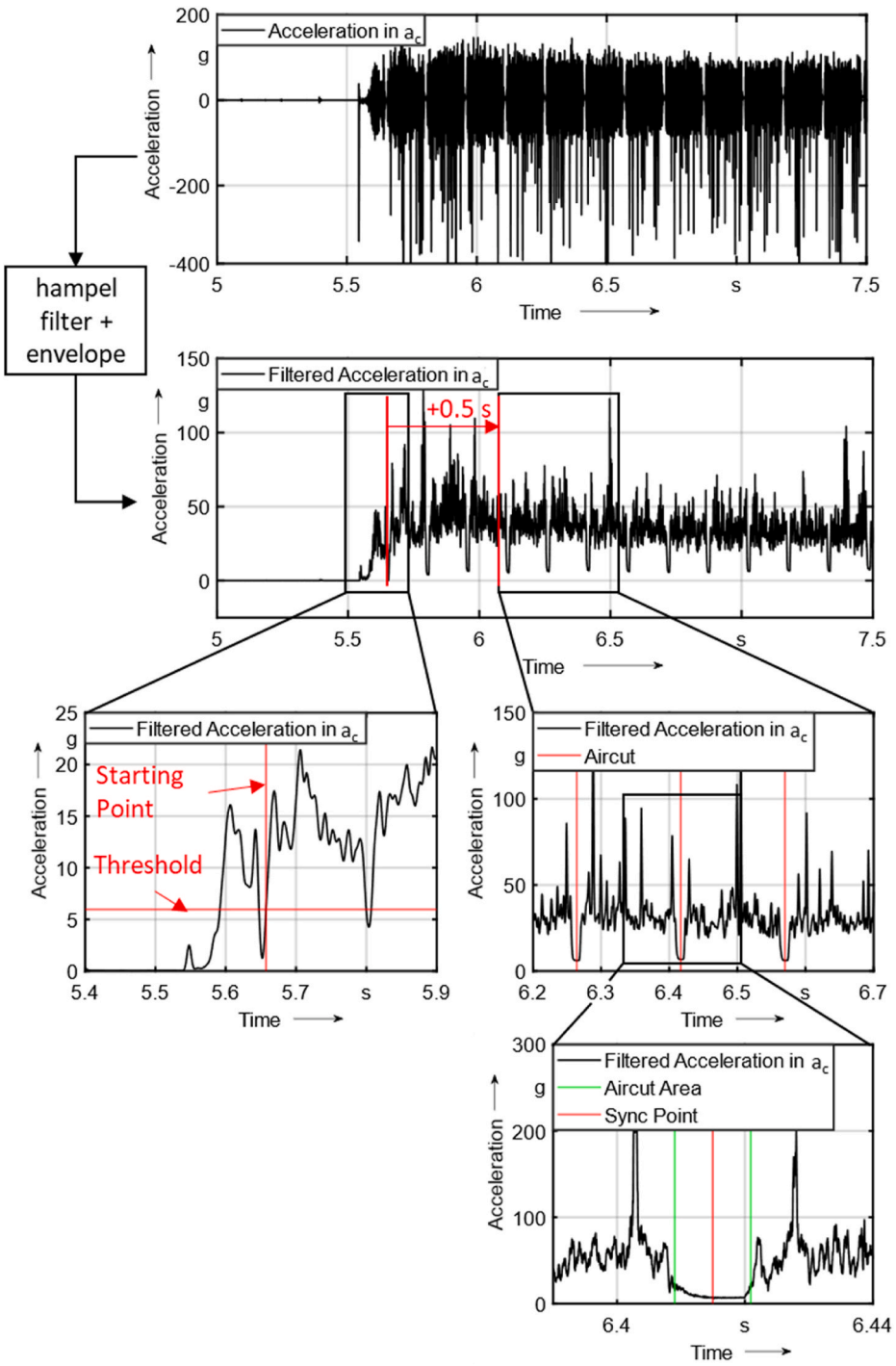


Fig. 12. Visualization of the synchronisation process using the acceleration sensor (Complete cutting time at start: 90 s; process parameters: $a_p = 1.6$ mm, $v_c = 170$ m/min, $f = 0.35$ mm).

controlled by the Software Matlab Simulink R2024b. When the software detects that the state of the digital input channel changed, it sends a trigger signal via the Uno Controller digital output channel to start the Optimzer4D measurement system and the 3-channel signal amplifier. Additionally, it starts the recording of data from the Ethernet connector chassis. When the state of the digital input channel changes back, the trigger signal is dropped and recording stops. The recorded data from the Ethernet connector chassis is then saved. For the measurements using the scatter light sensor, the switch is activated a second time. In this case a different digital output is used to send a trigger signal which activates the Uno Trigger. A recording of the Ethernet connector chassis is also started, using only the trigger signal from the Uno Trigger and the rotary axis position signal from the NI cDAQ. To capture the surface scatter sensor signal, the measurement must be set to 'ready' manually in the device software, after which recording will start upon detection of the Uno Trigger's trigger signal. After switching back a second time, the trigger signal is dropped again and the recorded data from the Ethernet connector chassis is saved.

3. Data postprocessing

In order to be used for modelling later on, the data has to be

formatted and processed so that it can be accessed uniformly. Following a cutting experiment, four types of data file are created. One is created by Matlab (.mat) and contains force, acceleration, temperature, and rotary axis encoder signals. A second file (.txt) is created by scattered light sensor software and contains reflection intensity data of the sensor from 16 different angles. Another Matlab file (.mat) is also created in addition, which includes the Uno Trigger signal and the rotary axis encoder signal. The Optimzer4D measurement system can be used to export the AE measurement (.bin). Finally, the camera images are saved in JPEG format and the height information is extracted into an file (.csv) using a Keyence exporter script.

The first step in the post processing stage is to merge the data from all the files. Next, the data must be synchronised so that it can be interpreted simultaneously. Finally, additional values such as the number of rotations or the area of air cutting can be calculated. Postprocessing is performed using the Matlab software.

The first step involves using custom import functions to read the different files and load them into the Matlab workspace. Next, the signals contained in the files must be synchronised. Initially, the acceleration sensor data is synchronised with the signals captured by the Ethernet connector chassis, as the ADC NI 9234 module introduces a time delay depending on the frequency f_{NI}.

As the AE sensor uses its own measurement system, it needs to be synchronised with the other sensors captured by the Ethernet connector chassis, such as those measuring cutting force, cutting temperature, tool vibration and workpiece rotation. As the acceleration sensor captures a signal most closely related to the AE sensor, it is selected for synchronisation. Specifically, the cutting acceleration a_c is used as comparison, because the largest amplitudes in the signal occur in this direction. The interruptions of the cut are used to synchronise the signals. These interruptions must occur at the same time in both signals and can therefore be used to align them. However, as they only appear once per rotation, simply shifting the signals to match is not enough to correct for an asynchrony of complete rotations. The process therefore begins by identifying the starting point of the acceleration a_c, which is done by looking for increasing values starting from the resting position of the signal around zero when cutting is initiated, Fig. 11. This signal is then processed using a Hampel filter on its envelope. This provides a rough estimate of the start time of the signal. Next, the exact start of the cutting process is determined. To do so, the previously estimated start point is used as a reference and a 1-s section of the filtered signal is selected and filtered using a moving average filter. Fig. 11 shows just a 0.5 s section of this step to improve the visibility of the signal. This section is then searched again for the exact start of the cutting process. This is achieved by identifying the first instance in which the signal surpasses a predetermined threshold within a specified time window. The threshold is calculated by taking half the mean value of the signal in the final third of the section. Now that the exact start of the cutting process has been established, the distinctive air-cutting features of the signal can be selected as a reference point for synchronisation. To achieve this, a small time window of the stable turning process 1 s after the starting point is used. The distance of 1 s to the start point ensures that the cutting process is already in a steady state and does not exhibit artefacts that would make identifying air-cut windows difficult. During this time window, the minima representing the air-cut areas are searched for. The second minimum found is then selected, after which a time window half the distance between the next minimum is applied. The Matlab function "findchangepts" then identifies the start and end of the air cut in this window by splitting the signal in this window in three parts, that have similar mean values. Because the signal in the while the air-cut is close to zero, as seen in Fig. 12, the algorithm can reliably find this area. The point in the middle is then used as the synchronisation point. These steps are repeated for the AE sensor. The two synchronisation points are then used to align the AE sensor data with that of the acceleration sensor by shifting the timesteps to match at the synchronisation point. As the dynamometer, temperature and rotation axis encoder signals are already aligned with the acceleration sensor, further alignment between them and the AE sensor is unnecessary.

The next step is to synchronise the data from the scattered light sensor. As this measurement is taken in a second process step of the experiment with a different rotation velocity v_{OS}, synchronisation is performed differently to that of the AE sensor. First, the measurement points of the scattered light sensor are matched to the trigger signal in the measurement of the Ethernet connector chassis, which was captured in parallel for the second step of the experiment. This measurement includes the rotation encoder signal, which is used to determine the position of the shaft. This measurement is also performed in the first step of the experiment. As the rotation encoder signal is captured in both steps of the experiment and the feed f and distance of the cut are the same, these can be used to synchronise the scattered light sensor. The idea is that the scattered light sensor measurement at a given point of rotation can be mapped to the same position in the cutting process. As the position of the absolute rotation encoder is captured relative to the starting point, a common point in the process first has to be identified, which is then enhanced by synchronising the areas of air cutting in the scattered light sensor. The stopping of the shaft at the end of the experiment is used as the common point, as this is the most concise point to identify in the rotation encoder signal. The position of the scattered light sensor is then adjusted to match the cutting process values at this point. The air-cut areas of the scattered light sensor are identified using the intensity change of the reflected light caused by the flattened area. The intensity reaches a maximum exactly in the middle of the flattened area, as the flat surface reflects a higher proportion of the emitted light than the curved surface of the shaft. For a specific rotation, before the

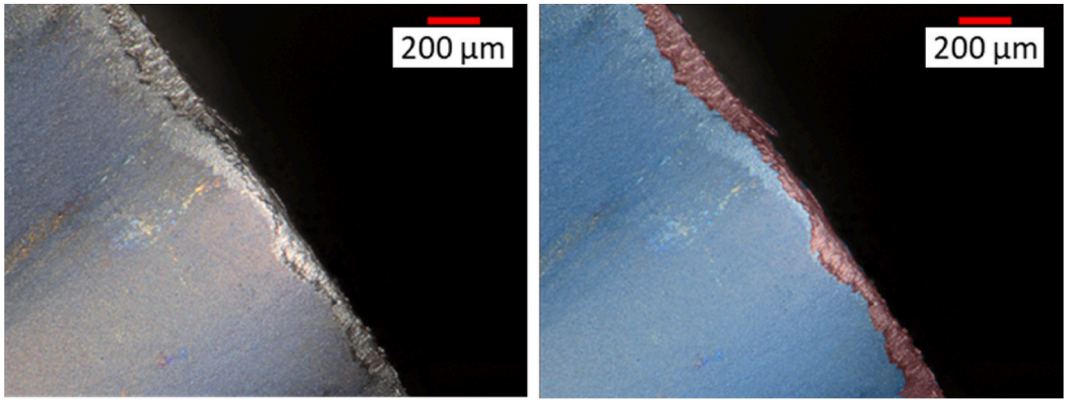


Fig. 13. Example of an annotated microscope image of the free flank (blue: visible tool, red: free Flank wear, Black: Background). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

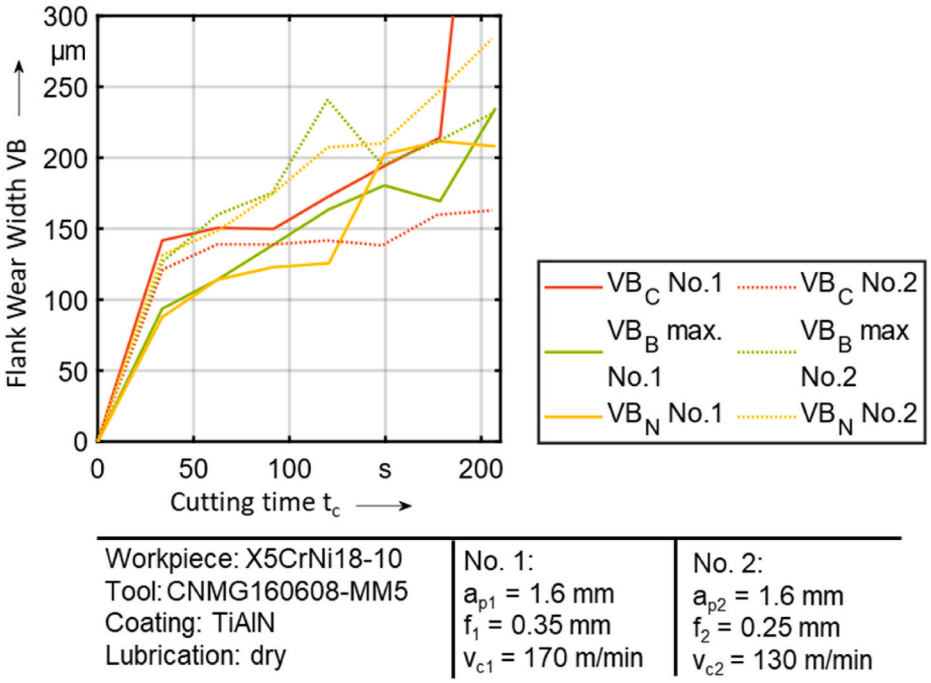


Fig. 14. Free flank tool wear progress.

end of the process, this maximum is aligned with the synchronisation point in the cutting acceleration a_c, which is found in the same way as described for synchronising the AE sensor for this rotation.

The synchronised signals (cutting force, tool vibration, temperature, AE, rotation of the workpiece) are then truncated so that the data includes up to 4 s before and after the cut, in order to identify noise levels in the sensors. If there is not enough time before or after the cut, this time period is reduced in 0.25 s increments until it fits.

As a final step, the collected data is enhanced. First time windows of cutting and air cutting are determined through identifying parts of the tool vibration which show no vibrations caused by the interruption of the cut. Additionally, the number of interruptions to the cut is counted to mark the revolutions. Next, the temperature signal is evaluated when the rake face first becomes visible to the fibre during an interruption to the cut. Finally, the intensities of the scattered light sensor are used to calculate the surface roughness (Aq) and surface form [16].

After processing each cut individually, the next step is to combine all cuts made using the same cutting edge. To achieve this, the first experiment of that particular cutting edge is loaded as the starting point. In this experiment, everything after the fifth last rotation at constant feed is cut. Dropping the last five rotation is a compromise between using as much of Signal as possible, without including parts of the cutting process that don't represent the constant cutting as the cutting process ends. All further experiments with this cutting edge are then loaded iteratively and cut so that the signals start after the fifth rotation and end after the last fifth rotation at constant feed. The final experiment retains the additional time window at the end. The time and rotation distance of every loaded experiment are shifted so that they fit with the previous experiment.

The images of the flank face captured by the microscope are semantically annotated. To achieve this, the 2D image of the flank is loaded into the 'Image Labeler' annotation software in Matlab. The image is then segmented into two areas: 'Visible Tool' and 'Free Flank Wear', Fig. 13. The first category includes any part of the image that is in focus. The second category includes standard flank wear, as well as adhesion to the flank face and cutting edge. Segmentation is performed at pixel level, and the result is exported as a JSON file in the COCO image segmentation dataset format.

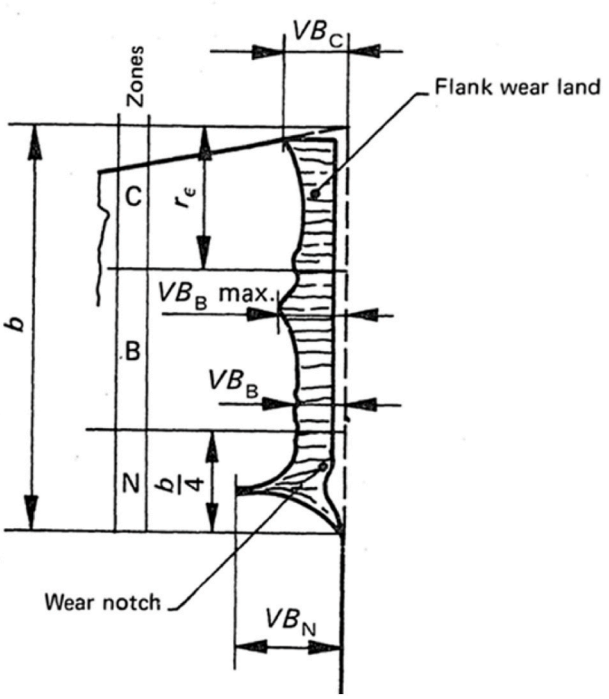


Fig. 15. Tool wear on the free flank after ISO:3685-1993 [ISO93].

4. Analytic validation of the setup

To validate the sensitivity of the setup to specify the tool wear, two experiments in X5-CrNi18-10 are carried out and subsequentially analysed in relation to changes in the different signals as the tool wears down. The hypothesis is that any change in signal contains information that can be used for subsequent machine learning. If the signals do not change, either the recorded channel does not reflect connection to the tool wear, or it is not sensitive enough to reflect the level of change in the tool. Two different tools were used until they reached end of tool life.

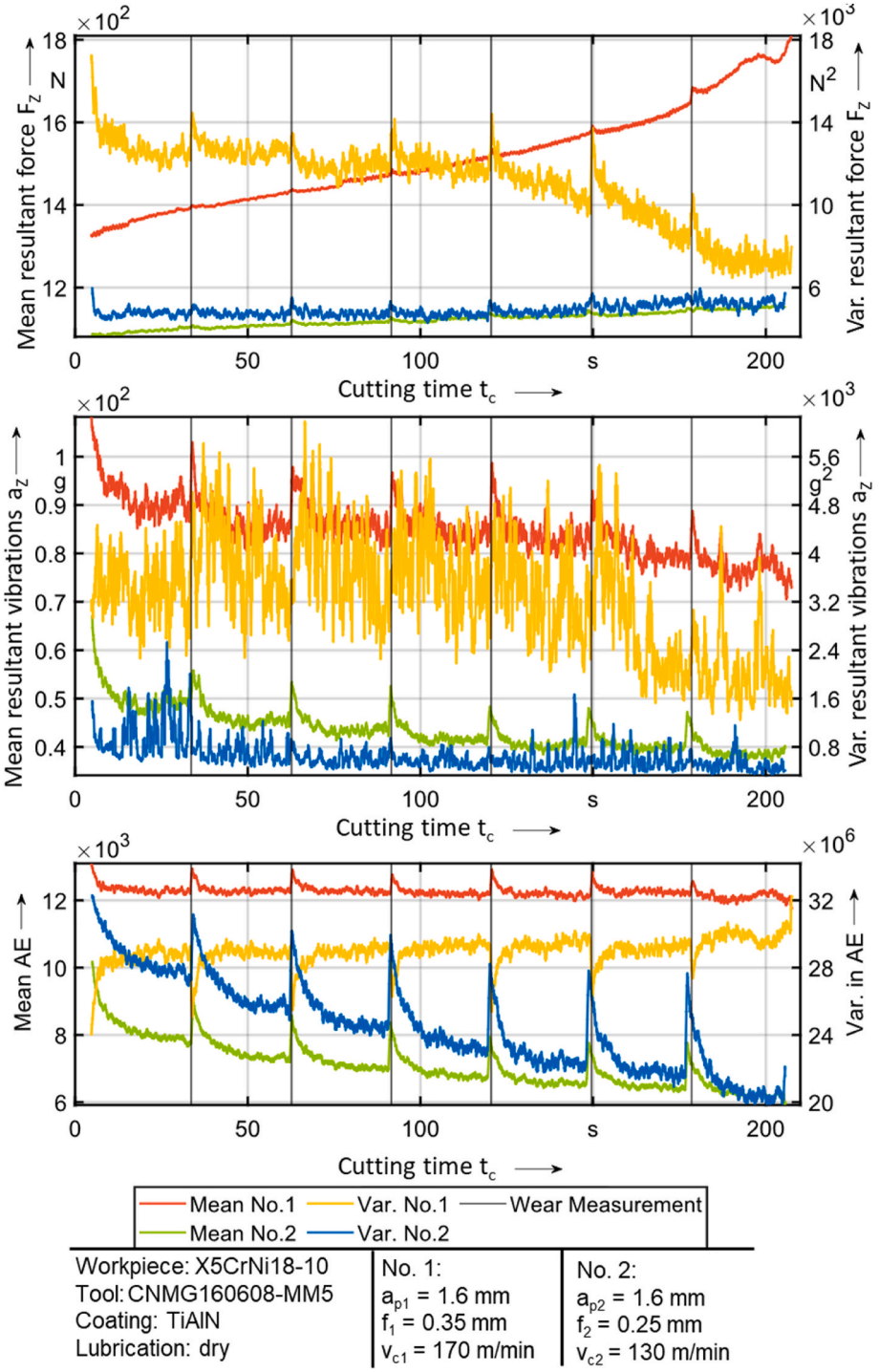


Fig. 16. Mean and variance over the process time from selected sensors.

The experimental parameters used for validation, reflect changes in the cutting parameters while maintaining the coating properties and workpiece material, Table 5.

In both experiments a tool life of $t_c = 210$ s was reached. Fig. 14 shows the progression of wear on the free flank over the combined cutting time $t_c = 210$ s. The wear was evaluated manually and the different areas of wear on the clearance flank are labelled using the definitions from ISO:3685-1993 [17], as shown in Fig. 15 In Experiment No. 1, the rake face of the tools broke. It exhibits dominant wear in Area C at the beginning, which remains dominant throughout the tool's lifespan. Wear in Area C increases drastically towards the end of the tool's lifespan, caused by chipping of the cutting edge, indicating its end. In Experiment No. 2, chip formation could no longer be sustained with the given setup. The tool used in Experiment No. 2 exhibits similar wear in Areas N, B and C, with Area VB_N being the dominant wear area and is the source of the unsustainable chip formation at the end of the experiment. Both tools exhibited significant run-in wear within the first $t_c = 30$ s of cutting, but displayed different dominant wear areas over their lifespans.

Fig. 16 shows the resultant force F_Z, resultant vibrations a_Z and AE signal from combined cutting time $t_c = 210$ s of from both tools. The mean and variance over a moving window of $t_{MW} = 0.5$ s are displayed

Table 3
Experiment parameters.

No.	a_p [mm]	v_c [m/min]	f [mm]	Workpiece Material	Tool	Coating	$t_{Coating}$ [µm]	$HV_{Coating}$ [HV]
1	1.6	170	0.35	X5-CrNi18-10	CNMG160608-MM5	TiAlN	3.5 ± 0,3	2071.6 ± 168
2	1.6	130	0.25	X5-CrNi18-10	CNMG160608-MM5	TiAlN	3.5 ± 0,3	2071.6 ± 168

for the cutting force, vibrations, and acoustic emission (AE). The resultant force F_Z and resultant tool vibrations a_Z are calculated as the square root of the sum of the squares of the three channels of the dynamometer and acceleration sensor while for the AE signal, the absolute value is taken to calculate the mean and variance. The AE signal is measured in mV, which can be scaled arbitrarily at the preamplifier. Consequently, the plot of the signal is unitless and is intended for qualitative comparison.

In Experiment No. 1, the resultant force $F_{Z,}$ follows a clear S-curve. The variance for Experiment No. 1, however, follows an inverted S-curve. Together, they depict a cutting process that begins with high variation and low forces, followed by a run-in phase and a linear change that transitions to an exponential change at the end, resulting in high forces and low variation. The mean cutting vibration value for Experiment No. 1 initially remains low, increases after $t_c = 30$ s and then drops to an even lower level at the end. The AE signal for Experiment No. 1 does not change significantly during the runtime of the experiment.

In Experiment No. 2, the resultant force F_Z increases linearly throughout the entire cutting time. There does not appear to be a run-in phase, and the variation and mean increase linearly throughout the experiment. In contrast, the mean cutting vibration value for Experiment No. 2 exhibits a brief run-in phase, after which the signal decreases linearly for the remainder of the experiment. The greatest variation in cutting vibration occurs during the first $t_c = 30$ s of cutting, after which it drops for the rest of the runtime, increasing slightly only periodically towards the end. Lastly, Experiment No. 2 shows the greatest change in AE sensor readings over the tool life. Here, the mean and variance of the signal follow a linear downward trend.

As would be expected due to the different cutting parameters, the overall signals in both experiments differ. Experiment No. 1 shows higher means and variance throughout the entire cutting time for all signals shown. This is due to the higher feed rate and cutting speed. Both experiments are interrupted at $t_c = 30$ s intervals to measure tool wear. As can be seen, after each stop, the resultant force $F_{Z,}$ resultant vibrations a_Z and AE signal show a short run-in period, which repeats after every stop. Nevertheless, the $t_c = 30$ s time window of continuous cutting is long enough to reach a stable process. Both experiments demonstrate an increase in mean cutting force, which correlates directly with wear of the cutting tool. The mean cutting vibration value is similar for both sensors. Initially, there is a brief run-in phase, after which the signal decreases linearly for the remainder of the experiment. The AE sensor shows different behaviour in the two experiments. Most interestingly, the change in variance of this sensor after the stoppage between the $t_c = 30$ s cutting windows appears inverted between the two experiments. This behaviour is not evident in the mean value of the AE sensor or the other two sensors.

As can be seen, the three different sensors show significant changes due to tool wear and changes in cutting parameters. The sensors show different behaviour over the course of the experiment and are therefore not redundant. Together, they can be used to draw conclusions about the status of tool wear at a given point.

5. Data analysis with machine learning

This section introduces an initial data analysis using autoencoders. We describe the data set and preprocessing, summarize convolutional autoencoders in some detail, and present results.

Table 4
Parameters for sampling the training data.

Categories	a_x, a_y, a_z, F_x, F_y, F_z
Cutting cycles	$n_c = 7$
Sampling rotations in each cycle	$n_r = 124$
Snippets in each rotation	$n_s = 10$
Time steps	256
Overlap of time steps	32

Table 5
Autoencoder family.

k_s	ℓ	k_f	Parameters
5	14	5	848
7	9	5	9064
9	6	9	8184
11	5	5	8360
13	4	5	8152
15	3	9	7560
17	2	17	6584
19	2	11	7096
21	2	5	7608
23	1	21	5864
25	1	17	6184
27	1	13	6504
29	1	9	6824
31	1	5	7144
33	0	33	4248

5.1. Generating dataset for ML-based validation

The data was collected using the experimental setup as described in Section 5. For the following, we solely focus on the synchronised acceleration and force data. Acceleration and force are each measured in x, y, and z direction, i.e., the data has six channels in total. The workpieces are slotted to enable measurements with the optical temperature sensors, thus periods of actual cutting are interlaced with shorter periods of no cutting. Since the radius of the workpiece is reduced with every round, the lengths of the respective intervals change during time. At the same time, there are also significant data changes when the tool touches and leaves the edge of the workpiece as the acceleration and force values are 0 without contact. We consider the recorded six channels for a complete run, i.e., Experiment No. 1 in Table 3.

We used the z-component of the measured force to determine the boundaries of the regions where cutting takes place, corresponding to regions where the force is around 680 N. The data is rapidly oscillating because of the high sensor frequency (51.2 kHz). Let f_i denote the measured oscillating forces, where i denotes the time step. We used an exponential moving average ((EMA) $e_{i+1} = e_i\gamma + f_i(1-\gamma)$ with $\gamma = 0.995$ and $e_1 = 0$ in forward and backward direction to compute two functions with less variations. The EMA is known to lag behind the actual data changes. We use a threshold of 500 to cut out start and end points of actual cutting intervals. The start point is determined by the backward EMA, the end point by the forward EMA. In this signal we detect 197 intervals of cutting. The first and last 10 intervals are discarded, compare Fig. 17, which leaves 177 intervals of cutting. In total we have 7 cutting cycles in experiment 1 with a total of 1253 intervals, as well as 7 cycles in experiment 2 with 978 intervals.

An example for one rotation of all six channels around $t = 15$ s is visualized in Fig. 18. There we depict the data at a fine scale, amounting

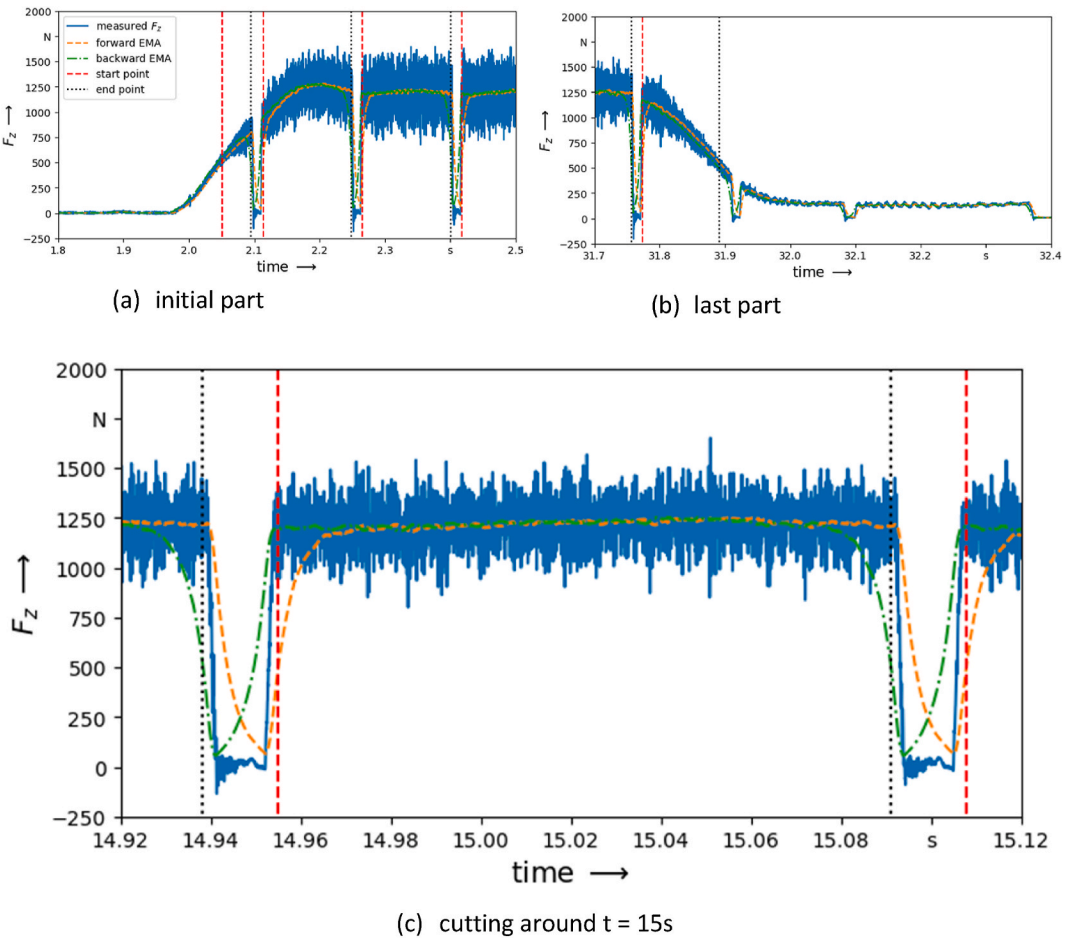


Fig. 17. The determination of boundary of cutting process using EMA and Fz force data. Initial (a) and last part (b) are discarded, (c) shows one interval of cutting in more detail.

to one rotation as explained in Section 4, together with one sample of 256 times steps. For the training of the ML models, we sampled data snippets of length 256 in the constant cutting intervals, i.e., when the tool touches the workpiece. One such sample is marked in purple in Fig. 19. There are 7 cutting cycles for each experiment datasets. After every 30-s cutting cycle, the turning machine stopped to capture images of the free flank wear. We sampled 124 rotations in each cutting cycle, and 10 snippets in each rotation. To train ML models we collected training snippets s_i overlapping with s_{i+1} in 32 positions, where sample number $i = 1, \cdots, 7$. For computing validation and test losses of the models, we collect the sample snippets without any data overlapping with the training data set. The sampling parameters that we used are listed in Table 4.

All channels are normalized over all samples to have mean zero and variance one to ensure that all the input channels have the same scale. In total, the sampled data $\mathbf{s} \in \mathbb{R}^{n,t,c}$ has dimensions (n, t, c), where $n = 8680$ denotes the overall number of samples taken, $t = 256$ denotes the number of time steps, and $c = 6$ the number of channels We split the data into 6944 training samples, 868 validation samples, and 868 test samples.

5.2. Outlier detection based on autoencoders

The time-series data is high-dimensional due to the high sampling frequency. The data is imbalanced, since the points that mark tool failure are rare. The occurrences of abnormal events and/or tool failure are difficult to predict [18]. Most research about machine health monitoring is based on large amounts of labelled data (see, e.g., Refs. [19–21]), but the labelling is often very time-consuming in real manufacturing applications. Therefore, we use autoencoders, a type of neural networks, to reduce the dimensionality and to distinguish the different phases to validate the proposed experimental setup.

An autoencoder [22] is a mathematical function $a : \mathbb{R}^m \to \mathbb{R}^m$ that aims to reconstruct the input on output. It comprises an encoder network $e : \mathbb{R}^m \to \mathbb{R}^n$, where typically $n < m$, and a decoder network d: $\mathbb{R}^n \to \mathbb{R}^m$, i.e., $a = d \circ e$. The encoder part of the network maps the input data $\boldsymbol{x} \in \mathbb{R}^m$ to the so-called latent space representation $\boldsymbol{z} \in \mathbb{R}^n$ which represents a form of compressed data. The decoder part then maps the latent space representation $\boldsymbol{z} \in \mathbb{R}^n$ back to a high-dimensional reconstruction $\boldsymbol{x}' \in \mathbb{R}^m$. In this paper, the data for reconstruction is temporal sequence data with six channels. There are different approaches to train an autoencoder for such a task: neglect temporal and channel dependencies and use dense layers, i.e., a simple fully-connected feedforward neural network; use a one-dimensional convolutional neural network (CNN); map the data to the frequency domain using some transform and use a two-dimensional CNN. In this paper we focus on 1D fully convolutional autoencoders which not only still keep the temporal and channel dependencies without frequency transformation but also have lower

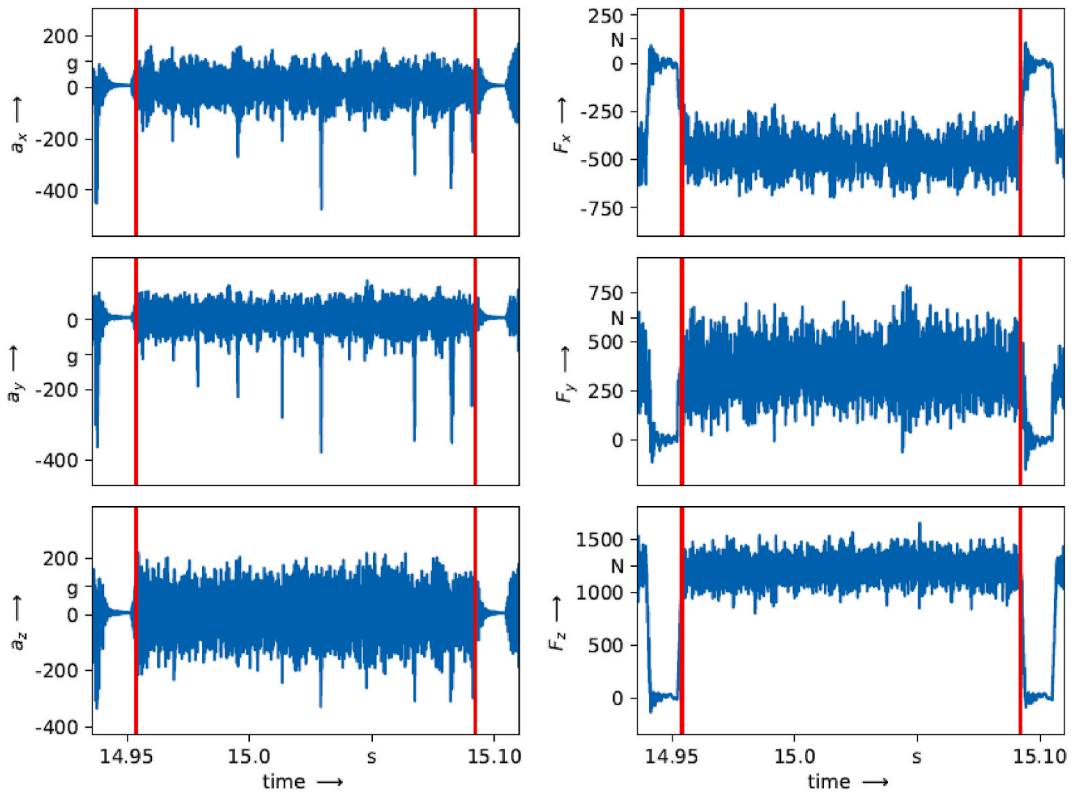


Fig. 18. One cutting frame of the six channels of the recorded signals of both the acceleration and force sensor in the first example run.

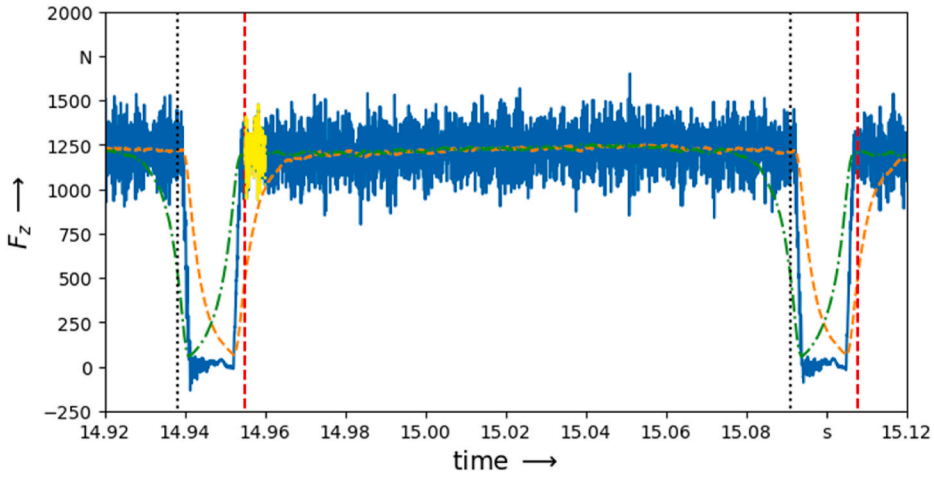


Fig. 19. A representative cut showing the force data along the z-axis (blue); a sample used in the training (yellow). (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

computation complexity.

5.2.1. 1D fully convolutional autoencoder

A convolutional neural network (CNN) is a class of deep neural networks that is robust and easy to understand, so the application of CNNs in the real world has been widespread. The convolutional neural networks were first proposed by LeCun et al. for handwriting recognition [23]. Classical CNNs are usually used for two-dimensional image and three-dimensional data recognition in computer vision. In industrial applications, one-dimensional (1D) CNNs allow time-series sensor data from the machining process to be used without preprocessing. Moreover, 1D CNNs require significantly less computational power compared to 2D/3D CNNs, since they operate on simple array operations rather than matrix operations. This makes them faster to train and suitable for real-time applications [24].

To ease the understanding of the properties of a fully convolutional autoencoder, we sketch a simple example in Fig. 20. In this figure a fully convolutional autoencoder is represented that is based on the following mathematical structure, where : $\mathbb{R}^6 \to \mathbb{R}^2$, $e : \mathbf{x} \mapsto \mathbf{z_2}$ is the encoder and $d : \mathbb{R}^2 \to \mathbb{R}^6$, $d : \mathbf{z_2} \mapsto \mathbf{z_4}$ the decoder:

$$\begin{array}{llllllll} \boldsymbol{a}_1 & := \boldsymbol{x} \in \mathbb{R}^6, & \boldsymbol{z}_1 & := \boldsymbol{a}_1 \star_{\text{valid}} \boldsymbol{f}_1 \in \mathbb{R}^4, & \boldsymbol{a}_2 & := \text{ReLU} \in \mathbb{R}^4, & \boldsymbol{z}_2 & := \boldsymbol{a}_2 \star_{\text{valid}} \boldsymbol{f}_2 \in \mathbb{R}^2, \\ & & \boldsymbol{z}_3 & := \boldsymbol{a}_3 \star_{\text{full}} \boldsymbol{f}_3 \in \mathbb{R}^4, & \boldsymbol{a}_4 & := \text{ReLU} \in \mathbb{R}^4, & \boldsymbol{z}_4 & := \boldsymbol{a}_4 \star_{\text{full}} \boldsymbol{f}_4 \in \mathbb{R}^6. \end{array}$$

Even though CNN are termed *convolutional*, the standard frameworks are based on the *cross-correlation* of the input and a filter. In the encoder we usually aim for a reduction of the dimension, thus we use the valid cross-correlation defined for input $\mathbf{a} = (a_1, \ldots, a_n)^T \in \mathbb{R}^n$ n and filter $\mathbf{f} = (f_1, \ldots, f_k)^T \in \mathbb{R}^k$ by

$$z_j = (\mathbf{a} \star_{\text{valid}} \mathbf{f})_j := \sum_{i=1}^{k} a_{i+j-1} f_i, j = 1, \ldots, n-k+1.$$

Neural networks often satisfy a so-called universal approximation property, for this a non-linear so-called activation function is necessary. We utilized a very basic example, the so-called rectified linear unit (ReLU). This function acts component-wise on the input and is defined by $\text{ReLU}(x) = \max(0, x)$. In this basic example, these two are used to define the encoder. Mostly, several valid cross-correlations followed by application of an activation function are used. The dimension reduction is linear in the sizes of the filters, since the activation function do not change dimensionality. With ℓ convolutional layers with filter sizes $k_1, \ldots, k_\ell$, we reach a dimensional reduction of size

$$n - \sum_{j=1}^{\ell} (k_j - 1) = n + \ell - \sum_{j=1}^{\ell} k_j.$$

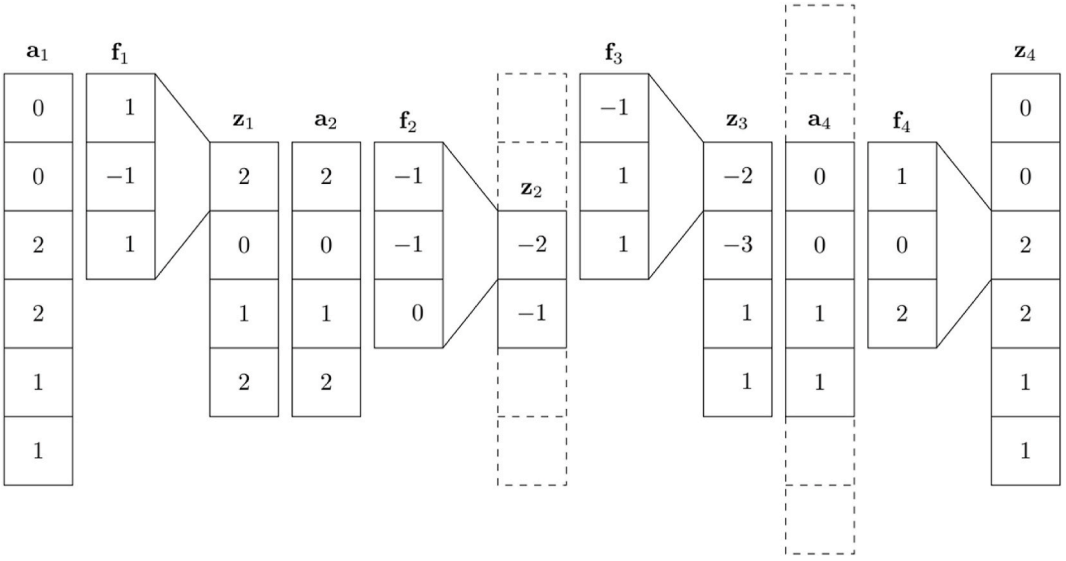


Fig. 20. Simple example of a 1D fully convolutional autoencoder.

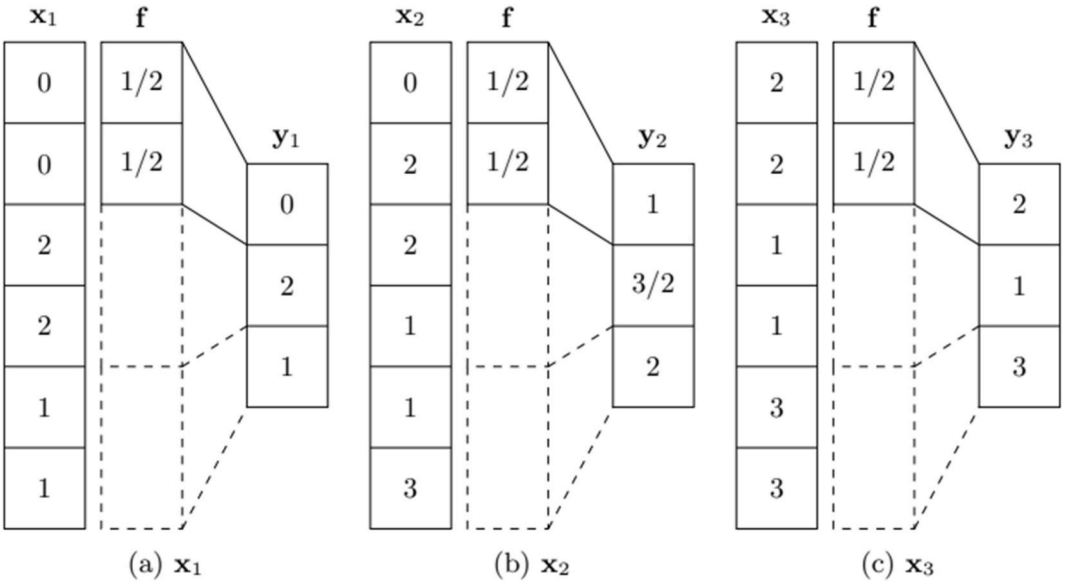
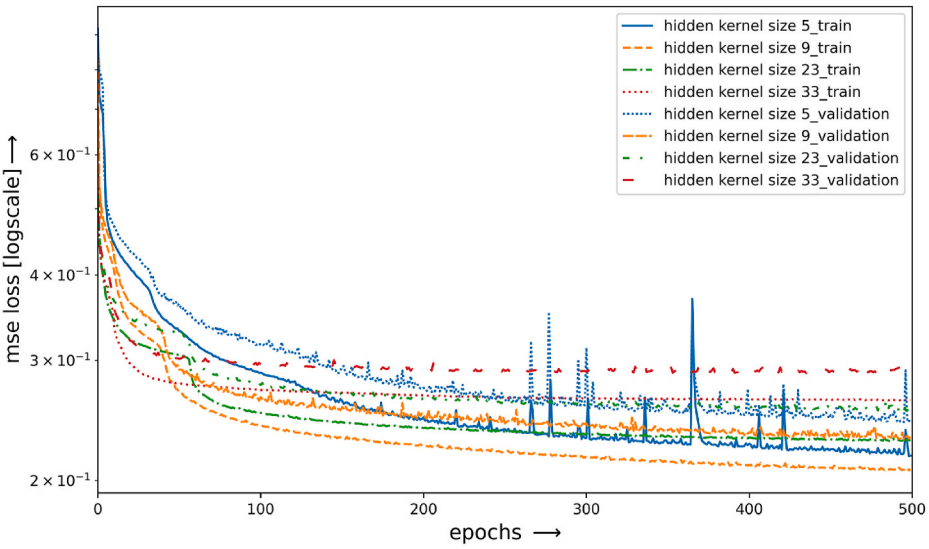
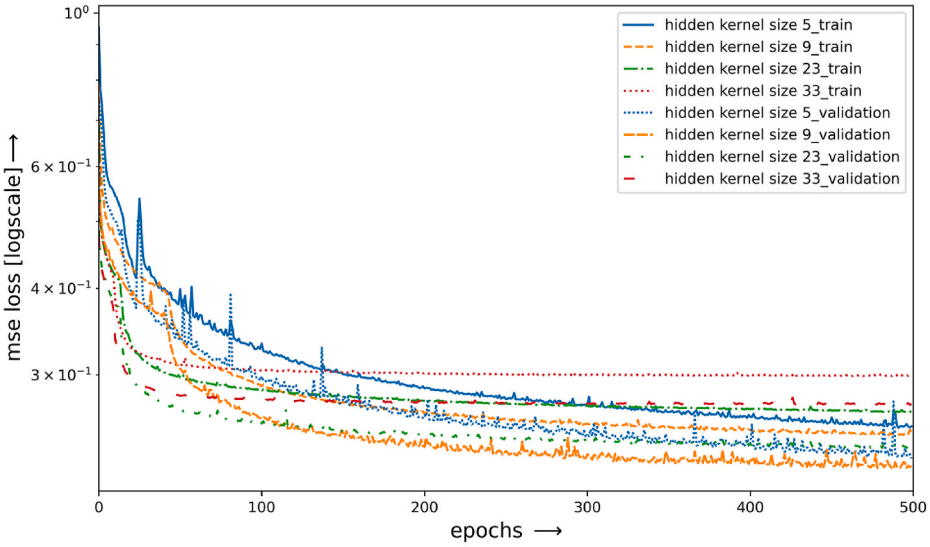


Fig. 21. Example for average pooling of size 2.



(a) Training on dataset 1



(b) Training on dataset 2

Fig. 22. Training of autoencoders with various kernel sizes in the hidden layers on both datasets provided. Depicted are the losses over 500 epochs for the kernel sizes $k \in \{5,9,23,33\}$. The smallest loss was achieved consistently in both cases for a kernel size of $k = 9$. For the training the 8680 training samples of length 256 for each dataset have been used.

This can be used to tune the size of the latent space or to select the number of layers given a fixed filter size. For the decoder we need to recover the original dimension, here so-called transposed of full cross-correlations are used. They can be described by the valid cross-correlation of the input padded by $k-1$ zeros on top and bottom and the filter. The arising increase in the dimension is given by $n+2(k-1)-k+1 = n+k-1$. The autoencoder in the scheme in Fig. 20 reconstructs the given input $a1 = x$ on output, $z4 = a1$. The two-dimensional latent space contains all information necessary to reconstruct a six-dimensional vector. All computations can be carried out for vectors with more entries, only the latent space dimension grows linearly with the input dimension. The first entry only depends on the receptive field of the first five entries, the second on the last five. All-convolutional neural networks based solely on convolutions (i.e., cross-correlations) are shift invariant. Using the valid cross-correlation with filters only reduces the dimension linearly. A higher dimension reduction is obtained using pooling. We explain average pooling with size 2 in Fig. 21. Average pooling with size k reduces the dimension by at least a factor k, since non-integer fractions are not used in the pooling. It can be interpreted as a valid cross-correlation with fixed filter $(1,\ldots,1)^T/k \in \mathbb{R}^k$ using a stride of k. Shifting input by k results in the shifted pooled data, as is visible when comparing Fig. 21 a and 21c–. 21 b shows that in general a shift that is not a multiple of k generates completely different pooled arguments. Convolutions and pooling can be combined, then the shift invariance only holds for multiples of the pooling size.

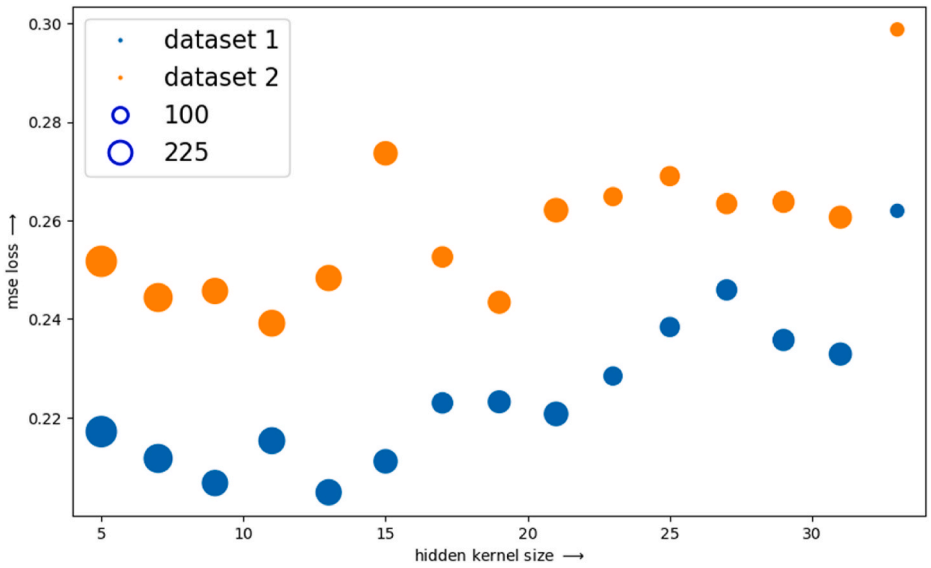


Fig. 23. Training loss on the dataset of experiment 1 and 2. The size of the dots reflects the model size in terms of trainable parameters.

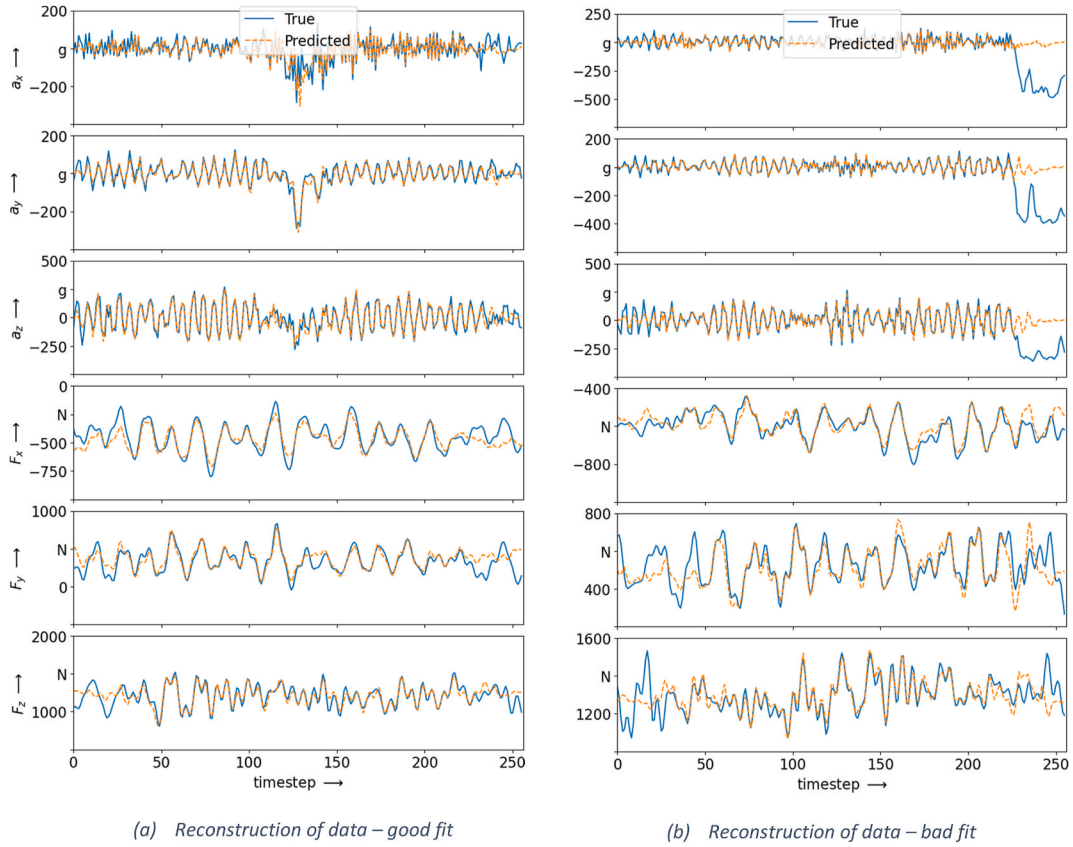
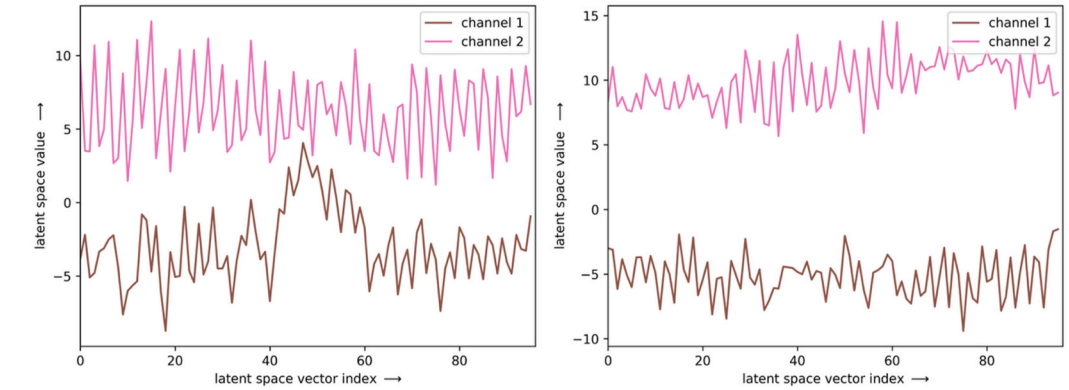


Fig. 24. Data reconstructed using the trained autoencoder with $k = 9$. While the reconstruction of the data in (b) is worse then in (a), the performance is reasonably good in both cases. (a) Reconstruction of data – good fit (b) Reconstruction of data – bad fit.

In our case the training is based on batches of portions of the time series with six channels, i.e., in the so-called "channels last" format the input to the network has a shape (batch, steps, features), where steps = 256 and features = 6. In the convolutional layers not one, but several filters are used. They comprise a filter bank of shape (size, channels, filters), where size is usually some odd integer, channels matches the number of features on input and filters determines the number of filters. Per filter usually one constant bias value is added to retain the shift invariance. This gives

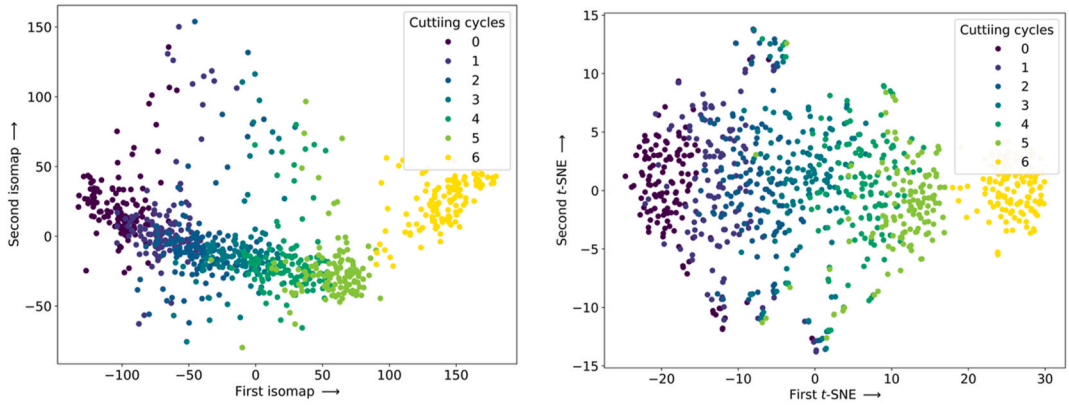
$$\text{size} \cdot \text{channels} \cdot \text{filters} + \text{filters}$$

trainable parameters in a 1D convolutional layer. The output is again 3D and of shape (batch, steps − size + 1, filters), i.e., the dimension reduction behaves still as described above, and the number of filters defines the number of channels on output. Increasing the number of layers and channels increases the capacity of the network to represent



(a) Latent space signals of well fitted data of dataset 1 (b) Latent space signals of badly fitted data of dataset 1

Fig. 25. Visualization of latent spaces corresponding to good and bad fits shown in Fig. 24a and b.



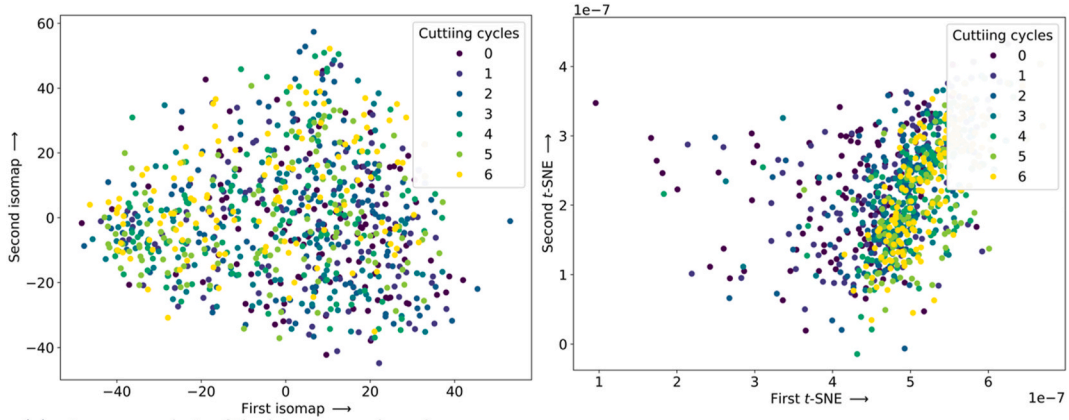
(a) Isomap analysis of the latent space based on experiment No.1 (b) t-SNE analysis of the latent space based on experiment No.1

Fig. 26. Isomap and t-SNE visualization of the latent space from the test dataset of experiment 1. A clear clustering of the cutting cycles is visible.

functions.

5.2.2. *A family of fully convolutional autoencoders*

Selecting the number of layers and the kernel sizes and numbers of channels in each, both for encoder and decoder, influences the dimension of the latent space, the number of trainable model parameters, and the training process. We focus on symmetric layers in the encoder and decoder. To ensure a comparable amount of dimension reduction we fixed the dimension of the latent space to 96 time steps and 2 channels, i.e., of shape (96, 2), given training samples of 256 time steps with 6 channels, i.e., of shape (256, 6). To ease the search for a suitable architecture we decided to select a fixed kernel size k_s and a fixed number



(a) Isomap analysis of the latent space based on experiment No.2 (b) t-SNE analysis of the latent space based on experiment No.2

Fig. 27. Isomap and t-SNE visualization of the latent space from the test dataset of experiment 2. Here, no clusters are visible, and the analysis of the latent space remains inconclusive.

of filters

$n_f = 8$ in the first layers of the encoder and thus the last layers of the decoder. We use one average pooling layer of size 2 as last layer in the encoder and a corresponding upsampling layer in the first layer of the decoder, leaving the channels intact. We need to find a structure that maps 256 to $192 = 2 \cdot 96$ using 1D convolutional layers. For this we need a filter bank of 2 filters with possibly smaller filter size k_f. We use Leaky Rectified Linear Unit (LeakyReLU) activation functions defined as $\text{LeakyReLU}(x) = \begin{cases} ax, x < 0 \\ x, x > 0 \end{cases}, a = 0,3$. Using the filter arithmetic for valid convolutions, we arrive at the structure for the encoder shown in Table 5, the decoder has a mirrored structure.

5.3. Results

We trained a family of autoencoders with varying kernel sizes k_s, k_f (see Table 5).

We use the mean squared error (MSE) loss, and the AdamW optimizer. Training was conducted over 500 epochs with a learning rate of 0.001 and a batch size of 64. Fig. 22a and b show training and validation losses. In all configurations, the initial losses were reduced significantly, only the training loss for $k = 33$ appears to stall at a higher value. For the remaining experiments and results we use kernel size $k = 9$, which appears to be performing well for both experiments. Fig. 23 shows final training losses for both experiments for the family of autoencoders, i.e., models with the same latent space, after training on sample snippets of length 256 for 500 epochs. Models with larger sizes typically achieve lower loss. However, if the CNN kernel size is too small, such as $k = 5$, it may lead the model to capture more noise than meaningful signal features. We also note that loss values are consistently higher for data coming from experiment 2. Fig. 24 shows the reconstructed signal predicted by the autoencoder together with the true data. Fig. 24a shows a data set that is well predicted. For Fig. 24b we chose data that a high loss value, indicating a worse fit. However, in both cases the signals are predicted reasonably well. Data near the boundary is reconstructed less accurate due to the use of CNNs without padding, thus missing information on the boundary.

Fig. 25 shows latent space signals corresponding to the fits shown in Fig. 24. While a difference in the two cases is visible, it is unclear how to interpret these signals. To further visualize the latent space, we thus use the nonlinear dimension reduction techniques *t*-SNE [25] and Isomap [26]. Both algorithms were executed 10 times with different random seeds and to check if the identified clusters change due to different random initial start. Results are shown in Figs. 26 and 27. For experiment 1, the cutting cycles form relatively clear clusters in the latent space. This suggests the latent vectors of shape (96,2) have learned the features from different cutting cycles, as early cutting cycles are mapped to a different part of the latent space than later cycles. For experiment 2, some ordering according to cycle number is visible in the *t*-SNE plot. However, as no clear clusters are visible in the Isomap analysis this might be an artifact of the method. One reason for the difference to experiment 1 might be that experiment 2 used lower cutting speed and feeding rate. The change of tool wear was more stable and does not show significantly fast signal change. While further research is required to investigate these differences, the results indicate that the trained autoencoder can successfully reduce the dimensionality of the data and can identify clusters, thus can serve as a basis for future outlier detection and tool wear prediction.

6. Conclusion and outlook

As demonstrated, the proposed novel data acquisition concept can generate unique data that is sensitive to changes in tool wear a and process parameters. The setup enables in situ measurement and makes subsequent cutting experiments for one tool more comparable. The high degree of automation in data capture reduces possible user errors and makes the collected data more uniform. This was achieved by analysing cutting force, vibration and acoustic emission (AE), as well as using an auto-encoder approach. Furthermore, the high potential of auto-encoders for feature extraction was demonstrated. This is a promising starting point for future machine learning approaches that require a smaller input size, which can be extracted from the latent space of the encoders. The main achievements of this work are.

- A novel data acquisition concept for cutting based on a combined sensor concept is developed
- The concept has been validated and sensitivity in monitoring the tool wear during cutting is improved
- The application of autoencoder on the signal data as part of the concept enables fast process parameter adaption

It is planned to use the shown setup in a larger series of tests to develop a grey box model that is able to predict the tool life for the two materials used analysing different tool coating properties and parameter combinations with the help of the data recorded in the process. Various greybox approaches such as SINDY will be used and compared. Furthermore, an active learning approach based on a Gaussian process will be implemented to minimise the number of experiments required to achieve this goal. The dataset which is created for this purpose will be curated and made publicly available in accordance with the guidelines of the DFG.

CRediT authorship contribution statement

Sebastian Schibsdat: Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Ya-Jing Wu:** Writing – original draft, Software. **Martin Keunecke:** Writing – original draft, Methodology, Investigation. **Sarah Baron:** Writing – review & editing, Investigation. **Daniel Höche:** Writing – review & editing, Supervision. **Jens-Peter Zemke:** Writing – review & editing, Writing – original draft, Supervision, Methodology. **Sebastian Götschel:** Writing – review & editing, Supervision, Methodology. **Christoph Herrmann:** Writing – review & editing, Supervision. **Jan Hendrik Dege:** Writing – review & editing, Supervision.

Declaration of competing interest

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

Acknowledgement

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 521385417 (SPP 2402 - Greybox models for the qualification of coated tools for high-performance cutting, subproject D2: Extrapolative digital greybox models for describing and predicting the macroscopic system behavior of TiAlN-coated cutting tools).

Data availability

The data used in this publication is available at https://doi.org/10.15480/882.15882.

References

[1] M. Balazinski, E. Czogala, K. Jemielniak, J. Leski, Tool condition monitoring using artificial intelligence methods, Eng. Appl. Artif. Intell. 15 (2002) 73–80, https://doi.org/10.1016/S0952-1976(02)00004-0.

[2] T. Wong, W. Kim, P. Kwon, Experimental support for a model-based prediction of tool wear, Wear 257 (2004) 790–798, https://doi.org/10.1016/j.wear.2004.03.010.
[3] M. Pratama, E. Dimla, C.Y. Lai, E. Lughofer, Metacognitive learning approach for online tool condition monitoring, J. Intell. Manuf. 30 (2019) 1717–1737, https://doi.org/10.1007/s10845-017-1348-9.
[4] S. Capasso, J.M. Paiva, E.L. Junior, L. Settineri, K. Yamamoto, F.L. Amorim, R. D. Torres, D. Covelli, G. Fox-Rabinovich, S.C. Veldhuis, A novel method of assessing and predicting coated cutting tool wear during Inconel DA 718 turning, Wear 432–433 (2019) 202949, https://doi.org/10.1016/j.wear.2019.202949.
[5] L. Pagani, P. Parenti, S. Cataldo, P.J. Scott, M. Annoni, Indirect cutting tool wear classification using deep learning and chip colour analysis, Int. J. Adv. Manuf. Technol. 111 (2020) 1099–1114, https://doi.org/10.1007/s00170-020-06055-6.
[6] S. Dutta, S.K. Pal, R. Sen, On-machine tool prediction of flank wear from machined surface images using texture analyses and support vector regression, Precis. Eng. 43 (2016) 34–42, https://doi.org/10.1016/j.precisioneng.2015.06.007.
[7] H. Mamledesai, M.A. Soriano, R. Ahmad, A qualitative tool condition monitoring framework using convolution neural network and transfer learning, Applied Sciences 10 (2020) 7298, https://doi.org/10.3390/app10207298.
[8] H. Miao, Z. Zhao, C. Sun, B. Li, R. Yan, A U-Net-Based approach for tool wear area detection and identification, IEEE Trans. Instrum. Meas. 70 (2021) 1–10, https://doi.org/10.1109/TIM.2020.3033457.
[9] T. Bergs, C. Holst, P. Gupta, T. Augspurger, Digital image processing with deep learning for automated cutting tool wear detection, Procedia Manuf. 48 (2020) 947–958, https://doi.org/10.1016/j.promfg.2020.05.134.
[10] L. Colantonio, L. Equeter, P. Dehombreux, F. Ducobu, A systematic literature review of cutting tool wear monitoring in turning by using artificial intelligence techniques, Machines 9 (2021) 351, https://doi.org/10.3390/machines9120351.
[11] E. Pintelas, I.E. Livieris, P. Pintelas, A grey-box ensemble model exploiting black-box accuracy and white-box intrinsic interpretability, Algorithms 13 (2020) 17, https://doi.org/10.3390/a13010017.
[12] T.J. Rogers, G.R. Holmes, E.J. Cross, K. Worden, On a grey box modelling framework for nonlinear System identification, in: N. Dervilis (Ed.), Special Topics in Structural Dynamics, vol. 6, Springer International Publishing, Cham, 2017, pp. 167–178, https://doi.org/10.1007/978-3-319-53841-9_15.
[13] S. PalDey, S.C. Deevi, Single layer and multilayer wear resistant coatings of (Ti,Al)N: a review, Mater. Sci. Eng. A342 (2003) 58–79.
[14] M. Keunecke, C. Stein, K. Bewilogua, W. Koelker, D. Kassel, H. van den Berg, Modified TiAlN coatings prepared by d.c. pulsed magnetron sputtering, Surf. Coating. Technol. 205 (2010) 1273–1278, https://doi.org/10.1016/j.surfcoat.2010.09.023.
[15] Aharon Inspektor, A. Paul, Salvador; Architecture of PVD coatings for metalcutting applications: a review, Surf. Coating. Technol. 257 (2014) 138–153, https://doi.org/10.1016/j.surfcoat.2014.08.068.
[16] R. Brodmann, B. Brodmann, Optical roughness and form measurement by means of light scattering, VDI-Ber. (2007) 79–88.
[17] DIN E.V. (Publisher): ISO 3685:1993, tool-life Testing with Single-point Turning Tools, Beuth-Verlag, Berlin, 1993.
[18] M. Hassan, A. Sadek, H. Attia, A real-time deep machine learning approach for sudden tool failure prediction and prevention in machining processes, Sensors 23 (8) (2023) 3894, https://doi.org/10.3390/s23083894.
[19] D. Jana, J. Patil, S. Herkal, S. Nagarajaiah, L. Duenas-Osorio, CNN and Convolutional Autoencoder (CAE) based real-time sensor fault detection, localization, and correction, Mech. Syst. Signal Process. 169 (2022) 108723, https://doi.org/10.1016/j.ymssp.2021.108723.
[20] M.A.Z. Harandi, C. Li, C. Schou, S.L. Villumsen, S. Bøgh, O. Madsen, STAD-FEBTE, a shallow and supervised framework for time series anomaly detection by automatic feature engineering, balancing, and tree-based ensembles: an industrial case study, in: 2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Seattle, WA, USA, 2023, pp. 840–846, https://doi.org/10.1109/aim46323.2023.10196288.
[21] T. von Hahn, C. Mechefske, Self-supervised learning for tool wear monitoring with a disentangled-variational-autoencoder, Int. J. Hydromechatronics 4 (1) (2021), https://doi.org/10.1504/IJHM.2021.10035377.
[22] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, MIT Press, Cambridge, MA, USA, 1986, pp. 318–362, isbn: 026268053X. Foundations.
[23] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551, https://doi.org/10.1162/neco.1989.1.4.541.
[24] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, 1D convolutional neural networks and applications: a survey, Mech. Syst. Signal Process. 151 (2021) 107398, https://doi.org/10.1016/j.ymssp.2020.107398.
[25] L.J.P. van der Maaten, G.E. Hinton, Visualizing high-dimensional data using t-SNE, J. Mach. Learn. Res. 9 (86) (2008) 2579–2605, issn: 1532-4435.
[26] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323, https://doi.org/10.1126/science.290.5500.2319.