Production Engineering (2023) 17:847–859
https://doi.org/10.1007/s11740-023-01209-3

COMPUTER ASSISTED APPROACHES IN PRODUCTION ENGINEERING

Deep representation learning and reinforcement learning for workpiece setup optimization in CNC milling

Vladimir Samsonov[1] · Enslin Chrismarie[1] · Hans-Georg Köpken[2] · Schirin Bär[1] · Daniel Lütticke[1] · Tobias Meisen[3]

Received: 16 September 2022 / Accepted: 12 May 2023 / Published online: 29 June 2023
© The Author(s) 2023

Abstract
Computer Numerical Control (CNC) milling is a commonly used manufacturing process with a high level of automation. Nevertheless, setting up a new CNC milling process involves multiple development steps relying heavily on human expertise. In this work, we focus on positioning and orientation of the workpiece (WP) in the working space of a CNC milling machine and propose a deep learning approach to speed up this process significantly. The selection of the WP's setup depends on the chosen milling technological process, the geometry of the WP, and the capabilities of the considered CNC machining. It directly impacts the milling quality, machine wear, and overall energy consumption. Our approach relies on representation learning of the milling technological process with the subsequent use of reinforcement learning (RL) for the WP positioning and orientation. Solutions proposed by the RL agent are used as a warm start for simple hill-climbing heuristics, which boosts overall performance while keeping the overall number of search iterations low. The novelty of the developed approach is the ability to conduct the WP setup optimization covering both WP positioning and orientation while ensuring the axis collision avoidance, minimization of the axis traveled distances and improving the dynamic characteristics of the milling process with no input from human experts. Experiments show the potential of the proposed learning-based approach to generate almost comparably good WP setups order of magnitude faster than common metaheuristics, such as genetic algorithms (GA) and Particle Swarm Optimisation (PSA).

Keywords CNC Milling · Deep representation learning · Learning-based optimization · Process optimization · Reinforcement learning · Workpiece setup optimization

✉ Vladimir Samsonov
vladimir.samsonov@ima.rwth-aachen.de

Enslin Chrismarie
chrismarie.enslin@ima.rwth-aachen.com

Hans-Georg Köpken
hans-georg.koepken@siemens.com

Schirin Bär
baerschi@amazon.de

Daniel Lütticke
daniel.luetticke@ima.rwth-aachen.de

Tobias Meisen
meisen@uni-wuppertal.de

[1] Chair of Information Management in Mechanical Engineering (WZL-MQ/IMA), RWTH Aachen University, Aachen, Germany

[2] Digital Industries, Siemens AG, Erlangen, Germany

[3] Chair of Technologies and Management of Digital Transformation, University of Wuppertal, Wuppertal, Germany

1 Introduction

Mass production with large series over long periods provides the possibility of investing more time and expert knowledge into process optimization. However, a modern trend towards customized products with shorter lifecycles [17] makes it crucial to be able to frequently set up new, efficient production processes with little effort. Setting up a CNC machine requires a high level of human involvement in the design and subsequent fine-tuning of the manufacturing process. Nevertheless, due to the high degree of automation, CNC machines are used frequently. For this reason, we investigate how to increase the level of automation of the initial setup process by methods of deep learning. This initial setup process comprises several substeps. Firstly, based on the required workpiece (WP) geometry and the chosen milling strategy, an expert generates a Numerical Control (NC) program defining the movement of the cutting tool in relation to the WP. For the processing on the CNC milling machine,

it is essential to decide upon the WP setup in terms of WP positioning and orientation, as this has a significant influence on the machining process [30]. Depending on the current orientation and positioning of the WP, the CNC milling machine might realize the relative trajectory of the milling tool for the WP with different axes movements, resulting in varying levels of total axes traveled distances, speeds, and accelerations. Those directly influence energy consumption, machine wear, and machine oscillations during the processing [14]. Therefore, finding a WP setup that minimizes energy consumption, machine wear, and oscillations during the milling process is essential and selected as our study's main application use case. Selection of a WP setup requires understanding the WP geometry, machining strategy, and machine kinematics. Currently, the WP setup is commonly defined by a CNC programmer or an experienced machine operator.

Considering the fast development of learning-based methods for optimization tasks [24, 29, 41], we hypothesize that deep learning is well-suited for finding a suitable WP setup. The possibility to use Reinforcement Learning (RL) for estimation of near-optimal WP positions and orientations has been shown by [35] and [7]. It is demonstrated that learning-based methods require a considerably smaller number of optimization iterations compared to metaheuristics and yield good results. Nevertheless, all previous approaches require a handcrafted description of the WP geometry and the related machining process, such as an explicit description and coordinates of all geometrical features of the WP. While this approach is suitable for initial viability studies, it lacks generalization possibilities and cannot account for a wide variety of possible WP geometries and machining strategies.

In this work, we introduce a learning-based approach that significantly increases generalization possibilities for the definition of the WP setup. This approach enables the learning of useful and generic representations of the milling process without the need for any human input (see Fig. 1). The learning process is divided into two stages. In the first step, we propose the usage of ordered point clouds describing the movements of the Tool Center Point (TCP) as a representation of milling processes. This enables the generation of compact representations of various milling processes with a Variational Autoencoder (VAE). Subsequently, compact representations of the milling process, generated with the VAE, are provided to the RL agent as a description of the optimization tasks. This allows the trained RL agent to suggest near-optimal WP positioning and orientation, even for previously unseen milling tasks. Finally, we use the solution suggested by the RL agent as a warm start for a hill-climbing heuristic search, leading to a refined quality of the WP positioning and orientation according to the current goal. Our evaluation experimentally shows that the proposed hybrid learning-based optimization approach finds near-optimal WP setups that eliminate axes collisions and minimize total axes traveled distances and accelerations. The comparison performed in this study against common optimization methods such as genetic algorithms (GA) and Particle Swarm Optimization (PSO) shows that the proposed method achieves a higher search efficiency while maintaining a good overall quality of WP setups.

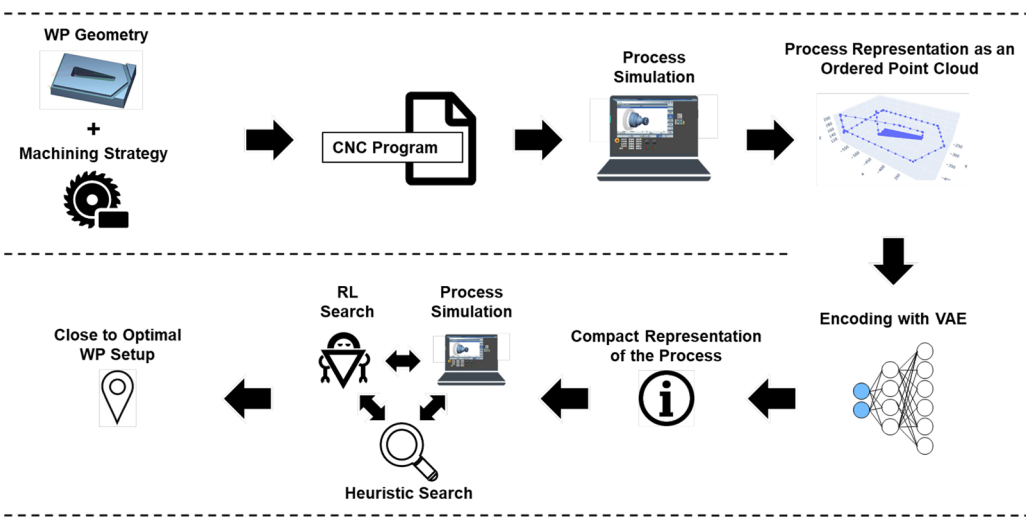


Fig. 1 Overview of the hybrid learning-based optimization method for finding near-optimal WP setups: it involves learning compact and generic representations of WP geometries and related machining strategies with the subsequent use of an RL agent for suggesting a near-optimal WP positioning and orientation, augmented with several simple heuristics for quick refining of the solutions. The given approach leads to finding near-optimal solutions within a small number of search iterations

As demonstrated in the analysis of the current state of the art, our solution delivers a viable contribution towards enabling automated WP setup by addressing five important problem aspects simultaneously:

1 Axis collision avoidance covering axis limit violations, collisions between the tool assembly and the WP, as well as collisions between moving parts of the CNC machine
2 Minimization of total traveled distances of the machine axes
3 Improvement of dynamic characteristics of the milling process, e.g., minimization of the total axes accelerations
4 Generalization to new, previously unseen WP geometries with no expert input
5 Fast solution generation

2 State of the art

The first section concentrates on analyzing the current state of the art for WP setup optimization. It demonstrates that no available solution is capable of simultaneously ensuring the avoidance of axis collision of different complexity, minimization of the axis traveled distances and improving the dynamic characteristics of the milling process. This motivates the importance of developing alternative solution methods covering all those optimization aspects. The second section motivates the use of RL for WP positioning optimization by outlining the advantages of the RL-based optimization methods and looking at the successful use of learning-based methods in other complex optimization domains.

2.1 Optimization of WP setup

Optimization of the milling process in complex CNC applications by adjusting WP position and orientation receives broad attention from the research community. [37] propose to minimize the total X-, Y-, and Z-axis movements by finding a suitable WP position on the rotation table of a 5-axis CNC milling machine. The toolpath in WP coordinates is generated by using CAD/CAM software and split into segments. For each point of the tool trajectory, inverse kinematic calculations are used to transform the WP coordinates into machine coordinates. Subsequently, movements of the machine axes are calculated by linear interpolation between the neighboring trajectory points. The sum of interpolated segments of machine movements serves as the objective function, which is minimized with the help of a GA program. Axis collision avoidance, WP orientation, and optimization of dynamic characteristics of the milling process are not included in the scope of the study by Shaw.

A similar idea is implemented by [43], with inverse kinematics derived for 5-axis CNC milling with two rotary axes. The WP setup is optimized regarding both WP position and orientation to minimize the traveled axes distances. Under the assumption of convex optimization, a gradient descent algorithm is deployed to solve the optimization task. Axis limit violations and collision avoidance between the tool assembly and the WP are considered. Nevertheless, it is not elaborated on what methods are used to detect axis collisions reliably. Collision avoidance between moving parts of the CNC machine is not explicitly mentioned. The optimization of dynamic characteristics of the milling process are not addressed by Wei.

[16] propose a method for finding tool orientation and WP setups with low angular accelerations of rotating axes for a 5-axis milling process. The central idea of the approach is the use of numerical differentiation of inverse kinematic equations to estimate angular accelerations, combined with a GA search for WP positioning and a custom-made heuristic for tool orientation. Again, axis collision avoidance and optimization of axis traveled distances are not considered.

[28] look at the WP setup problem as a two-stage process aiming to minimize the traveled distance of the rotary axes. Firstly, WP orientation is evaluated by determining all orientations resulting in a violation of axis limits, with a consecutive exhaustive search of the remaining discretized solution space for a suitable orientation. Similarly, the WP position is defined in the second stage of the WP setup optimization. Avoidance of collisions between moving parts of the CNC machine and optimization of dynamic characteristics of the milling process is out of the scope of the study.

[4] demonstrate the potential to decrease the energy consumption of finishing milling operations by selecting a suitable WP orientation. The work proposes to estimate the power consumption of a milling machine through weight, accelerations, and weight coefficients on every axis involved in the process. The part related to optimization of WP orientation involves a grid search over the solution space. Axes collision avoidance and WP positioning are not considered.

A study by [10] proposes to model the kinematic behavior of a 3+2-axis CNC milling machine with polytopes. The resulting models can determine well-suited tool and WP orientations, resulting in higher feed rates with low accelerations and fewer jerks. The returned solutions can be checked for violations of axis limits via inverse kinematics of the CNC machine. While offering reasonable solutions for optimizing dynamic parameters of the milling process, the given approach does not optimize the WP positioning and axes' total traveled distances. Additionally, axes collision avoidance does not account for possible collisions caused by moving parts of the milling machine.

As demonstrated in the overview from [39], it is hard to develop an analytical method universally suited for axes

collision detection and avoidance that accounts for both axes limit violations and possible collisions of various parts of the CNC machine. For this reason, [42] proposed to use supervised learning to predict WP setups resulting in collisions. A training dataset is generated via multiple milling simulation runs for a single WP geometry with different positions and orientations on simulation software. The generated dataset is suitable for training K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) classifiers capable of predicting possible collisions for the given WP geometry as a function of the WP setup. The general idea offers low inference time and allows for an efficient search for a suitable WP setup. While valuable on the conceptual level, the idea has limited application value because of the missing capability to generalize to new WP geometries without rerunning extensive simulations for the particular milling task.

As previously introduced, [35] and [7] apply RL for both WP positioning and orientation. The proposed solution is demonstrated to be capable of avoiding axis limit violations and collisions while minimizing total axes traveled distances and accelerations. A SinuTrain simulation of the CNC milling machine is used to simulate milling with reliable detection of axes collisions and limit violations, as well as precise milling process dynamics. The achieved results show that a trained RL agent can propose near-optimal WP setups for previously unseen WP geometries. Using previous experience, a trained RL agent can conduct a far more efficient search of a large solution space than the alternative solutions based on Simulated Annealing (SA) and GA methods. However, both works rely on a handcrafted description of WP geometries (e.g., coordinates of a particular geometrical feature on the WP) that cannot be universally transferred to other WP geometries.

The summary of the current state of the art, provided in Table 1, demonstrates that addressing WP setup optimization covering both dynamic characteristics of the milling process and total axis traveled distances via WP orientation and positioning, while accounting for axis limit violations, collisions between tool assembly and the WP, and collisions between moving parts of the CNC machine is a relevant research question.

2.2 Applicability of learning-based methods to optimization

Research interest in broad applications of learning-based methods for optimization tasks is continuously growing. Compared to common metaheuristic approaches, learning-based approaches do not rely on human experts to design the procedure of generating solutions. Suitable solution strategies can be derived from provided examples or be discovered via a trial-and-error approach [3]. A trained learning-based solution can anticipate good results for new yet similar problem instances without an intensive search of the solution space. This can lead to state-of-the-art performance while maintaining an overall low computation time.

Learning-based optimization is widely applied for optimization across wide variety of domains. ML approaches in engineering and manufacturing fields have been successfully applied, for example, to discover and optimize mechanical properties of new materials [21], process parameter optimization [27, 44], quality prediction [8, 40], process monitoring and control [36, 46], predictive maintenance [33, 47],

Table 1 Overview of the State of the Art for WP setup optimization

Source	WP		Axis collision avoidance			Optimization	
	Orientation	Positioning	Axis limit violations	Between tool and WP	Between moving parts of CNC machine	Axis traveled distances	Dynamic characteristics of milling
[37]	×	✓	×	×	×	✓	×
[43]	✓	✓	✓	✓	×	✓	×
[16]	✓	✓	×	×	×	×	✓
[28]	✓	✓	✓	✓	×	✓	×
[4]	✓	×	×	×	×	✓	✓
[10]	✓	×	✓	✓	×	×	✓
42][1]	✓	✓	✓	✓	✓	✓	×
[35][2]	✓	✓	✓	✓	✓	✓	✓
[7][2]	✓	✓	✓	✓	✓	✓	✓

Heavy check mark-capability is given

Bold times-capability is not available

[1] Missing capability to generalize to new WP geometries

[2] Handcrafted description of WP geometries cannot account for a wide range of possible variations in real milling applications

design optimization [5, 19], chip design [25], production planning [9, 45] and design of optimal control strategies in robotics [22, 31, 32].

In combinatorial optimization, methods of supervised Machine Learning (ML) and RL are successfully applied to such canonical tasks as the Traveling Salesmen Problem (TSP) [2], Vehicle Routing Problems (VRP) [26], Mixed Complementarity Problems (MCP) [23] or Max-Cut Problems [1].

The provided brief outline of learning-based optimization demonstrates that RL methods have the potential to demonstrate good results in the domain of WP setup optimization by utilizing the ability of RL to transfer the previous experience to new tasks and anticipate good solutions without having to search through the whole solution space. An interested reader is invited to refer to dedicated literature reviews [24, 41], offering more examples and details on learning-based optimization solutions.

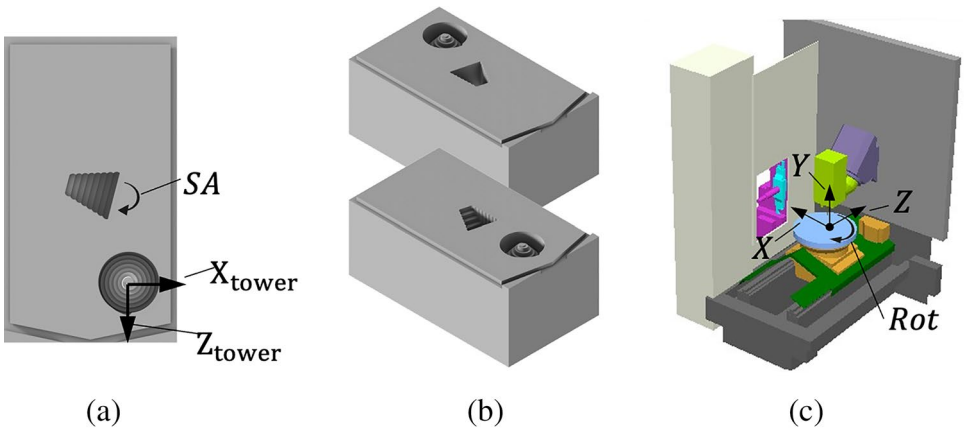
3 Problem description and approach

3.1 Milling process and related setup

The investigation is conducted on a group of WP geometries requiring 5-axis milling. The basic WP geometry can be modified by changing certain geometrical features, referred to as the tower and the milling slot. The tower represents a spiral-shaped surface milled into the WP with sides inclined towards the center. By moving the tower and changing the orientation of the milling slot, a new WP geometry can be generated requiring different WP setups, as demonstrated in Fig. 2b. In this work, WP geometry is changed with the following parameters: slot angle range 0°: 360, tower position range $X_{tower} \in [55\,mm : 145\,mm]$ and $Z_{tower} \in (60\,mm, 235\,mm)$ (see Fig. 2a and b). The Z range for tower position consists only of two discrete values to avoid intersections with the milling slot. Milling is conducted on a CNC machine with a rotary table and a swiveling spindle head as illustrated in Fig. 2c. The CNC machine requires all five axes to produce the WP geometry. Depending on the WP position in the working space of the CNC machine, the same relative movements of the milling tool around the WP will be realized with different axes, potentially resulting in excessive axes moves, accelerations and collisions.

The milling process is simulated with SinuTrain software, allowing for a realistic approximation of the entire milling process, from collisions detection to the determination of dynamic characteristics of axes movements. A simulation run can take up to 15 s, making it infeasible to repeatedly conduct multiple RL experiments requiring millions of runs directly on the SinuTrain software. We collect data over 80.000 SinuTrain simulations with various WP geometries and WP setups and train an ensemble of ML models to address the issue. Firstly, a Light Gradient Boosting Machine (LGBM) classifier decides if there is any axis collision based on WP geometry (tower position and milling slot angle) and WP position and orientation. If no collisions are expected, four LGBM regressors independently predict accelerations and total traveled distances required for the optimization. Overall, the developed approximation of the SinuTrain software can predict the axis collision with an F_1-score of 0.984, and the worst observed coefficient of determination (R^2-score) does not go below 0.984 for all three folds of cross-validation involved in the evaluation of the ML models. The outlined ML approximation is demonstrated to deliver reliable results in a fraction of a second, yet it is valid only for the considered family of WP geometries and the given CNC milling machine. To save computational resources and enable the envisioned training and benchmarking, all reported results in this work are conducted on the ML approximation of the SinuTrain software.

Fig. 2 Milling setup



3.2 Problem formalization

Firstly, we define the problem domain of the WP setup problem by $\Omega \subset \mathbb{R}^4$, such that Ω represents the working space of the 5-axis milling machine. The four input elements that can be controlled are the placement coordinates of the WP in the working space along the X-, Y-, and Z-axis, as well as the rotation angle (Rot) of the rotary table. There also exists a space $\mathcal{S}_i \subset \Omega$ for each NC program, $i = 1, 2, ...$, that does not violate any of the boundary constraints of the milling machine and where no axis collisions occur.

The WP setup task is formalized as a continuous multi-objective optimization problem. We wish to minimize a set of milling process measures $\{D_1, ..., D_M\}$ (for example, accumulated velocities or traveled distances) s.t. $D_m = f_m(X, Y, Z, Rot \mid NC)$, $D_m \in \mathbb{R}$, $f_m : \Omega \mapsto \mathbb{R}$, $m = 1, 2, ..., M$, $f_m(. \mid NC)$ is dependent on the WP-specific NC-program, $\forall\ (X, Y, Z, Rot) \in \Omega$. The functions $f_m(. \mid NC)$ represent the actual milling process, taking the WP setup as input and each returning a specific milling process measurement. We define an indicator function,

$$\mathbf{1}_m(X, Y, Z, Rot \mid NC) = \begin{cases} 0, & (X, Y, Z, Rot) \in \mathcal{S}_i \\ 1, & (X, Y, Z, Rot) \notin \mathcal{S}_i, \end{cases} \tag{1}$$

such that the constrained multi-objective optimization is subject to $\mathbf{1}_m(X, Y, Z, Rot \mid NC) = 0$. To facilitate the application of RL, the multi-objective optimization is transformed into a single-objective optimization via an aggregation function for optimal milling $N = g(D_1, D_2, ..., D_M)$, where $g : \mathbb{R}^M \mapsto \mathbb{R}$.

The D_m-values that are of further interest for the given milling case are defined by domain experts as the integral over squared acceleration (e) and the distance traveled (d) along the Z- and X-axis, therefore $(e_Z, e_X, d_Z, d_X) \in \mathbb{R}^4_+$ are the values to be considered for minimization. These terms are normalized in such a way that the minimization problem becomes a maximization problem through:

$$e'_Z = (e_{Z,max} - e_Z)/(e_{Z,max} - e_{Z,min}), \tag{2}$$

$$e'_X = (e_{X,max} - e_X)/(e_{X,max} - e_{X,min}), \tag{3}$$

$$d'_Z = (d_{Z,max} - d_Z)/(d_{Z,max} - d_{Z,min}), \tag{4}$$

$$d'_X = (d_{X,max} - d_X)/(d_{X,max} - d_{X,min}), \tag{5}$$

where e' and d' are normalized values and $e_{.,min}$, $e_{.,max}$ and $d_{.,min}$, $d_{.,max}$ are the respective minimum and maximum values for e_Z and e_X and d_Z and d_X.

3.3 Learning WP representations

When choosing a WP setup that minimizes the axes traveled distances and related axes accelerations, it is essential to understand how the cutting tool is moved around the WP during processing. The milling strategy and resulting WP geometry are defined in the form of an NC program. In previous studies [7, 35] NC programs are identified via handpicked parameters for a set of geometrical features of considered WP geometries (e.g., orientation of the milling slot and coordinates of the tower position). The practical applicability of such a formalization is limited since it does not account for possible changes in the milling strategy and lacks the flexibility to cover a wide range of geometries. Instead, we propose expressing an NC program as a history of TCP positions during the whole milling process represented by an ordered point cloud.

We place a WP in a fixed position in the CNC machine's working space and record the TCP's positions with constant time intervals. This approach generates an ordered point cloud (see Fig. 3) with detailed information on the milling process. The sequence of positions of the TCP provides an insight into the WP geometry and the involved milling strategy. The distance between the recorded neighboring TCP positions is a proxy for the TCP speed, whereas a change in the distance between the TCP positions enables the derivation of tool acceleration.

The resulting representation of the milling process can contain thousands of TCP position observations and is hard to comprehend for any downstream learning tasks. A meaningful compact representation of the milling process can be learned by using a Variational Autoencoder (VAE)

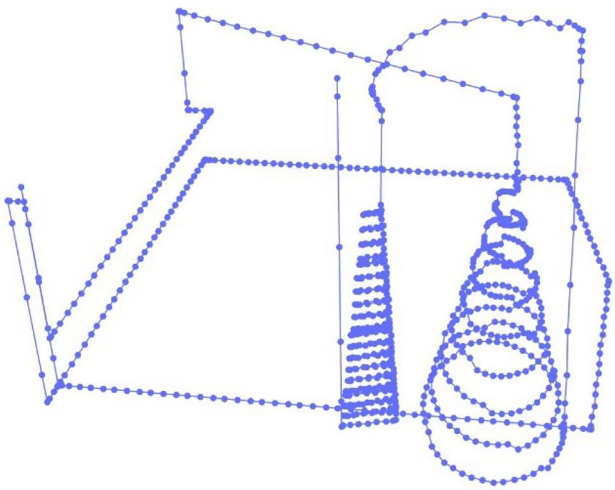


Fig. 3 Representation of TCP movements as an ordered point cloud is a chronological record of the milling process. It contains information on the milling strategy, TCP's traveled directions, distances, speeds, and accelerations

[6]. A VAE is a technique used for deep representation learning [18] which is trained in an unsupervised manner by compressing the input into a latent space by its encoder part and reconstructing it by the decoder part. The training loss is derived from the deviations between the input and reconstructed instances. Unlike the common Autoencoder architecture, a VAE maps each input observation not to a single point in the latent space but to a certain probabilistic distribution. This allows for the regularization of the latent space distribution and learning latent space representations preserving both the local and global structure of the input space. Therefore a trained VAE can generalize to new input instances unseen during the training process. Our custom VAE setup is depicted in Fig. 4.

The input to the VAE is a tabular representation of the ordered point cloud with the TCP observations chronologically ordered in rows. The duration of the milling process varies from WP geometry to WP geometry resulting in a varying number of observed TCP positions. The longest observed milling process for the considered case contains 496 TCP positions. To make sure no other milling process recordings exceed the VAE expected input size, the length of each TCP recording is augmented to the size 550. The padding values are equal to the last TCP observation of the given milling process. All data is normalized with a min-max normalization in a range between zero and one. The first two layers of the VAE encoder consist of 1D convolutions and pooling operations followed by one fully connected layer compressing the information into a latent space. 1D convolutions are similar to 2D convolutions, commonly used in computer vision to detect patterns independent of their position on the image. Sliding 1D convolutions along the tabular representation of an ordered point cloud with TCP positions allows detecting TCP movement patterns, e.g., covered distances, speeds, accelerations and decelerations, and changes in the movement direction. As a result, a trained VAE can create meaningful projections of a multidimensional point cloud describing a milling process into a smaller latent space and generalize to unseen milling tasks.

3.4 RL setup, training and evaluation process

The RL setup is related to the main design choices in the state space, action space, reward function, training, and evaluation procedures. If required, an interested reader is invited to refer to the work of [38] for more details on the mentioned basic concepts of RL. *State space* is a vector that includes the VAE generated compact representation of the milling process for which the WP setup needs to be optimized. We include information regarding the current WP placement in the working space of the CNC milling machine: X, Y, Z coordinates and rotation angle. Additionally, information on the sum of axis accelerations (e_X, e_Y, e_Z), traveled distances (d_X, d_Y, d_Z), and any collisions are provided for the current WP setup. Besides, state space includes the embedding of ordered point cloud (D_1, D_2, D_3) for the current milling task generated by the encoder part of the trained VAE (see Sect. 3.3).

Action space is a vector of length five (ΔX, ΔY, ΔZ, ΔRot, search_stop) allowing to freely move the WP along the X-, Y-, and Z-axis and to change its orientation. The last element of the action space is designed to stop the search as soon as the RL agent assumes the current WP position represents a good solution.

Reward is a value defining the goodness of the solution. The reward function is designed in a way to depict the objectives of the optimization problem defined in subsection 3.2 and serves as the primary performance indicator in the study. The reward function is formulated as a weighted sum of milling process measures:

$$R = \begin{cases} 0.7(2e'_Z + e'_X) + 0.3(2d'_Z + d'_X), & \text{no collisions} \\ -1, & \text{collisions} \end{cases} \quad (6)$$

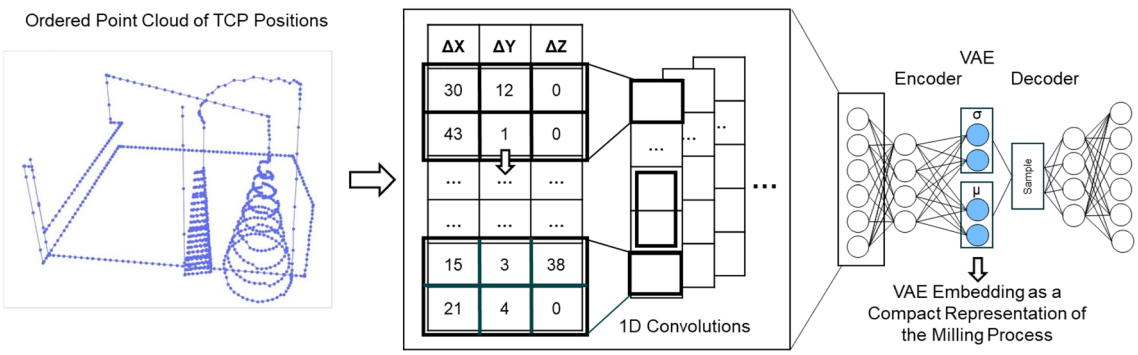


Fig. 4 VAE featuring 1D convolution layers trained on ordered point clouds of TCP positions enables the learning of compact representations of milling processes in an unsupervised manner

The given reward formulation is not designed to be universal. It can be adjusted by changing the weight coefficients and adding further terms. Properties of the CNC milling machine primarily dictate the weight coefficients (in our case, the Z-axis is heavier, therefore movement minimization along this axis is prioritized) and priorities defined by milling experts. In case of a changed reward function, the RL agent will need to be retrained to generate solutions satisfying the adjusted requirements.

The RL training is conducted in episodes. Within each episode a new WP is generated by randomly selecting the slot angle from the range [0°: 40] & [50°: 360°] and tower position from the range and $Z_{tower} \in (60mm, 235mm)$. The selected WP is placed in an arbitrary location and orientation in the CNC machine working space. A simulation run is conducted to detect possible collisions and characteristics of the milling process with the given WP setup. Based on the provided information, the agent can freely move and rotate the WP up to two times before the episode is terminated. After each WP setup change, the evaluation of the setup with a simulation run is repeated. As soon as the maximum number of optimization iterations is achieved or the RL agent activates the stop action, the current WP setup is considered to be the final solution and the episode ends. Only at the end of the episode the RL agent is provided with a non-zero reward calculated as defined in [6]. The use of sparse reward ensures that the agent concentrates on the quality of the final solution and not on the intermediate search steps.

A fixed set of testing WP geometries from the defined training range (*seen WP geometries*) is used to evaluate the current RL performance during the training process every 100,000 training steps. An RL agent from the training stage with the best performance on the test set is selected for further evaluations. The selected RL agent is validated on an *unseen WP geometry* with the slot angle SA = 45°, tower position $X_{tower} = 95mm$, and $Z_{tower} = 60\,mm$. It should be noted that for the evaluation of the generalization capabilities of the whole system to new WP setup optimization tasks, both VAE and RL agent do not get exposed to any WP with geometry from the range $SA \in [40° : 50°]$ and $X_{tower} \in [90mm : 100mm]$ at the training stage and during the selection of the best RL agent for validation.

4 Experimental results

4.1 Learning compact representations for WP geometries and related milling strategies

To train our VAE we create ordered point cloud representations of the milling process (as presented in Fig. 3) for 8,000 randomly generated WP geometries with $SA \in [0° : 360°]$, tower position $X_{tower} \in [55\,mm : 145\,mm]$ and $Z_{tower} \in (60\,mm, 235\,mm)$. Any WP geometries within the range of $SA \in [40° : 50°]$ and $X_{tower} \in [90mm : 100mm]$ are excluded from the training to eliminate potential information leakage related to the validation task.

The architecture of the VAE encoder consists of two 1D convolutional layers with 256 and 64 filters, respectively. Kernel size is set to five, and the average pooling of size two is applied after each convolutional layer. A fully connected layer compresses the information extracted by convolutional layers to a vector of size 70 before the last transformation to the intended embedding size is generated with the second fully connected layer. The proposed architecture is decided upon from a limited number of trial-and-error iterations and can be enhanced through more elaborate tuning.

The VAE training is executed for 200 epochs with a batch size of 32. All RL experiments in this study are conducted with the size of the latent space equal to three. The considered size of the latent space is sufficient for considered milling tasks and can be extended in the case of more complex WP geometries. To look into a learned latent space of a trained 1D VAE, we encode and visualize a set of ordered point cloud representations of milling processes (see Fig. 5).

In the provided example, only the slot orientation is varied, leading to a simple visualization of the latent space. A change of the slot angle results in a proportional shift of the corresponding WP representation in the latent space. This indicates the possibility of the proposed approach of learning meaningful representations of the WP geometries and related milling processes with no human supervision. The satisfactory RL training results presented in the next chapter

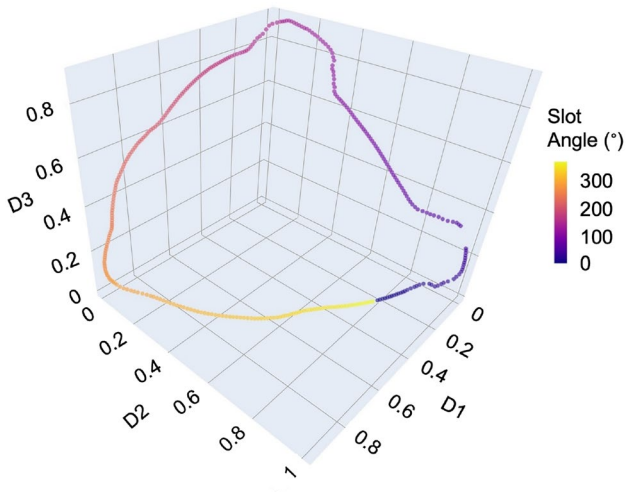


Fig. 5 Latent space encodings of the WP geometries and milling strategies with the encoder part of a trained 1D convolutional VAE. Each point represents a milling process of a separate WP. The only variation from WP to WP in this example is the orientation of the milling slot. The VAE encoder learns meaningful representations with a clear structure

prove that the assumption of a good WP representation holds for other variations of the WP geometry considered in this study's context.

4.2 WP setup optimization with RL

We conduct RL training and validation as described in subsection 3.4. A Soft Actor-Critic (SAC) RL agent [11] is used as it demonstrated promising results in previous works involving RL for WP setup optimization [7, 35]. One major difference is that the RL agent is not dependent on an explicit description of the WP geometry and milling strategy. Instead, a single simulation run represents the current milling task as an ordered point cloud encapsulating the TCP movement. Using our trained VAE encoder, the ordered point cloud is reduced to a compact representation used as a part of the state space. Another modification is the increase of the search space from $X \in [-200\,mm{:}0\,mm]$, $Y \in [-50\,mm{:}0\,mm]$ *and* $Z \in [-300\,mm{:}0\,mm]$ to $X \in [-400\,mm{:}400\,mm]$, $Y\,Z \in [-50\,mm{:}0\,mm]$ *and* $Z \in [-400\,mm{:}400\,mm]$ in order to bring it closer to the actual size of the working space of a CNC milling machine. The total number of training steps is set equal to 1,200,000. We chose the Stable Baselines RL framework [15] as a basis for the RL implementation in this work.

All evaluations are conducted on a validation WP geometry located in the middle of the range excluded from the VAE and RL agent training process: $SA \in [40° : 50°]$ and $X_{tower} \in [90\,mm : 100\,mm]$. Z_{tower} is set equal to $235\,mm$. A trained RL agent is required to solve 20 evaluation episodes with the previously unseen validation WP initialized with random orientations and positions in the working space of the CNC machine. To evaluate the stability of the RL training process and exclude particularly good or bad results by chance, the training and evaluation are independently repeated with three different random seeds defining the initialization of the RL actor and critic neural networks along with the WP positions and orientations.

We compare the quality of found solutions by looking at the resulting value of the optimization function (reward) representing the resulting axes collisions, traveled distances and accelerations. The same optimization function guides all compared methods. This is motivated by the multi-objective nature of the considered optimization task, which makes it hard to make a judgment on the quality of the solution by simply looking at separate parameters of found solutions. The resulting solutions are WP positions in the machine's workspace that are problematic to judge by the human expert without running a simulation. Even looking at accelerations and traveled distances along different machine axes is hard to compare directly because of the different importance weights for different machine axes. For this purpose, while analyzing experimental results we operate with the average reward value and show reward distributions to compare considered solution methods against each other directly. The absence of reward values equal to -1 ensures no axis collisions. The closer reward gets to 1, the smaller the overall axis movements and accelerations.

Figure 6 provides a comparison of the absolute performance for different solution methods. Each box plot encapsulates twenty independent runs with different WP initialization points. The first approach relies on the unmodified use of a trained RL agent that learned to avoid axes collisions and generate WP setups resulting in high rewards. Averaged over all attempts and random seeds, the RL agent achieves a reward of 0.924. Depending on the initial WP position, the RL agent tends to come up with different solutions resulting in noticeable deviations in solution quality.

WP setups found by a trained RL agent are close to the optimal solutions and can serve as a warm start for local search heuristics. We run an aftersearch with the mlrose [13] implementation of a hill-climbing algorithm [34] on top of solutions provided by the RL agent. The search is terminated as soon as five consecutive search iterations yield no improvements compared to the current solution. The combination of the RL agent with subsequent use of a local search results in an average solution quality improvement from 0.924 to 0.950.

The observed variation of the RL solutions depending on the WP setup initialization in the working space of the milling machine can be used to improve generated WP setups as demonstrated by [7]. In this case, the RL agent is asked to solve the validation tasks not only once but several times with different initializations. The solution with the highest reward is adopted as the recommended WP setup. We give the RL agent 20 attempts with different WP initializations and refer to the approach as the 20 attempts RL agent. As demonstrated in Fig. 6, this leads to an improvement of the average reward up to 0.948.

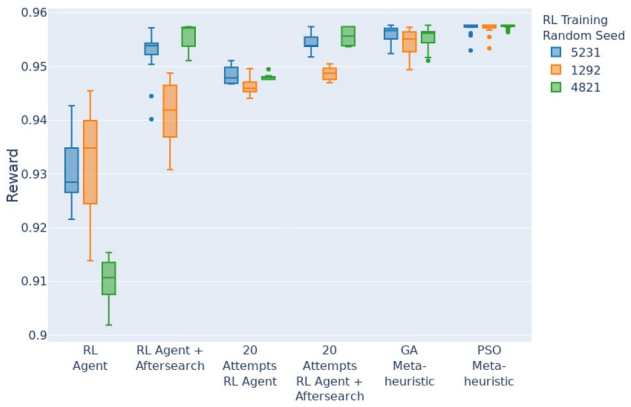


Fig. 6 Comparison of different solution methods for WP setup optimization

Both the aftersearch and the 20 attempts RL approaches are not mutually exclusive. Initialization points generated with the 20 attempts RL method can serve as a better warm start for the local search heuristic. Combining both methods proves to yield small improvements in the overall solution quality with an average reward of 0.953.

We chose GA and PSO metaheuristics to create a baseline for the optimization tasks. Both methods are commonly used for similar optimization tasks and are capable of yielding state-of-the-art results [12, 20]. We use the GA implementation from the mlrose and PSO implementation from the PySwarms Python libraries. Both methods are allowed to conduct up to 45,000 interactions with the milling environment to find a suitable WP setup that maximizes the reward considered in this work. Similar to the RL evaluation scheme, we conduct three sets of 20 independent runs for each metaheuristic. On average, GA and PSO heuristic surpass the absolute performance of most RL-based evaluation runs and achieve a mean reward of 0.955 and 0.957, respectively. Deviations of solution quality generated by the PSO heuristic are particularly low. We opt to run all benchmarking runs on the ML-based approximation of the milling process since the developed ML models demonstrate high levels of accuracy for the range of milling tasks considered in this work. Running metaheuristic optimizations directly on the SinuTrain simulation is considerably slower. Additionally, this ensures the comparability to results achieved with RL-based methods. Therefore it is essential to evaluate how long each of the proposed solutions would require in a scenario without the availability of fast ML approximation.

Figure 7 summarizes the results of quantitative comparative analysis of the RL-based methods with GA and PSO methods in terms of the number of search iterations required to generate comparable results and, as a result, the convergence speed. Figure 7a shows that a trained RL agent will require one initialization run of the simulation to represent the milling task and up to two WP setup changes controlled by the RL agent resulting in three simulation runs in total. The use of the 20 attempts RL approach results in an increased number of required search iterations, up to 40 simulation runs, as well as one initialization simulation. Adding on top of it heuristic aftersearch requires up to 50 additional iterations. Independently of the selected RL-based method, the average runtime of the search process with SinuTrain do not exceed 11.6 min.

Figure 7b demonstrates the number of search steps required for each metaheuristic run to match the average performance of a 20 attempts RL approach augmented with the aftersearch heuristic. Averaged over three random seeds, GA requires 11,160 interactions with the milling environment to match the performance of the given learning-based optimization methods. PSO demonstrates higher search efficiency even though it still requires 5,654 interactions on average with the milling environment. According to our estimations, one average GA run could require 30.5 h of computation time to solve a new WP setup task directly on SinuTrain. An average WP optimization run for the validation task with PSO could take 15.4 h. This by far exceeds the observed average runtime for 20 attempts RL approach augmented with the aftersearch heuristic.

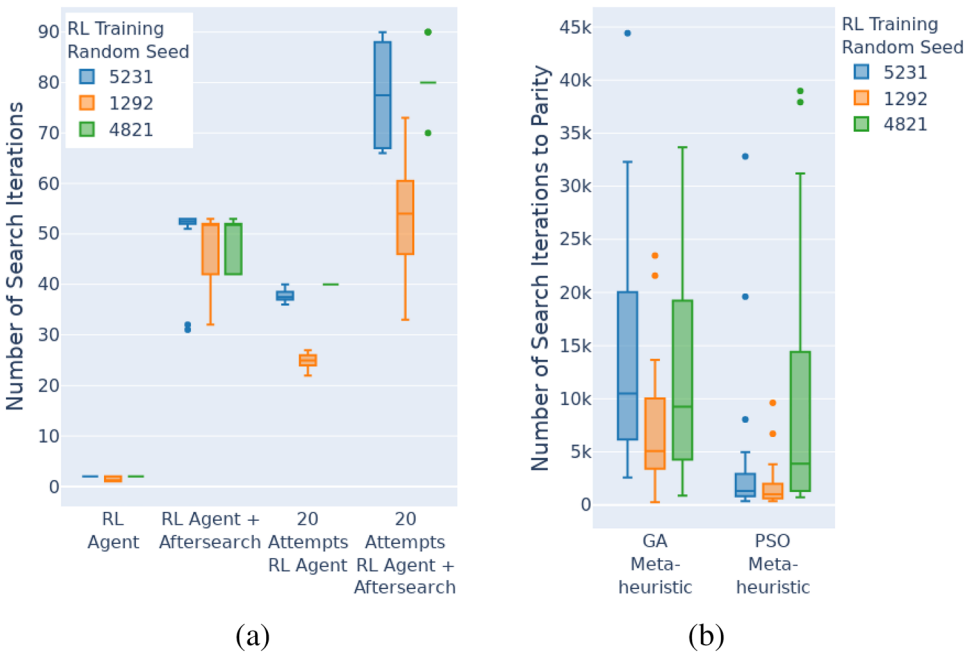


Fig. 7 Comparison of RL-based methods with GA and PSO heuristics in terms of search iterations

5 Conclusion

In this work, we investigated how to automate the selection process of WP setup in the working space of a CNC milling machine. Our approach builds on the works of [35] and [7] and demonstrates for the first time a fully automated learning-based solution with no dependence on expert knowledge. We suggest using an ordered point cloud as a 3D representation of a new milling process. The latent space encodings of a trained 1D convolutional VAE are demonstrated to deliver compact and generic representations of complex milling processes. Generated compact representations describe the considered milling process to the RL agent and enable it to propose a near-optimal WP positioning and orientation.

The provided evaluations demonstrate the following capabilities of learning-based methods augmented with simple search heuristics:

- Generating WP setups ensuring the axis collision avoidance, covering the axis limit violations, collisions between the tool assembly and the WP, as well as collisions between moving parts of the CNC machine. Therefore problem aspect 1 formulated in the introduction is addressed.
- Generating WP setups comparable to state-of-the-art metaheuristics in terms of minimization of total traveled distances of the machine axes and total axis accelerations, resulting in the fulfillment of problem aspects 2 and 3.
- Generalizing to new, previously unseen WP geometries with no expert input. As a result, problem aspect 4 is satisfied.
- Offering an order of magnitude increase in search efficiency by utilizing the RL agent's previous experience during the training phase and extrapolating it to new, previously unseen, milling tasks. Therefore, problem aspect 5 is addressed.

At the time, the achieved search efficiency of the final solution comes at the cost of a computationally intensive training phase. Such RL training is a challenging engineering task. Training directly on SinuTrain is infeasible due to long runtimes. The studies of [35] and [7] have demonstrated how to approximate realistic milling simulations with ML for a range of milling tasks. This study demonstrates that the combination of VAE with RL enables a generic understanding of complex milling tasks without requiring expert input.

However, all demonstrated results should be considered as proof of concept. Although the SinuTrain simulation is able to simulate traveled axes distances, axes accelerations and axes collisions very accurately, physical experiments are an addition to the validation scheme we aim for in the future. While having defined and tested the steps required for creating a learning-based optimization system, a production-ready solution would need to cover a far more extensive range of milling tasks and is only possible with considerably higher computational resources during the training phase. At the same time, a fully trained production-ready RL agent will have to remain lightweight in terms of the required computational resources and search-efficient enough to run on computationally intensive and realistic commercial milling simulations.

Acknowledgements We want to thank the company Siemens AG for the cooperation in this project.

Funding Open Access funding enabled and organized by Projekt DEAL. Open Access funding enabled and organized by project DEAL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Abe K, Xu Z, Sato I, et al (2019) Solving NP-hard problems on graphs with extended alphago zero. In: arXiv:1905.11623. http://arxiv.org/abs/1905.11623v2
2. Bello I, Pham H, Le QV, et al (2016) Neural combinatorial optimization with reinforcement learning. In: arXiv preprint arXiv:1611.09940
3. Bengio Y, Lodi A, Prouvost A (2021) Machine learning for combinatorial optimization: a methodological tour d'horizon. Eur J Oper Res 290(2):405–421. https://doi.org/10.1016/J.EJOR.2020.07.063
4. Campatelli G, Scippa A, Lorenzini L et al (2015) Optimal workpiece orientation to reduce the energy consumption of a milling process. Int J Prec Eng Manuf-Green Technol 2(1):5–13. https://doi.org/10.1007/s40684-015-0001-3
5. Cui H, Turan O, Sayer P (2012) Learning-based ship design optimization approach. Comput Aided Des 44(3):186–195. https://doi.org/10.1016/j.cad.2011.06.011
7. Enslin C, Samsonov V, Köpken HG, et al (2021) Optimisation of a workpiece clamping position with reinforcement learning for complex milling applications. In: Nicosia G, Ojha V, La Malfa E, et al (Eds) Machine Learning, Optimization, and Data Science, Lecture Notes in Computer Science, vol 13164. Springer

International Publishing, p 266–280, https://doi.org/10.1007/978-3-030-95470-3_20
8. Fertig A, Preis C, Weigold M (2022) Quality prediction for milling processes: automated parametrization of an end-to-end machine learning pipeline. Prod Eng. https://doi.org/10.1007/s11740-022-01173-4
9. Gannouni A, Samsonov V, Behery M, et al (2020) Neural combinatorial optimization for production scheduling with sequence-dependent setup waste. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp 2640–2647, https://doi.org/10.1109/SMC42975.2020.9282869
10. Grandguillaume L, Lavernhe S, Tournier C (2021) Optimal tool orientation in 3 + 2-axis machining considering machine kinematics. Int J Adv Manuf Technol 115(9–10):2765–2783. https://doi.org/10.1007/s00170-021-07036-z
11. Haarnoja T, Zhou A, Hartikainen K, et al (2018) Soft actor-critic algorithms and applications. In: arXiv:1812.05905
12. Hassan R, Cohanim B, de Weck O, et al (2005) A comparison of particle swarm optimization and the genetic algorithm. In: 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference. American Institute of Aeronautics and Astronautics, Reston, Virigina, https://doi.org/10.2514/6.2005-1897
13. Hayes G (2019) mlrose: Machine learning, randomized optimization and search package for Python. https://github.com/gkhayes/mlrose
14. Heisel U, Feinauer A (1999) Dynamic influence on workpiece quality in high speed milling. CIRP Ann 48(1):321–324. https://doi.org/10.1016/S0007-8506(07)63193-X
15. Hill A, Raffin A, Ernestus M, et al (2018) Stable baselines
16. Hu P, Tang K (2011) Improving the dynamics of five-axis machining through optimization of workpiece setup and tool orientations. Comput Aided Des 43(12):1693–1706. https://doi.org/10.1016/j.cad.2011.09.005
17. Hu SJ (2013) Evolving paradigms of manufacturing: from mass production to mass customization and personalization. Procedia CIRP 7:3–8. https://doi.org/10.1016/j.procir.2013.05.002
18. Lesort T, Díaz-Rodríguez N, Goudou JFI et al (2018) State representation learning for control: an overview. Neural Networks 108:379–392. https://doi.org/10.1016/j.neunet.2018.07.006
19. Li Y, Lei G, Bramerdorfer G et al (2021) Machine learning for design optimization of electromagnetic devices: recent developments and future directions. Appl Sci 11(4):1627. https://doi.org/10.3390/app11041627
20. Lim SP, Haron H (2013) Performance comparison of genetic algorithm, differential evolution and particle swarm optimization towards benchmark functions. In: 2013 IEEE Conference on Open Systems (ICOS). IEEE, pp 41–46, https://doi.org/10.1109/ICOS.2013.6735045
21. Liu H, Fu Z, Yang K et al (2021) Machine learning for glass science and engineering: a review. J Non-Cryst Solids 557(119):419. https://doi.org/10.1016/j.jnoncrysol.2019.04.039
22. Liu R, Nageotte F, Zanne P et al (2021) Deep reinforcement learning for the control of robotic manipulation: a focussed mini-review. Robotics 10(1):22. https://doi.org/10.3390/robotics10010022
6. Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: Yoshua Bengio, Yann LeCun (Eds) 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, 14–16, Conference Track Proceedings
23. Manchanda S, Mittal A, Dhawan A, et al (2019) Learning heuristics over large graphs via deep reinforcement learning. In: arXiv:1903.03332
24. Mazyavkina N, Sviridov S, Ivanov S, et al (2020) Reinforcement learning for combinatorial optimization: a survey. In: arXiv preprint arXiv:2003.03600
25. Mirhoseini A, Goldie A, Yazgan M et al (2021) A graph placement methodology for fast chip design. Nature 594(7862):207–212. https://doi.org/10.1038/s41586-021-03544-w
26. Nazari M, Oroojlooy A, Snyder LV, et al (2018) Reinforcement learning for solving the vehicle routing problem. In: arXiv:1802.04240
27. Park HS, Nguyen DS, Le-Hong T et al (2022) Machine learning-based optimization of process parameters in selective laser melting for biomedical applications. J Intellig Manuf 33(6):1843–1858. https://doi.org/10.1007/s10845-021-01773-4
28. Pessoles X, Landon Y, Segonds S et al (2013) Optimisation of workpiece setup for continuous five-axis milling: application to a five-axis BC type machining center. Int J Adv Manuf Technol 65(1–4):67–79. https://doi.org/10.1007/s00170-012-4151-y
29. Pierrot T, Ligner G, Reed S, et al (2019) Learning compositional neural programs with recursive tree search and planning. In: advances in neural information processing Systems (NeurIPS)
30. Rangarajan A, Dornfeld D (2004) Efficient tool paths and part orientation for face milling. CIRP Ann 53(1):73–76. https://doi.org/10.1016/S0007-8506(07)60648-9
31. Recht B (2019) A tour of reinforcement learning: the view from continuous control. Ann Rev Control Robot Auton Syst 2(1):253–279. https://doi.org/10.1146/annurev-control-053018-023825
32. Ren F, Hb Hu, Tang H (2020) Active flow control using machine learning: a brief review. J Hydrodyn 32(2):247–253. https://doi.org/10.1007/s42241-020-0026-0
33. Rosati R, Romeo L, Cecchini G et al (2023) From knowledge-based to big data analytic model: a novel iot and machine learning based decision support system for predictive maintenance in industry 4.0. J Intellig Manuf 34(1):107–121. https://doi.org/10.1007/s10845-022-01960-x
34. Russell SJ, Norvig P (1995) Artificial intelligence: a modern approach. Prentice Hall series in artificial intelligence, Prentice Hall, Upper Saddle River
35. Samsonov V, Enslin C, Köpken HG, et al (2020) using reinforcement learning for optimization of a workpiece clamping position in a machine tool. In: Proceedings of the 22nd International Conference on Enterprise Information Systems. SCITEPRESS - Science and Technology Publications, pp 506–514, https://doi.org/10.5220/0009354105060514
36. Schwarz EB, Bleier F, Guenter F et al (2022) Improving process monitoring of ultrasonic metal welding using classical machine learning methods and process-informed time series evaluation. J Manuf Processes 77:54–62. https://doi.org/10.1016/j.jmapro.2022.02.057
37. Shaw D, Ou GY (2008) Reducing and axes movement of a 5-axis AC type milling machine by changing the location of the workpiece. Computer-Aided Design 40(10–11):1033–1039. https://doi.org/10.1016/j.cad.2008.09.001
38. Sutton RS, Barto A (2018) Reinforcement learning: An introduction, second, edition. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts and London, England
39. Tang TD (2014) Algorithms for collision detection and avoidance for five-axis NC machining: a state of the art review. Comput Aided Des 51:1–17. https://doi.org/10.1016/j.cad.2014.02.001
40. Tercan H, Meisen T (2022) Machine learning and deep learning based predictive quality in manufacturing: a systematic review. J Intellig Manuf 33(7):1879–1905. https://doi.org/10.1007/s10845-022-01963-8
41. Vesselinova N, Steinert R, Perez-Ramirez DF, et al (2020) Learning combinatorial optimization on graphs: a survey with applications to networking. IEEE Access 8:120,388–120,416. https://doi.org/10.1109/ACCESS.2020.3004964

42. Weber J, Risse S, Laroque C (2018) Workpiece positioning based on supervised learning methods for simulation-based optimization of virtual tooling processes. In: 2018 Winter Simulation Conference (WSC). IEEE, pp 3168–3179, https://doi.org/10.1109/WSC.2018.8632523
43. Wei C, Lee W (2020) Optimization of the setup position of a workpiece for five-axis machining to reduce machining time. Adv Mechan Eng. https://doi.org/10.1177/1687814020975544
44. Weichert D, Link P, Stoll A et al (2019) A review of machine learning for the optimization of production processes. Int J Adv Manuf Technol 104(5–8):1889–1902. https://doi.org/10.1007/s00170-019-03988-5
45. Zhang C, Song W, Cao Z, et al (2020) Learning to dispatch for job shop scheduling via deep reinforcement learning. In: Advances in Neural Information Processing Systems, pp 1621–1632
46. Zhang Y, Yan W (2022) Applications of machine learning in metal powder-bed fusion in-process monitoring and control: status and challenges. J Intellig Manuf pp 1–24. https://doi.org/10.1007/s10845-022-01972-7
47. Zhuang L, Xu A, Wang XL (2023) A prognostic driven predictive maintenance framework based on bayesian deep learning. Reliab Eng Syst Safety 234(109):181. https://doi.org/10.1016/j.ress.2023.109181

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.