



Tool remaining useful life prediction using deep transfer reinforcement learning based on long short-term memory networks

Jiachen Yao¹ · Baochun Lu¹ · Junli Zhang¹

Received: 23 March 2021 / Accepted: 23 August 2021 / Published online: 13 September 2021
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

Tool wear and faults will affect the quality of machined workpiece and damage the continuity of manufacturing. The accurate prediction of remaining useful life (RUL) is significant to guarantee the processing quality and improve the productivity of automatic system. At present, the most commonly used methods for tool RUL prediction are trained by history fault data. However, when researching on new types of tools or processing high value parts, fault datasets are difficult to acquire, which leads to RUL prediction a challenge under limited fault data. To overcome the shortcomings of above prediction methods, a deep transfer reinforcement learning (DTRL) network based on long short-term memory (LSTM) network is presented in this paper. Local features are extracted from consecutive sensor data to track the tool states, and the trained network size can be dynamically adjusted by controlling time sequence length. Then in DTRL network, LSTM network is employed to construct the value function approximation for smoothly processing temporal information and mining long-term dependencies. On this basis, a novel strategy of Q-function update and transfer is presented to transfer the deep reinforcement learning (DRL) network trained by historical fault data to a new tool for RUL prediction. Finally, tool wear experiments are performed to validate effectiveness of the DTRL model. The prediction results demonstrate that the proposed method has high accuracy and generalization for similar tools and cutting conditions.

Keywords Tool wear monitoring · Remaining useful life (RUL) prediction · Deep reinforcement learning · Transfer learning

1 Introduction

Tool is a key part in manufacturing process, including turning, milling, and cutting. Tool wear in manufacturing process affects the machining performance and reduces the productivity of high-speed computer numerical control (CNC). Thus, effective tool wear monitoring and remaining useful life (RUL) prediction are of great significance for improving machining quality and predictive maintenance [1–3]. Generally, tool

monitoring and RUL prediction methods can be roughly classified into statistical model-based and data-driven methods.

In the statistical model-based methods, the core idea is to establish a failure mechanisms model for RUL prediction on the basis of stochastic process. Si et al. [4] developed a Wiener process-based degradation model for RUL prediction, and recursive filter was used to reduce the estimation error. Yan et al. [5] designed a stage-based Gamma process to predict the probability density function of tool unobservable degradation. Wang et al. [6] presented a particle filtering method for tool wear state prediction.

In the data-driven methods, machine and deep learning approaches are used to process the observation data for diagnosis and prognosis [7]. In this kind of method, vibration sensors, torque sensors, or other kinds of sensors are installed on machining centers to monitor tool working states. Sensory signals are extracted by signal processing technology to get discriminant signal features [8–10]. Due to the advantages of high prediction accuracy and easy modeling, data-driven

✉ Jiachen Yao
791344334@qq.com

Baochun Lu
nlgbbc@126.com

Junli Zhang
maysevenhero@126.com

¹ Nanjing University of Science and Technology, Nanjing, China

methods have been a research hotspot for tool state monitoring and RUL prediction. For instance, Widodo et al. [11] reviewed the implement of support vector machine (SVM) in machine condition monitoring and diagnosis. Chen et al. [12] utilized logistic regression model to process vibration signals for cutting tool monitoring. Karandikar et al. [13] evaluated the performance of two different machine learning methods in predicting tool life curve. Yang et al. [14] established a v -support vector regression (v -SVR) model to study the relationship between fused features and actual tool wear for tool wear monitoring. Zhang et al. [15] used a least square support vector machine (LS-SVM) to predict tool wear of cutting edge position under joint effect of machining conditions. Kong et al. [16] presented a Gaussian process regression technique for accurately monitoring flank wear width. Kong et al. [17] developed a Gaussian mixture hidden Markov models to determine the tool wear states. Zhou et al. [18] utilized extension neural networks (ENNs) to fast recognize cutting tool conditions with high precision.

The machinery health prognostic program generally follows a similar technical process: first is extracting artificially designed features from acquired signals for determining the state change of tool, and then establishing the nonlinear mapping function between extracted features and tool state by regression methods. But in the above methods, there are two main shortcomings in artificial neural network (ANN)-based fault prognosis approaches. First, the inputs rely heavily on signal preprocessing techniques. Second, the simple architecture of ANNs lacks sufficient breadth and depth to map complex nonlinear relationship. The development of deep learning has relieved the above problems to a certain extent [19, 20]. Deep learning can adaptively learn hierarchical representation without extracting the fault features manually [21, 22], which is beneficial to improve the adaptability of the model. In addition, more hidden layers are added to process nonlinear inputs, which is more likely to learn deeper hidden information and then to improve prediction accuracy. Deep learning models have attracted increasing attention in fault diagnosis and prognosis. Jia et al. [23] designed a deep neural network (DNNs)-based method for fault diagnosis in rolling element bearings and planetary gearboxes. Shao et al. [24] constructed a convolutional deep belief network for fault diagnosis of rolling bearing, which used compressed sensing (CS) for reducing the amount of data. Wu et al. [25] utilized bidirectional long short-term memory neural network (BiLSTM) to deal with singular value decomposition features to predict current tool wear value. Zhao et al. [26] proposed a deep residual network with dynamically weighted wavelet coefficients for planetary gearbox fault diagnosis.

Different from above-mentioned approaches, deep reinforcement learning can directly map raw extracted features to the corresponding tool wear state, which is helpful to further improve intelligence of prediction

methods. Combining the advantages of deep learning and reinforcement learning, deep reinforcement learning is able to construct the environment according to extracted features, from which artificial agents can learn observations and rewards. Reinforcement learning gives agents the ability to interact with its environment, while deep learning enables agents to learn the better decisions to scale to problems with high-dimensional state and action spaces [27]. As the most significant breakthrough in the field of artificial intelligence, AlphaGo proves the effectiveness of DRL mechanism [28]. Since then, DRL algorithms have been widely applied in the domain of modern manufacturing systems, natural language processing, and automated machine learning. In modern manufacturing systems, as a common solution for optimization problems by trial and error, DRL has already been used in fields such as robot training [29, 30], management of Industrial Internet of Things [31, 32], dynamic scheduling of flexible job shop [33, 34], and machinery fault diagnosis [35], while how to transfer a DRL network to an effective application against the limited availability of training data for RUL prediction is still a hotspot issue in accurate prediction of tool RUL.

To overcome the deficiencies of limited data and to further improve the accuracy and intelligence of prediction methods, a deep reinforcement transfer learning (DTRL) method is researched in this paper. Two optimization strategies, including value function approximation through LSTM, Q-function update and transfer, are researched to realize the transfer of a trained DRL network to a new application scene. In DTRL method, local features are first extracted from consecutive time series data to reduce the network size. Then in DRL prediction method, LSTM network is adopted to construct the value function approximation for deeply mining temporal information. A novel strategy of Q-function update and transfer has been proposed to guarantee transferability of trained network to new domain. Finally, tool wear experiments are carried out and the effectiveness of the proposed method is verified by analyzing the datasets.

The rest of this paper is organized as follows. Theoretical foundation about DRL is introduced in Section 2, based on which the framework of proposed DTRL method is shown in Section 3. Tool wear experiments and case study on RUL prediction are conducted in Section 4. Model comparison and validation are shown in Section 5. Finally, the conclusions are summarized in Section 6.

2 Theoretical foundation

Deep reinforcement learning (DRL) is a branch of dynamic programming-based reinforcement learning, in which agents

interact with the environment while learning. The interactive learning process can be modeled by Markov decision process (MDP) expressed by a tuple:

$$M = (S, A, T, R, \gamma) \tag{1}$$

where S is a finite set of states, A is a finite set of actions, $T : S \times A \times S \rightarrow [0, 1]$ is the transition function, $R : S \times A \times S \rightarrow \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor. $\pi : S \times A \rightarrow [0, 1]$ is a deterministic policy to demonstrate the probability of the action. The state-action value function $Q : S \times A \rightarrow \mathbb{R}$ following policy π can be defined as:

$$Q^\pi(s, a) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_{t+k} | s_k = s, a_k = a \right] \tag{2}$$

The goal of each MDP is to find an optimal policy π^* , and it owns expected return $V^*(s)$ and value function $Q^*(s, a)$. The Q-function in DRL satisfies the Bellman optimality equation. Therefore, an optimal state-action value satisfies the following equation:

$$Q^*(s, a) = \sum_{s' \in S} T(s, a, s') (R(s, a, s') + \gamma V^*(s')) \tag{3}$$

One of the most popular methods to estimate the value of state action is the Q learning algorithm. The basic idea of deep Q learning is to estimate Q -values based on rewards and the agent’s Q -value function. The Q -update rule for model-free online learning can be expressed as:

$$Q'(s, a) = Q(s, a) + \alpha \left[R + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \tag{4}$$

where α is the learning rate. The max error is utilized to evaluate the quality of Q-function:

$$error = \left(R + \gamma \left| \max_{a'} Q(s, a) - Q^*(s, a) \right| \right) \tag{5}$$

3 Proposed DTRL architecture

In the practical application of deep learning, there are two common problems: the first is to deal with extremely large state space of tabular Q learning in time series analysis; the second is to process unlabeled data. To solve the above issues, the DTRL architecture is designed, which combines deep learning with transfer reinforcement learning. More specifically, the DTRL method inputs the current state and action, then adopts a LSTM to estimate the value of $Q(s, a)$, and at last

transfers the Q -values to another LSTM network. The estimated value of $Q(s, a)$ is defined as:

$$Q(s, a) = \mathbb{E} \left[R + \gamma \max_{a'} Q(s', a') \right] \tag{6}$$

3.1 Parameter reinforcement Q learning

To solve the problem caused by large state space in the RUL prediction and improve the generalization ability of deep Q-function, Deep Q-Network (DQN) is adopted. DQN is a model parameterized by weights and biases collectively denoted as θ . In DQN, Q -values at each training iteration t can be denoted by $Q_{\theta_t}(s, a)$. More specifically, Q -values are estimated by performing forward propagation then querying the output nodes. To obtain the estimation of Q -values shown in Eq. (6), the proposed DTRL method adopts the experience replay method [36]. After one prediction, the experiences at current time step, denoted as $e_t = (s, a, R, s')$, are recorded in the replay memory $M = \{e_1, e_2, \dots, e_t\}$, and then sampled randomly at training time. Instead of updating Q-table lookups, now the network parameters θ are updated with stochastic gradient algorithm to minimize the differentiable loss function:

$$L(\theta_t) = \mathbb{E}_{(s, a, r, s')} \left[\left(R + \gamma \max_{a'} Q_{\theta_t}(s', a') - Q_{\theta_t}(s, a) \right)^2 \right] \tag{7}$$

When the Q-function changes quite rapidly, the updates may oscillate or diverge. At the same time, when there are too many iterations, the algorithms will be inefficient. To avoid the above problems, the proposed DTRL method adopts the fixed Q-targets method. Instead of using the latest parameters θ_t to calculate the maximum possible reward of the next state $\gamma \max_{a'} Q_{\theta_t}(s', a')$, we update the parameters θ' every certain iteration. Differentiating the loss function with respect to the parameters, the gradient is shown as follows:

$$\nabla_{\theta_t} L(\theta_t) = \mathbb{E}_{(s, a, r, s')} \left[(y_{\theta_t} - Q_{\theta_t}(s, a)) \nabla_{\theta_t} Q_{\theta_t}(s, a) \right] \tag{8}$$

where $y_{\theta_t} = R + \gamma \max_{a'} Q_{\theta_{t'}}(s', a')$ is the stale update target.

3.2 Deep Q-network based on LSTM

In the DQN-based RUL prediction method, we use densely connected networks to capture the state correlations. However, the real-word RUL prediction tasks also exist temporal correlations and vanishing gradient problems, which may result in degradation of DQN’s performance. Therefore, in the proposed DTRL method, LSTM layers are adopted instead of dense layers to carry information across many timesteps. More specifically, the Q-function in the DTRL-based RUL prediction can be defined as:

$$Q_{\theta_t}(s, a, h_{t-1}) \tag{9}$$

where h_{t-1} is the supplementary input calculated by LSTM layers according to the previous information. Consequently, the gradient of the loss function is shown as:

$$\nabla_{\theta_t} L(\theta_t) = \mathbb{E}_{(s,a,r,s')} [(y_{\theta_{t-1}} - Q_{\theta_t}(s, a, h_{t-1})) \nabla_{\theta_t} Q_{\theta_t}(s, a, h_{t-1})] \tag{10}$$

$$y_{\theta_{t-1}} = R + \gamma \max_{a'} Q_{\theta_t}(s', a', h'_{t-1}) \tag{11}$$

The DTRL method adopts LSTM layers instead of simple dense layers to process temporal series. As shown in Fig. 1, LSTM layers take as input temporal sequential experiences $e = \{e_1, e_2, \dots, e_t\}$ and Q-values are calculated after the output layer. In the practical prediction process, as LSTM layers save multi-timesteps information, we are supposed to choose experiences traces with certain length instead of single experience.

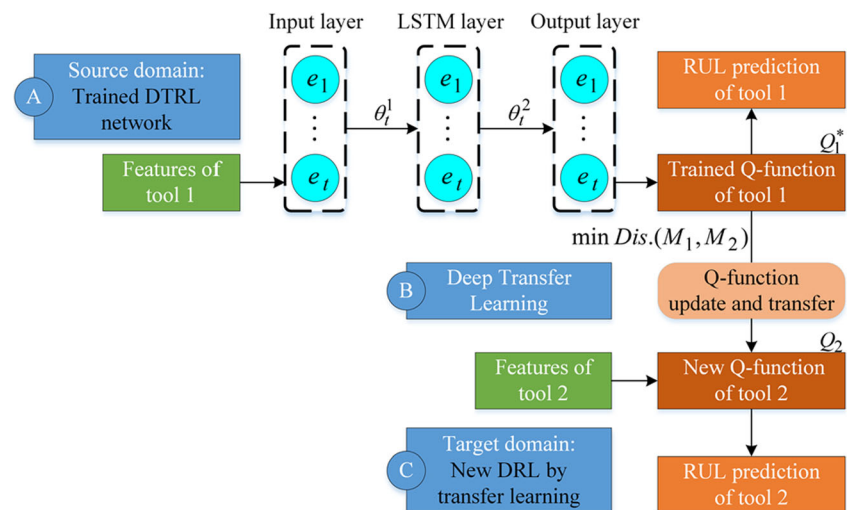
3.3 Deep Q transfer learning

After deep reinforcement Q learning, we can calculate the Q-function of each state-action pair, which can help to select the values with the least error in the prediction tasks. But the above algorithms require a large amount of experimental fault datasets, which are almost impossible to be acquired at the beginning of practical manufacturing. Hence, the transfer learning method is introduced to reduce the amount of training datasets and make full use of the trained Q-function.

In the RUL prediction tasks, the proposed DTRL method transfers the Q-function calculated from different tool tasks (source domain) to another tool task (target domain), which aims to improve the learning ability in new prediction tasks by introducing knowledge from a similar learned prediction task.

According to the DRL, the tool RUL prediction tasks can be defined as $M = (S, A, T, R, \gamma)$, and the tasks are different in transition function T , reward function R , and discount factor γ .

Fig. 1 Architecture of the deep transfer reinforcement learning network



As shown in Fig. 1, the source domain, denoted as $M_1 = (S, A, T_1, R_1, \gamma_1)$, is the trained DRL network, and the target domain, denoted as $M_2 = (S, A, T_2, R_2, \gamma_2)$, is the new DRL network. Assume Q_1^* and Q_2^* are corresponding optimal Q-functions. The main goal of the DTRL method is using the information of M_1 and Q_1^* to update Q_2 and improving the training speed of M_2 while ensuring the prediction accuracy.

To learn similar and joint Q-functions for the source domain and the target domain, the distance between Q-functions calculated by two networks is minimized. The distance between two tasks is defined as:

$$\begin{aligned} Dis.(M_1, M_2) &= \max_{s,a} |Q_1^*(s, a) - Q_2^*(s, a)| \\ &= \|Q_1^* - Q_2^* + Q_3^* - Q_4^* + Q_4^* - Q_2^*\|_{\infty} \leq \frac{\|R_1 - R_2\|_{\infty}}{1 - \gamma_1} \\ &\quad + \frac{\gamma_2 \|R_2\|_{\infty}}{(1 - \gamma_1)^2} \|T_1 - T_2\|_{\infty} + \frac{|\gamma_1 - \gamma_2|}{(1 - \gamma_1)(1 - \gamma_2)} \|R_2\|_{\infty} \end{aligned} \tag{12}$$

By minimizing the distance between two tasks, Q-functions learned in the target domain are restricted to be similar to those in the source domain, consequently deep Q transfer learning is achieved. According to the distance between two tasks calculated in Eq. (12), the Q-function update in the new DRL network is performed. Finally, the forward iteration process is implemented in the new DRL network, and the RUL prediction results of the target domain are presented.

The DTRL network is realized through above-mentioned steps, and the whole architecture is shown in Fig. 1.

3.4 DTRL for RUL prediction

In the real prediction tasks, the first step is feature extraction and data preprocess. We model the mapping function between the extracted features and the corresponding RUL sampling points, which can be expressed as:

$$\begin{bmatrix} f_{11} & f_{12} & \dots & f_{1m} \\ f_{21} & f_{22} & \dots & f_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \dots & f_{nm} \end{bmatrix} \Rightarrow \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix} \tag{13}$$

where $[f_1, f_2, \dots, f_m]$ denote the extracted features with m denoting numbers of input features at a certain point in time, and $[t_1, t_2, \dots, t_n]^T$ denote the RUL with n denoting sampling time points during run-to-failure. In the real tool processing, the tool performance is generally nonlinear due to the influence of nonlinear factors such as crack growth in material and sudden change of machining parameters. Therefore, a nonlinear activation function $\phi(*)$ is first adopted to fit mapping function between feature matrix and RUL series, and then linear regression is conducted on the feature. The nonlinear mapping can be defined as:

$$RUL = \varphi(F)\omega + b \tag{14}$$

where $RUL=[t_1, t_2, \dots, t_n]^T$ are time sample series, $\phi(F)$ is a nonlinear feature matrix, ω is the corresponding weights for nonlinear regression, and b is the bias.

In kernel-based method [37], weights can be expressed by feature samples as $\omega = \varphi(F)^T\alpha$ with α denoting representation coefficient. Then the product can be transformed as:

$$\varphi(F)\omega = \varphi(F)\varphi(F)^T\alpha = K\alpha \tag{15}$$

where K is a kernel matrix. Equation (14) can be defined as:

$$RUL = K\alpha + \mu \tag{16}$$

where the coefficient and the bias can be calculated by the least square method.

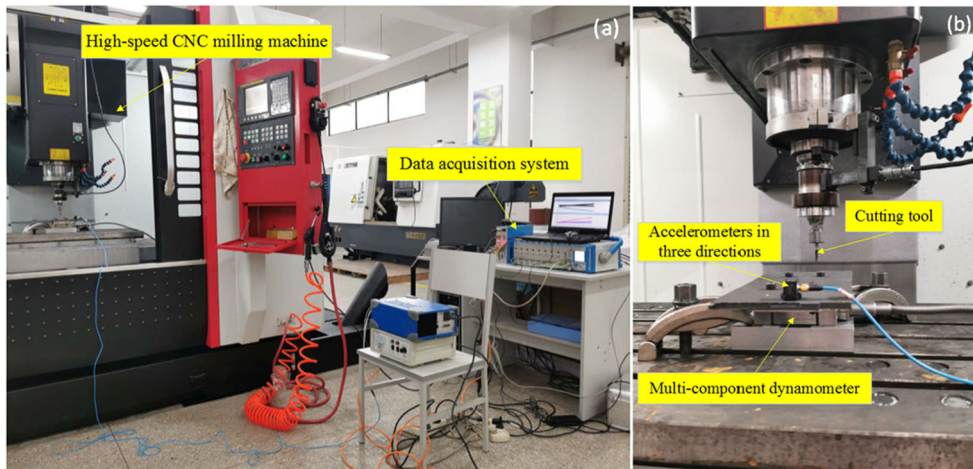
The DTRL-based RUL prediction algorithm is summarized in Algorithm 1.

4 Experiment validation

4.1 Benchmarking data description

To validate effectiveness of the proposed DTRL method, extensive experiments on tool performance degradation were conducted. The experiments were carried out on a high-speed CNC machine, as shown in Fig. 2. The cutting tools were end-milling cutters with 4 teeth, and the diameter and the length were 6mm and 50mm respectively, as shown in Fig. 3 b. The workpiece was steel alloy sheet and the material was CR12moV. The key signals collected synchronously were cutting force and vibration signals. The Kistler dynamometer was installed under the workpiece to collect cutting force signals by monitoring the workpiece. Meanwhile, the vibration signals were collected by the same method through installing the accelerometer on the workpiece and the radial vibration of the cutting tools was collected. Other experimental conditions for tool processing were as follows: The spindle rotation frequency was 3000 r/min; the feed rate was 1200 mm/min; the cutting depth in radial and axial direction was 0.5mm; the sampling frequency was 10kHz, and a run-to-failure tool produced a total of 312 cutting segments. A Kistler compact multi-component dynamometer 9129AA was mounted to collect cutting force signals in real time. A DAQ Elsys TraNET 404S8 and a triaxial accelerometer were used to collect vibration signals synchronously, and the measuring range and the frequency range were 50 g pk and 10k Hz. The tool was considered to be failure when the tool wear range was over 0.3 mm. Six kinds of sensor signals were acquired, including force in three directions and vibration in three directions. As shown in Fig. 3, the flank wear of the cutting tools was measured by a digital measuring microscope INSIZE ISM-WF200. Two sets of tool wear experiments were carried out: data sampled from tool1 were used for training, and data sampled from tool2 were used for testing. The cutting parameters kept constant to ensure the stability of the external

Fig. 2 CNC machine monitoring systems for tool wear data acquisition. **a** CNC machine tool. **b** Tool wear monitoring system



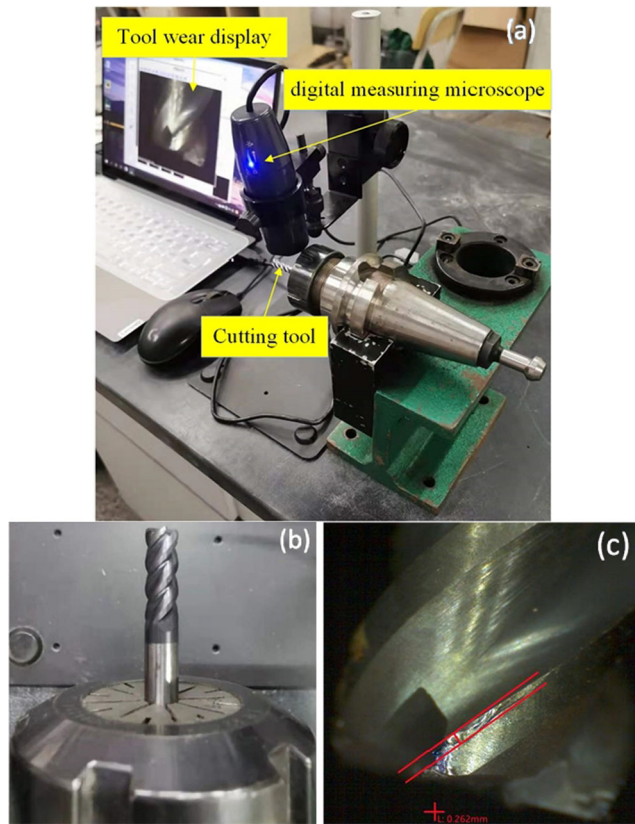


Fig. 3 Tool wear. **a** Tool wear monitoring system. **b** Tool used for processing. **c** Tool wear measurement

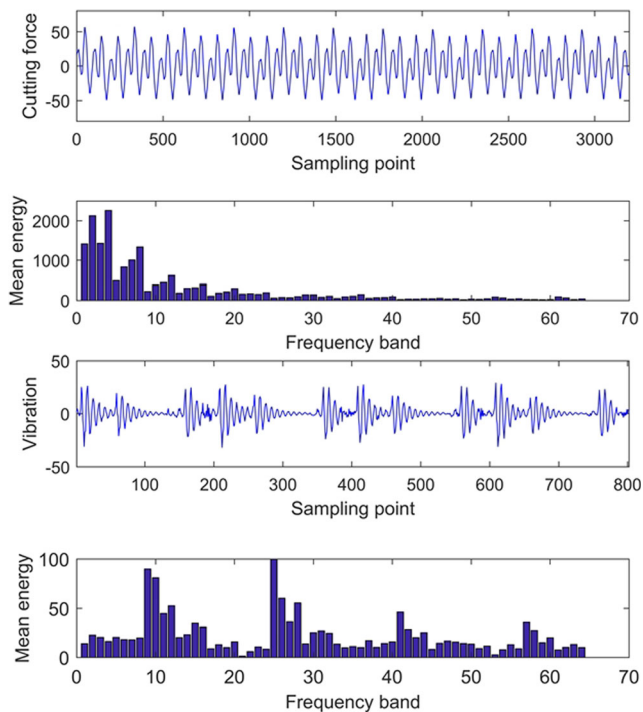


Fig. 4 Cutting force signals, vibration signals, and their energy in x direction calculated by OWPT

environment. The degree of tool wear constantly improved with the increase of cutting steps, which was revealed in the change of collected signals.

Algorithm 1 RUL prediction based on DTRL

1. Data Preprocessing:

Input: Historical failure datasets H_{nm} of a tool.

- 1) Extract features F_{nm}^* from acquired datasets.
- 2) Take data standardization and regularization on extracted features.

Output: Standard datasets F_{nm} as input of the DTRL.

2. Algorithm Training:

Input: Standard training and validation datasets and RUL series $RUL = [t_1, t_2, \dots, t_n]^T$.

- 1) Train a deep reinforcement learning network with training feature datasets.

Initialize memory replay D

Initialize DQN with random weights θ

For episode = 1, M **do**

Initialize feature sequence F_{nm}

For $t = 1, T$ **do**

Select action a_t with ϵ -greedy algorithm

Execute action a_t and observe reward r_t and new feature f_{t+1}

Store transition (f_t, a_t, r_t, f_{t+1}) in D

Sample random minibatch of transition from D

Perform a gradient descent step on loss function calculated by (10)

End For

End For

- 2) Train mapping model with training feature datasets and time series.

Output: The trained DRL network and mapping model.

3. RUL Prediction:

Input: Feature datasets of a new tool.

- 1) Transfer the trained Q-function to the new DRL network and then perform update to get the new Q-function Q_2 .

- 2) Input the new feature training datasets preform forward propagation by the new Q-function to achieve feature prediction.

- 3) Input the predicted feature matrix and realize the RUL prediction of the new tool by trained nonlinear mapping function.

Input the predicted feature matrix to the trained nonlinear mapping function for RUL prediction of the new tool.

Output: Feature and RUL prediction of the new tool.

4.2 Data preprocessing and DRL training

In order to make the raw sensor signal more aenable to models, the first step is data preprocessing, including feature extraction and normalization. As the energy features can

effectively reflect the tool wear state [38, 39], orthogonal wavelet packet transform (OWPT) is used to extract the energy features of cutting force and vibration signals in three directions for comparisons. OWPT shows a good effect on noise elimination and dimension reduction, which can improve the prediction accuracy and calculating speed. In this paper, to acquire discriminated feature, six-level OWPT using db1 is adopted to calculate the energy of each sub-band. Each feature can be separated into 64 sub-bands, of which the energy is defined as

$$E_i = \sum_{m=1}^n |x_{i,m}| \tag{17}$$

where $x_{i,m}$ is wavelet coefficient in scale 2^i , and n is the oscillation parameter. The magnitude and the distribution of energy can effectively reflect the tool wear state. The energy spectrum of cutting force and vibration signal in each cutting segment can be converted into the form shown in Fig. 4, in which the energy features are dimensionless indicators. Then, each energy feature is normalized independently to accelerate model convergence.

After data preprocessing, frequency spectrum of the signals monitored from tool wear process is extracted as weighted average energy, which is used as input to DTRL for model training and RUL prediction. To configure network structure of the first Q-function of tool1, the number of nodes in both input and output layers is set to be 6, which is equal to the number of extracted features. Some experiments were conduct

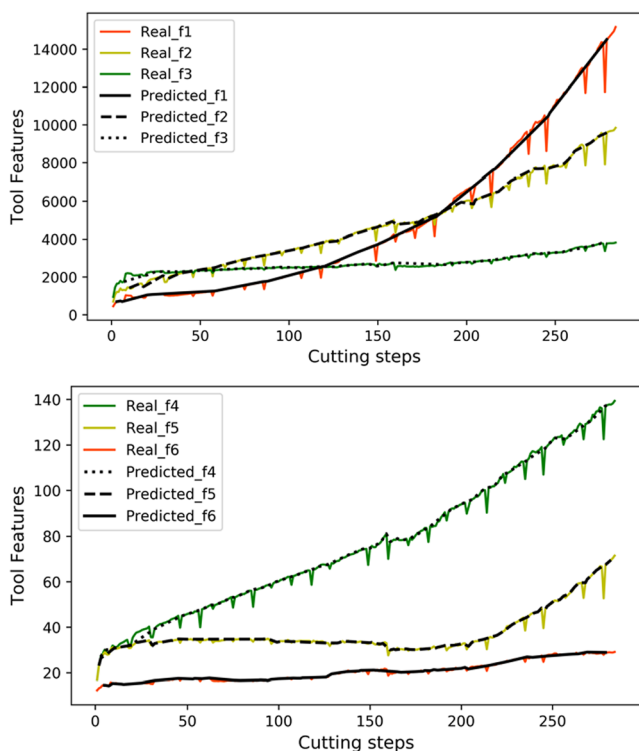


Fig. 5 Predicted features of tool2 learned by the DTRL network

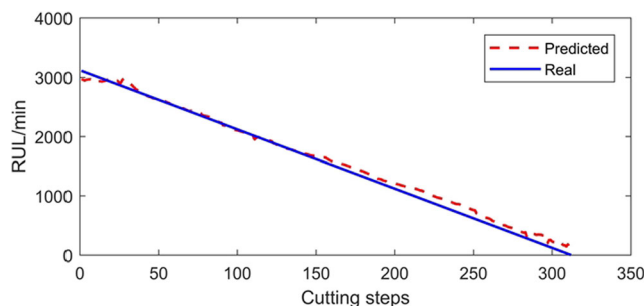


Fig. 6 RUL prediction result for tool2

to evaluate the state and reward settings and the discount factor γ is set to 0.9 and the learning rate α is set to 0.05 in Eq. (10). In order to ensure that the training results converge to the global optimum, the learning rate is set to decrease exponentially with the number of training epochs. Then to accelerate the convergence speed of network in training process, RMSprop is used to optimize gradient descent process. Finally, training the Q-function on the training dataset by conducting updating until the Q-function is converged.

4.3 Tool prediction results

After model training stage, a well-trained Q-function is obtained and applied in next testing stage. The network structure of the trained Q-function is first updated and transferred to establish a new DRL network for feature learning of tool2. Then the sensor data of tool2 is preprocessed and the wavelet packet energy is calculated by OWPT. To clearly display the prediction results, six features are drawn separately on two graphs. As shown in Fig. 5, the force and vibration energy features (f1 to f6) learned by the DTRL network are drawn independently. The X axis represents the cutting steps of tool2, and the Y axis represents the corresponding energy features. It can be seen that with the increase in time of tool running, the predicted value has a similar increasing trend with the real value. Increasing trend of each extracted feature is consistent with that of tool wear degree in tool2, which shows the extracted features can effectively reflect the degradation of tool performance. The result demonstrates that DTRL network is an effective feature prediction model for tool wear monitoring.

To further perform RUL prediction by the DTRL network for tool2, first is establishing nonlinear mapping function between predicted energy features and time series of RUL. Support vector regression (SVR) is used to construct the regression model, in which radial basis function (RBF) is

Table 1 MAE and RMSE achieved by DTRL for tool2

MAE	RMSE
16.54	20.23

Table 2 MAE and RMSE for tool2 without deep Q transfer learning

MAE	RMSE
20.61	26.29

chosen as the kernel function. The kernel function can be expressed as:

$$K(x, x_i) = \exp\left(-\gamma\|x-x_i\|^2\right) \quad (18)$$

where γ denotes the width parameter. The energy features and corresponding RUL time of tool1 form the training samples. In order to minimize the training error, the width parameter γ is adjusted to 30.

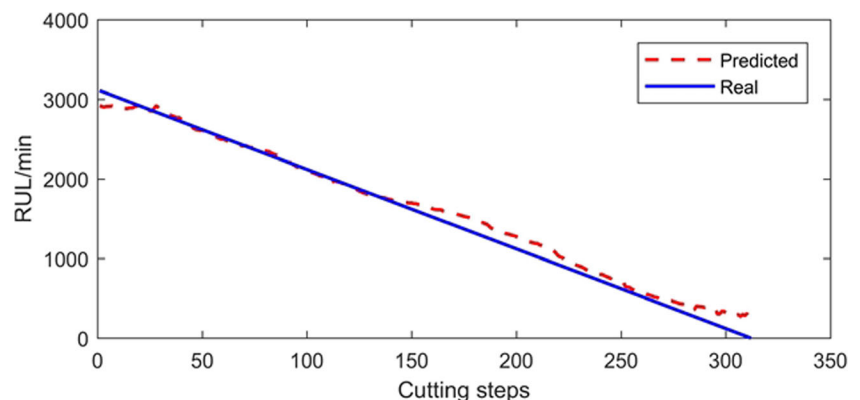
Finally, the energy features of tool2 are input to the trained regression model to further evaluate the effectiveness of the proposed DTRL method. RUL prediction result is shown in Fig. 6, and it can be seen that the predicted curve is very close to the real curve. To quantitatively measure the performance of DTRL method on RUL prediction tasks, two indicators for evaluating prediction precision are utilized including mean absolute error (MAE) and root mean squared error (RMSE). The corresponding equations to calculate errors are expressed as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |\tilde{T}_i - T_i| \quad (19)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (\tilde{T}_i - T_i)^2} \quad (20)$$

where \tilde{T}_i and T_i are true and predicted tool RUL, respectively.

The prediction error is shown in Table 1. It can be seen that the validation MAE and RMSE translate to RUL of 16.54min and 20.23min after denormalization. The results show a low prediction error, which demonstrate the effectiveness of DTRL method in RUL prediction.

Fig. 7 RUL Prediction result for tool2 without deep Q transfer learning

5 Model comparison and validation

In Section 4, the proposed DTRL method is used to predict RUL of a new tool. The tool types and cutting conditions are rather limited. To verify the effectiveness of deep Q transfer learning (DQTL) strategy in improving prediction performance, a comparison research without transfer learning for tool2 is performed. Furthermore, to investigate the generalization ability of the proposed DTRL method, more experiments are conducted.

5.1 Model comparison

The energy features of tool2 are input to the trained regression model without DQTL. RUL prediction result is shown in Fig. 7, which shows a larger prediction error compared with Fig. 6, especially from middle stage of cutting processing. The corresponding prediction error is shown in Table 2. It can be seen that the quantitative error is higher compared with that in Table 1, which demonstrates the proposed DQTL is an effective method to improve prediction accuracy of DRL network.

5.2 Model validation

To verify generality of the method, two kinds of experiments are extended. One kind is utilizing a new type of tool under same cutting conditions, in which the diameter of original tool is 6mm and the diameter of new type is 8mm. The other kind is under different cutting conditions, in which the spindle rotation frequency is 10000 r/min and the cutting depth in radial and axial direction is 0.25mm.

Comparing parameters include tool geometries and cutting parameters. During the experiments, tool geometries are fixed, thus cutting parameters have greater influence on force and vibration amplitude. Cutting parameters, including spindle speed, and cutting depth, may cause different dynamic ranges under similar cutting conditions. To eliminate the above effects and improve the generalization ability of the proposed model, a zero-centered step is first applied to the collected

Table 3 MAE and RMSE of extend experiments

	MAE	RMSE
Tool with different diameters	26.12	33.68
Tool under different cutting conditions	23.82	30.43

signals. Then, wavelet packet energy features are extracted from the cutting force and vibration signals of each sampling point. In this paper, 6-level wavelet packet decomposition is adopted, thus the harmonics are decomposed into 64 wavelet packet energy coefficients. Finally, energy features are input into the model instead of original signals (see Fig. 4), so the model is robust to various cutting conditions.

Table 3 shows the MAE and RMSE translated to RUL in two extended experiments. The results indicate that the DTRL method can effectively predict tool wear under various cutting conditions.

6 Conclusion

In this paper, a DTRL method is introduced to conduct deep Q-function transfer in reinforcement learning network for tool wear and RUL prediction. Two strategies, including introducing LSTM into DQN and Q-function update and transfer, are designed to realize DTRL network. Experimental results show that by introducing the deep Q transfer learning strategy, it contributes to more accurate and reliable tool wear prediction results. Furthermore, a DRL network trained by similar tools or conditions can be transferred to the target tool when researching on new types of tools or processing high value parts. As the reinforcement learning has been widely used in control system, further works will extend the model to cutting edge control and cutting path planning.

Author contribution Jiachen Yao performed the data analyses and wrote the manuscript; Baochun Lu contributed to the conception of the study; Junli Zhang performed the experiment and helped perform the analysis.

Funding This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1308300.

Data availability The data sets supporting the results of this article are included within the article.

Declarations

Ethics approval NA.

Consent to participate NA.

Consent to publish Written informed consent for publication was obtained from all participants.

Competing interests The authors declare no competing interests.

References

- Liao X, Zhou G, Zhang Z, Lu J, Ma J (2019) Tool wear state recognition based on GWO-SVM with feature selection of genetic algorithm. *Int J Adv Manuf Technol* 104(1-4):1051–1063
- Aghazadeh F, Tahan A, Thomas M (2018) Tool condition monitoring using spectral subtraction and convolutional neural networks in milling process. *Int J Adv Manuf Technol* 98(9-12):3217–3227
- Wang G, Qian L, Guo Z (2012) Continuous tool wear prediction based on Gaussian mixture regression model. *Int J Adv Manuf Technol* 66(9-12):1921–1929
- Si X, Wang W, Hu C (2013) A Wiener-process-based degradation model with a recursive filter algorithm for remaining useful life estimation. *Mech Syst Signal Process* 35(1-2):219–237
- Yan H, Zhou J, Pang C (2015) Gamma process with recursive MLE for wear PDF prediction in precognitive maintenance under aperiodic monitoring. *Mechatronics* 31:68–77
- Wang J, Wang P, Gao R (2015) Enhanced particle filter for tool wear prediction. *J Manuf Syst* 36:35–45
- Li C, Sanchez R, Zurita G (2016) Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mech Syst Signal Process* 76-77:283–293
- Javed K, Gouriveau R, Li X, Zerhouni N (2016) Tool wear monitoring and prognostics challenges: a comparison of connectionist methods toward an adaptive ensemble model. *J Intell Manuf* 29(8):1873–1890
- Dou J, Xu C, Jiao S, Li B, Zhang J, Xu X (2019) An unsupervised online monitoring method for tool wear using a sparse auto-encoder. *Int J Adv Manuf Technol* 106:2493–2507
- Sun C, Wang P, Yan R (2019) Machine health monitoring based on locally linear embedding with kernel sparse representation for neighborhood optimization. *Mech Syst Signal Process* 114:25–34
- Widodo A, Yang B (2007) Support vector machine in machine condition monitoring and fault diagnosis. *Mech Syst Signal Process* 21(6):2560–2574
- Chen B, Chen X, Li B (2011) Reliability estimation for cutting tools based on logistic regression model using vibration signals. *Mech Syst Signal Process* 25(7):2526–2537
- Karandikar J (2019) Machine learning classification for tool life modeling using production shop-floor tool wear data. *Procedia Manuf* 34:446–454
- Kong D, Chen Y, Li N, Tan S (2016) Tool wear monitoring based on kernel principal component analysis and v-support vector regression. *Int J Adv Manuf Technol* 89(1-4):175–190
- Zhang C, Zhang H (2016) Modelling and prediction of tool wear using LS-SVM in milling operation. *Int J Comput Integr Manuf* 29(1):76–91
- Kong D, Chen Y, Li N (2018) Gaussian process regression for tool wear prediction. *Mech Syst Signal Process* 104:556–574
- Kong D, Chen Y, Li N (2017) Force-based tool wear estimation for milling process using Gaussian mixture hidden Markov models. *Int J Adv Manuf Technol* 92(5-8):2853–2865
- Zhou Y, Wang T (2018) ENN-based recognition method for tool cutting state. *J Comput Sci* 27:418–427
- Serin G, Sener B, Ozbayoglu A, Unver H (2020) Review of tool condition monitoring in machining and opportunities for deep learning. *Int J Adv Manuf Technol* 109:953–974

20. Tao Z, An Q, Liu G, Chen M (2019) A novel method for tool condition monitoring based on long short-term memory and hidden Markov model hybrid framework in high-speed milling Ti-6Al-4V. *Int J Adv Manuf Technol* 105(7-8):3165–3182
21. Wang J, Ma Y, Zhang L, Gao R, Wu D (2018) Deep learning for smart manufacturing: methods and applications. *J Manuf Syst* 48: 144–156
22. Xu X, Wang J, Ming W (2020) In-process tap tool wear monitoring and prediction using a novel model based on deep learning. *Int J Adv Manuf Technol* 112(1-2):453–466
23. Jia F, Lei Y, Lin J, Zhou X, Lu N (2016) Deep neural networks: a promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech Syst Signal Process* 72-73:303–315
24. Shao H, Jiang H, Zhang H, Duan W, Liang T, Wu S (2018) Rolling bearing fault feature learning using improved convolutional deep belief network with compressed sensing. *Mech Syst Signal Process* 100:743–765
25. Wu X, Li J, Jin Y, Zheng S (2020) Modeling and analysis of tool wear prediction based on SVD and BiLSTM. *Int J Adv Manuf Technol* 106(9-10):4391–4399
26. Zhao M, Kang M, Tang B, Pecht M (2018) Deep residual networks with dynamically weighted wavelet coefficients for fault diagnosis of planetary gearboxes. *IEEE Trans Ind Electron* 65:4290–4300
27. Arulkumaran K, Deisenroth M, Brundage M, Bharath A (2017) deep reinforcement learning: a brief survey. *IEEE Signal Process Mag* 34(6):26–38
28. Silver D, Huang A, Maddison C (2016) Mastering the game of Go with deep neural networks and tree search. *Nature* 529:484–489
29. Wang Y, Fang Y, Lou P, Yan J, Liu N (2020) Deep reinforcement learning based path planning for mobile robot in unknown environment. *J Phys Conf Ser*:1576
30. Liu Q, Liu Z, Xu W, Tang Q (2019) Human-robot collaboration in disassembly for sustainable manufacturing. *Int J Prod Res* 57: 4027–4044
31. Yang H, Zhong W, Chen C (2020) Deep reinforcement learning based energy efficient resource management for social and cognitive internet of things. *IEEE Internet Things J* 7:5677–5689
32. Min M, Xiao L, Chen Y (2019) Learning based computation offloading for IoT devices with energy harvesting. *IEEE Trans Veh Technol* 68:1930–1941
33. Luo S (2020) Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Appl Soft Comput* 91:106208
34. Hu L, Liu Z, Hu W (2020) Petri-net-based dynamic scheduling of flexible manufacturing system via deep reinforcement learning with graph convolutional network. *J Manuf Syst* 55:1–14
35. Ding Y, Ma L, Ma J (2019) Intelligent fault diagnosis for rotating machinery using deep Q-network based health state classification: A deep reinforcement learning approach. *Adv Eng Inform*:42
36. Mnih V, Kavukcuoglu K, Silver D (2015) Human-level control through deep reinforcement learning. *Nature* 518:529–533
37. He Q, Kong F, Yan R (2007) Subspace-based gearbox condition monitoring by kernel principal component analysis. *Mech Syst Signal Process* 21(4):1755–1772
38. Zhang Z, Li H, Meng G, Tu X, Cheng C (2016) Chatter detection in milling process based on the energy entropy of VMD and WPD. *Int J Mach Tools Manuf* 108:106–112
39. Hong Y-S, Yoon H-S, Moon J-S, Cho Y-M, Ahn S-H (2016) Tool-wear monitoring during micro-end milling using wavelet packet transform and Fisher’s linear discriminant. *Int J Precis Eng Manuf* 17(7):845–855

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