The International Journal of Advanced Manufacturing Technology (2022) 121:6169–6192
https://doi.org/10.1007/s00170-022-09700-4

ORIGINAL ARTICLE

Deep Reinforcement Learning for autonomous pre-failure tool life improvement

Hussein A. Taha[1,2] · Soumaya Yacout[1] · Yasser Shaban[3]

Received: 11 March 2022 / Accepted: 6 July 2022 / Published online: 29 July 2022

Abstract

This paper develops an approach to improve a CNC machine's tool performance and slow down its degradation rate automatically in the Pre-Failure stage. A Deep Reinforcement Learning (DRL) agent is developed to optimize the machining process performance online during the Pre-Failure interval of the tool's life. The Pre-Failure agent that is presented in the proposed approach tunes the feed rate according to the optimal policy that is learned in order to slow down the tool's degradation rate, while maintaining an acceptable Material Removal Rate (MRR) level. The machine learning techniques and pattern recognitions are implemented to monitor and detect the tool's potential failure level. The proposed mechanism is applied to a CNC machine when turning Titanium Metal Matrix Composites (TiMMC). A CNC machine Digital Twin (DT) is developed to emulate the physical machine in the digital environment. It is validated with the physical machine's measurements. The proposed pre-failure mechanism is a model-free approach, which can be implemented in any machining process with fewer online computational efforts. It also validated on a wide range of cutting speeds, up to 15,000 RPM. Deployment of the proposed machine learning approach for the particular case study improves the tool's Time to Failure (T2F) by 40% and the MRR by 6%, on average, compared to the classical approach.

Keywords Degradation rate · Potential Failure · P-F curve · Reinforcement Learning · Tool performance

Nomenclature

α	Binary index
β	Scaling factor
$\Delta(O)$	Multi-class's discriminator function
γ	Discount factor
$\nabla_{Q^{\mu}}J$	Training losses of policy network
π^{*}	RL optimal policy
τ	Learning rate
θ^{μ}	Policy Network parameters
θ^{Q}	Q-Network parameters
a_t	Action at time t
A	Action space
F_x	Radial force N
F_y	Feed force N
F_z	Cutting force N
f	Feed rate rev/min
$L(\theta^{Q})$	Training losses of Q-network
N_h	Hidden neurons
N_s	Number of data observations
N_t	Random noise
$NTPI$	Normalized Tool performance degradation Index
$Q^{*}(s_t, a_t)$	Optimal Q-value
r_t	Reward at time t
s_t	State at time t
S	State space
VB_p	Potential failure tool wear
v	Cutting speed m/min
W_j	Pattern coverage wight

✉ Hussein A. Taha
hussein.taha@polymtl.ca

Soumaya Yacout
soumaya.yacout@polymtl.ca

Yasser Shaban
yasser.shaban@polymtl.ca

[1] Department of Mathematics and Industrial Engineering, Polytechnique de Montréal, Montreal, Canada

[2] Department of Electrical Engineering, Sohag University, Sohag, Egypt

[3] Department of Mechanical Design Engineering, Helwan University, Cairo, Egypt

1 Introduction

The integration of Artificial Intelligence (AI) in a Cyber-Physical System (CPS) is used to establish autonomous and self-driven machining processes [1–3]. The machining processes are usually associated with aspects of non-linear behavior and stochastic degradation that result in the difficulty of predicting the life span of the tool, especially when dealing with difficult-to-cut materials [4, 5]. There are many attempts in the literature to monitor the tool wear and detect the machining's tool failure. To achieve the highest possible material removal rates in machining, previous studies focused on offline optimization to schedule the machine feed rate, assuming ideal machining conditions [6]. Despite the previous attempts, an intelligent online system that can monitor and optimize the tool's performance in real-time is still needed. This paper fills this gap by offering a new approach that provides an intelligent-based extension of Tool Time to Failures (T2F) while maintaining an acceptable material removal rate level.

Offline mathematical optimizations were applied to the CNC machine processes to find the static machining parameters that maximize productivity [7]. Feed rate scheduling optimizations were developed to have a dynamic online feed rate setting [5, 8, 9]. One of the limitations of these approaches is the usage of empirical equations that assume ideal machining conditions. Adaptive control (AC) techniques take into consideration the environmental and sensor variations by mathematically estimating the forces at each time step, and then comparing the estimated values with the actual sensor measurements [5]. As such, the CNC machine controller's parameters are changed to achieve the offline optimized feed rate schedule. The estimation of online forces requires large computational time. Stemmler et al. [10] developed a Model Predictive Controller (MPC) to minimize the production time online for CNC milling machines. MPC is a model-based controller that predicts the values of the forces and adjusts the feed rate online accordingly to minimize the machining time. The MPC online optimization causes a processing delay, and thus, an additional signal processing synchronization is added to the machine controller. Both MPC and AC require mathematical modeling to estimate the forces. These models assumed a new tool at the beginning of cutting and ideal tooling conditions.

Shaban et al. studied tool wear monitoring for CNC machines to develop a failure alarm solution that could avoid producing defective pieces [11–13]. The authors applied Logical Analysis of Data (LAD) to detect the tool wear (VB) failure while monitoring the data of the machining forces. Sadek et al. [14] developed an adaptive mechanism that linked the tool wear monitoring to AC for a drilling machine. This mechanism is limited to two speeds and two feed rate adjustment levels. Shaban et al. used the time to Failure (T2F) and the Proportional Hazard Modeling (PHM) to obtain the optimal replacement time for the CNC machine tool [15]. The authors defined the tool replacement time at different machine settings using two types of analyses: tool availability and cost. The machining parameters were static settings, and it was adjusted before running the machine.

Taha et al. [16] developed a self-healing mechanism for a CNC milling machine. This mechanism dealt with the CNC machine under fault and approved self-healing mechanism to the machine. The authors used pattern-recognition machine learning to define the recovery patterns and each pattern is bounded by corrective settings. The self-healing mechanism selects the recovery pattern according to distance calculations to the current machine's faulty settings. From the selected pattern, the corrective actions are randomly selected through a uniform distribution that is bounded by the selected pattern.

Reinforcement Learning (RL) is a model-free approach that able to find the best value of a variable over its dimensional space, according to the optimal trained policy [17]. It successfully adapted to optimize the tool orientation [18], and manage the energy of a turning process under different cutting conditions [19]. The autonomous actions to improve tool life and productivity are needed to be addressed in the pre-failure zone. Table 1 provides a summary of many attempts in terms of what was achieved and what needs to be addressed. To fill the research gaps presented in Table 1, the main objective of this paper is to develop an autonomous pre-failure mechanism that interacts with the CNC machine in the P-F interval to extend the tool's useful life. Figure 1 depicts the P-F curve, which is a conceptual curve of degradation of any physical assets. The P-F curve has two main points that express its name: the potential failure point (P), and the function failure point (F) [20]. Practically, the degradation process is a stochastic phenomenon, and each tool has a unique P-F curve [3, 4].

This section presents the LAD algorithm to generate the patterns, which are used to develop the self-healing actions in module 3, and to classify and detect the out-of-specification in module 2.

The proposed Pre-Failure approach has the following features:

1. Model-free adjustment mechanism for the CNC machine.
2. Continuous feed rate adjustment.
3. Time to failure extension and degradation rate slow-down.
4. Lower online computation efforts.
5. Applicable for wide ranges of machining parameters.

In this work, a model-free Deep Reinforcement Learning (DRL) is proposed for continuous online feed rate adjustment. This

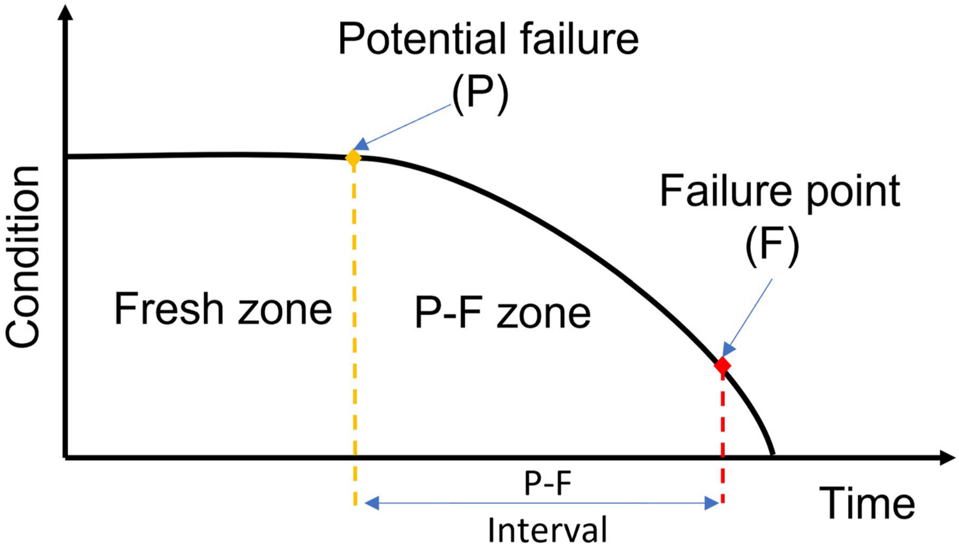
Table 1 Research Gaps

Achieved	Open Issues
Offline optimal machining parameters	-Assuming ideal machining and environmental conditions
	-Static machining parameters
-Feed rate scheduling optimization	-Offline feed rate optimization
-Adaptive control was applied to improve productivity	-Complex modeling and empirical equations
	-Ideal tooling conditions
-Online optimization	-Massive online computation
-Sensor predictions	-Complex modeling
	-Ideal tooling conditions
-Tool wear monitoring	-Passive system
-Failure alarm	-Correction mechanism was not addressed
-Adaptive correction mechanism	-Limited action values (feed rate)
-Adding Tool wear monitoring to adaptive control	-Tool Time to Failure (T2F) was not addressed
	-Limited spindle speeds (Two values)
	-Discrete value adjustment
-Optimal Replacement	-Static Machining settings (non-variable)
-Tool Time to Failure (T2F)	-Correction mechanism was not addressed
-Different machining parameters	-The failure detection module is time-independent
-Post-failure correction mechanism	The study is missing the following:
-Dynamic online machine settings	-T2F and degradation rate optimization
-Ease of implementation	-Optimal criteria to select the corrective settings
	-Effect of corrective action on productivity

approach is developed to add a tuning mechanism that optimizes the tool performance and productivity in the P-F zone of the machine's tool. The approach has the capability to achieve the highest possible material removal rate while maintaining an acceptable tool wear level. This approach can be implemented in any machining process with less computational effort.

This paper is organized as follows: Section 2 describes the system layout and Pre-Failure mechanism procedure. Section 3 contains the physical experiment and data review. Section 4 presents the proposed methodology. Section 5 describes the results of the implementation and provides a discussion. Lastly, Sect. 6 concludes the paper.

Fig. 1 P-F Conceptual Curve



2 System description

2.1 System layout

Figure 2 shows the system layout of the online autonomic closed loop for the CNC machine's pre-failure mechanism. In the pre-failure mechanism, there are two main phases and four modules. Phase 1 is the offline step for machine learning with Logical Analysis of Data (LAD), which is indicated by module 1. The software used is cbmLAD, which was developed for condition-based maintenance applications [21]. The cbmLAD is used to generate explanatory patterns that define the online P-F zones in module 2. The tool data is labeled according to the tool wear level VB as (a) new tool $VB < VB_p$, (b) Potential Failure (P-F) of the tool $VB_p < VB < VB_F$, and (c) Failure of the tool $VB < VB_F$. The data of the time-series forces are ingested into the cbmLAD to extract patterns that characterize the P-F zone. The tool's data is labeled as a failure when the tool wear level is more than, or equal to, a predefined value VB_F. The data in the potential failure zone is labeled in the same way, as shown in Sect. 4.1.

P-F zone monitoring in module 2 is based on online rules extracted from the cbmLAD's generated patterns. This module monitors the tool performance and detects the instant of Potential failure and the tool failure. Module 3 represents the CNC machine in the Digital environment. The developed CNC Digital Twin (DT) is proposed to work online and in parallel with the physical machine. The DT model is supported by an artificial Neural Network (NN) that reads the CNC machine settings of cutting speeds speed (v) and feed rate f. It estimates the machine's forces measurements $[F_x, F_y, F_z]$ at each time step (t+1), based on the sensor's readings of forces at the time (t).

Module 4 is a Deep Reinforcement Learning (DRL) Pre-Failure agent that generates action a_{t+1} to adjust the feed rate f_{t+1} according to the optimal policy that the agent learned in the training phase. In the online mode, module 2 reads the CNC machine forces sensor measurements at time (t) and enables the Pre-Failure agent in module 4 once the P-F zone is detected. The Pre-Failure agent reads the CNC machine measurement of the radial force F_x, the feed force F_y, the cutting F_z force, and the cutting speed v, at each time step. Accordingly, the proposed DRL agent adjusts the feed rate f_{t+1} to slow down the tool degradation rate.

2.2 Pre-failure intelligent mechanism procedure

This section presents the design steps to achieve the research objective of having a tool Pre-Failure mechanism for autonomous CNC machines. The proposed Pre-Failure mechanism has five main steps, as follows:

1. CNC machine experimental data: the material-tool pair Time to Failure (T2F) data is an essential process in order to improve the tool performance in the Pre-Failure stage. In this paper, the raw T2F data is analyzed to monitor and detect the tool performance degradation in the P-F zone. The developed CNC machine's Digital Twin (DT) model is validated and tested with the Physical raw data. These data were collected for the process of turning Titanium Metal Matrix composite (TiMMC) material. The raw data collection experiment was fully described in [13]. The data is presented in Sect. 3.
2. Tool P-F zone monitoring: the tool performance degradation is studied by building the tool P-F curve as the general one in Fig. 1. P-F curve shows the performance degrada-

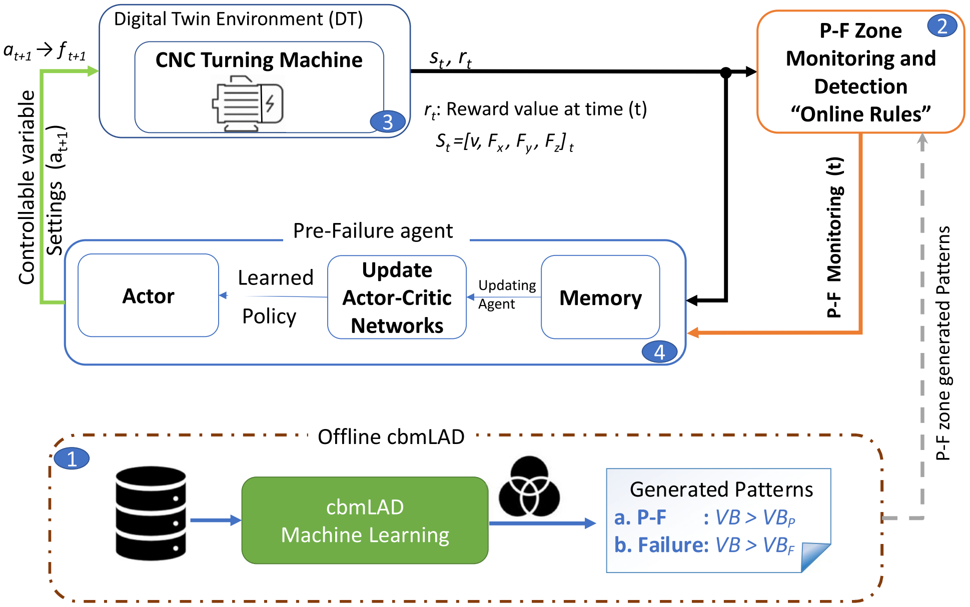

Digital Twin Environment (DT)
CNC Turning Machine
3
$a_{t+1} \rightarrow f_{t+1}$
s_t, r_t
r_t: Reward value at time (t)
S_t=[v, F_x, F_y, F_z]$_t$
P-F Zone Monitoring and Detection "Online Rules"
2
Controllable variable Settings (a_{t+1})
Pre-Failure agent
Actor
Learned Policy
Update Actor-Critic Networks
Updating Agent
Memory
4
P-F Monitoring (t)
P-F zone generated Patterns
Offline cbmLAD
1
cbmLAD Machine Learning
Generated Patterns
a. P-F : $VB > VB_P$
b. Failure: $VB > VB_F$

Fig. 2 Autonomic closed loop to achieve Pre-failure Mechanism

tion versus the lifetime of the tool. Section 4.1 presents the data analysis of tool performance degradation and the proposed algorithm to define the tool potential failure instant. It also includes a Logical Analysis of Data (LAD) and the online generated patterns that monitor the P-F zone. By the end of Sect. 4.1, module 3 in Fig. 2 is achieved.

3. Deep Reinforcement Learning (DRL) Pre-Failure agent: This is the step of designing the DRL agent (module 4 in Fig. 2). Section 4.2 explains the DRL for continuous feed rate adjustment, and it defines the pre-failure agent objective and state vector that describes the CNC machine status from the perspective of DRL. Section 4.2 includes a description of the agent's architecture, training algorithm, and communication links to modules 2 and 3. The added value and machining improvement for the tool's performance are discussed and verified in the results in Sect. 5.
4. CNC machine Digital Twin (DT): The DRL's environment is an essential part of Pre-Failure agent learning. A Digital Twin model is developed to interact with the DRL agent. The developed model lies on the collected experimental data, and it is validated with the physical machine tool's degradation. By the end of Sect. 4.3, the CNC machine's digital module 3 in Fig. 2 is accomplished.

3 Review of the experimental data

The Proposed Pre-Failure algorithm will be implemented on a CNC machine during turning Titanium Metal Matrix Composites (TiMMC), and all experimental data used in this study is based on [13]. In the data collection phase, the experimental data was recorded under different static machining parameters on a 5-axis Boehringer NG 200 CNC turning center [13]. The tool diameter was 1.6 mm, and the tool wear was measured with an Olympus SZ-X12 microscope. In [13], two design variables are included; feed rate *f* (mm/rev) and cutting speed *v* (m/min). In terms of the machining outputs and experiment response, the forces and flank wear *VB* (mm) are recorded. The experiment consisted of five runs, and at least five replications for each run.

For high-speed turning machines, ISO 3685:1993 stated that the maximum allowable *VB* before tool failure is 0.3 mm [22]. This criterion is uncommon in the aerospace industry to avoid the impairing surface damage caused by tool degradation [23]. In this paper, the experiments were conducted on a CNC machine turning TiMMC material for aerospace applications, and the tool wear level is recommended to be lower than 0.2 mm [13]. Therefore, *VB* was measured every 2 min until its value exceed the failure level of 0.2 mm.

Full factorial Design of Experiment (DoE) is the most conservative DoE type, as it aims to cover all combinations of machining parameters [24, 25]. In this paper, the experimental method was a full factorial 2-factors, 2-level full DoE. The levels of DoE were the maximum and minimum operation settings that were recommended by tool's supplier: speed $(v = 40\ ;\ 80\ m/min)$ and feed rate $(f = 0.15\ ;\ 0 : 35\ mm/rev)$. One more level $(v = 60\ m/min$, and $f = 0 : 25\ mm/rev)$ was added to the full factorial DoE in order to address the non-linearity of the tool degradation, and increase the prediction accuracy. The raw experimental data consists of 247 observations. A sample of the experimental data is given in Table 2.

For example, Fig. 3 shows the radial forces of the experimental data provided in Table 2 for cutting speeds of 80 m/min. Figure 3 contains five replications of run 2 and run 4. It should be stated from Fig. 3 that increasing the feed rate at the same cutting speed leads to higher radial forces. Accordingly, the T2F becomes shorter when increasing the feed rate.

4 Materials and methods

4.1 Tool degradation monitoring on PF curve

4.1.1 Tool potential failure point (P)

The P-F curve presents the tool performance's degradation against its operating time, and it defines the potential failure point (P) and the functional failure point (F) [20]. The tool has functionally failed when it exceeds the tool wear *VB* level that is recommended by the tool's manufacturer. The potential failure point (P) is the point at which the tool's failure propagation starts to increase, and it could be detected. In the P-F zone, the tool's performance has a significant deviation from its normal behavior when the tool is first installed. This performance is observed by the machine's sensor measurements of forces [13, 15]. P-F zone is an important mode as maintenance activities take place in this time interval [13, 15, 20].

In this paper, the knee point detection algorithm in [26] is adapted to define the potential failure point (P) for tool performance degradation according to the experimental tool wear data in Table 2. To plot the tool performance degradation P-F curve, an index that takes values in the interval of [0,1] is developed. The Normalized Tool performance degradation Index *NTPI* is given by $NTPI = 1 - 5 \times VB$. It equals one with the new tool and zero at the tool failure limit of $VB = 0.2$mm. Figure 4 shows an example of the P-F curve for a tool operated at 40 m/min cutting speed and feed rate of 0.35 mm/rev. The potential failure point (P) is detected at 560 sec, and the wear is 0.073 mm for this replication.

The knee detection algorithm [26] calculates the Euclidian displacement between all of the points on the *NTPI* graph and the perpendicular point on an imaginary reference line. This straight-line links the maximum and the minimum points of the tool performance, given by the dashed line in Fig. 5. The

Table 2 A sample of experimental Raw Data

No	Run-Rep	*t (s)*	*v (m/min)*	*f (mm/rev)*	F_x *(N)*	F_y *(N)*	F_z *(N)*	*VB (mm)*
1	1-1	120	40	0.15	120.4	51.1	116.2	0.0525
2	1-1	240	40	0.15	126	50	109.4	0.06
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
59*	1-5	1560	40	0.15	452.1	74.5	162	0.2
60	2-1	20	80	0.15	85	32.7	83.2	0.035
61	2-1	40	80	0.15	86	34	84	0.0425
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
101*	2-5	270	80	0.15	780.9	138.6	191.7	0.3
102	3-1	30	40	0.35	113.1	42.8	171.5	0.04
103	3-2	60	40	0.35	118	46.7	176.7	0.0475
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
188*	3-6	1320	40	0.35	618.8	161.1	252.3	0.22
189	4-1	15	80	0.35	136.2	41.5	159	0.05
190	4-1	30	80	0.35	153	47	164.2	0.065
:	:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:	:
247*	4-6	120	80	0.35	880	224.8	293	0.25

* The first recorded observation of the tool's failure in each run-replication

potential failure point is a point on the *NTPI* that has the maximum positive Euclidian distance. In Fig. 5, the red curve is the Euclidian displacement between *NTPI* and the reference line. Figure 6 depicts the Normalized Tool performance degradation Index *NTPI* and the potential failure (P) points for all the runs and replications of the experimental data given in Table 2. The potential failure points are indicated by dashed lines. As the tool degradation is a stochastic process, the detected potential points are not the same for all of the runs and replications. Table 3 summarizes the potential failure levels of the tool wear *VB* for each run and replication for the cutting speed $v1 = 40m/min\ and\ v2 = 80m/min$ and the feed rate $f1 = 0.15mm/rev\ and\ f2 = 0.35mm/rev$. The average potential (P) tool wear over the collected experimental data

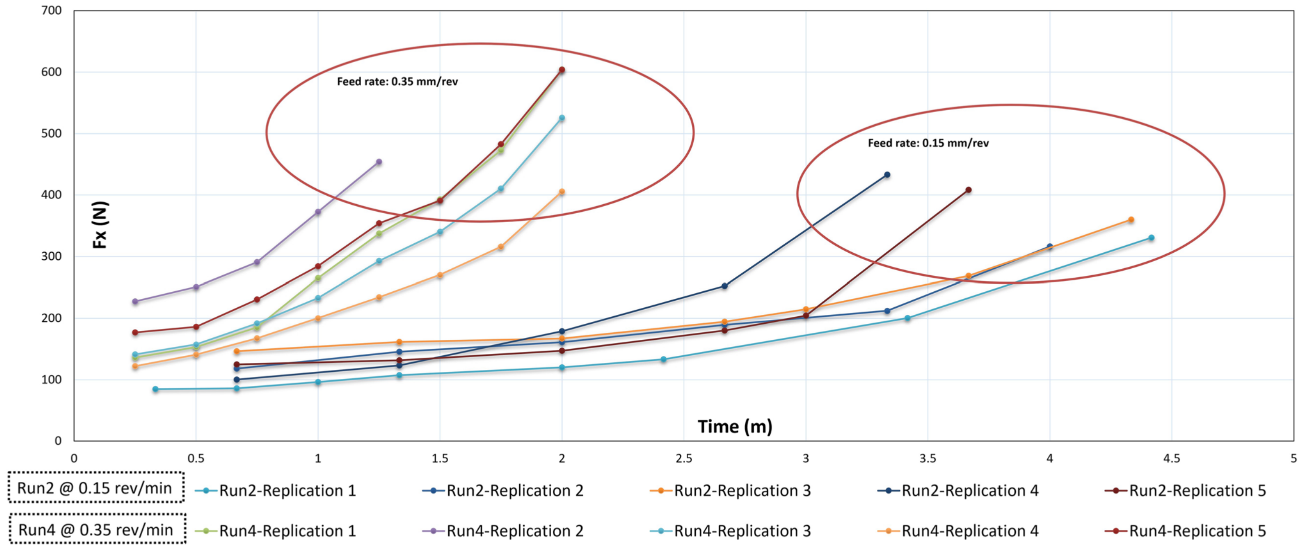


Fig. 3 Experimental radial forces F_x at cutting speed of 80 m/min and different feed rates for different replications

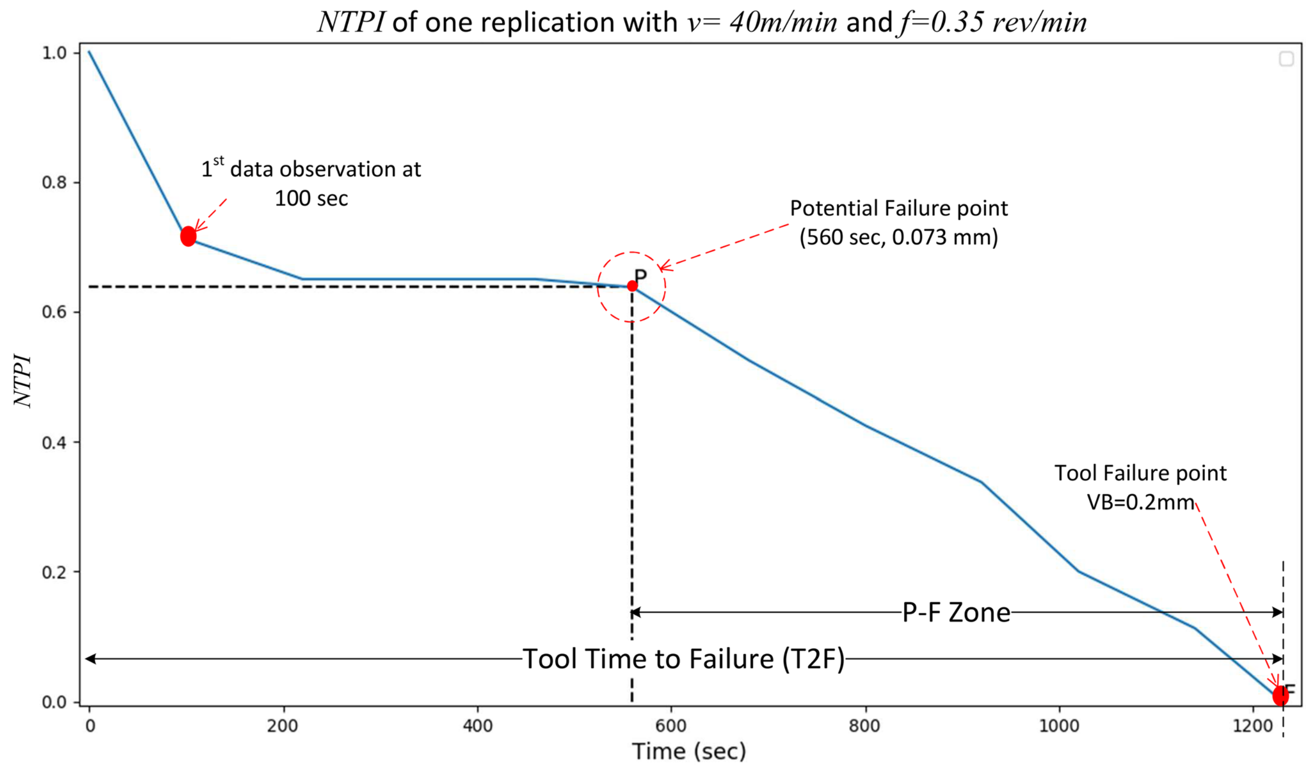


Fig. 4 P-F curve of one tool replication under 40 m/min cutting speed and 0.35 mm/rev feed rate

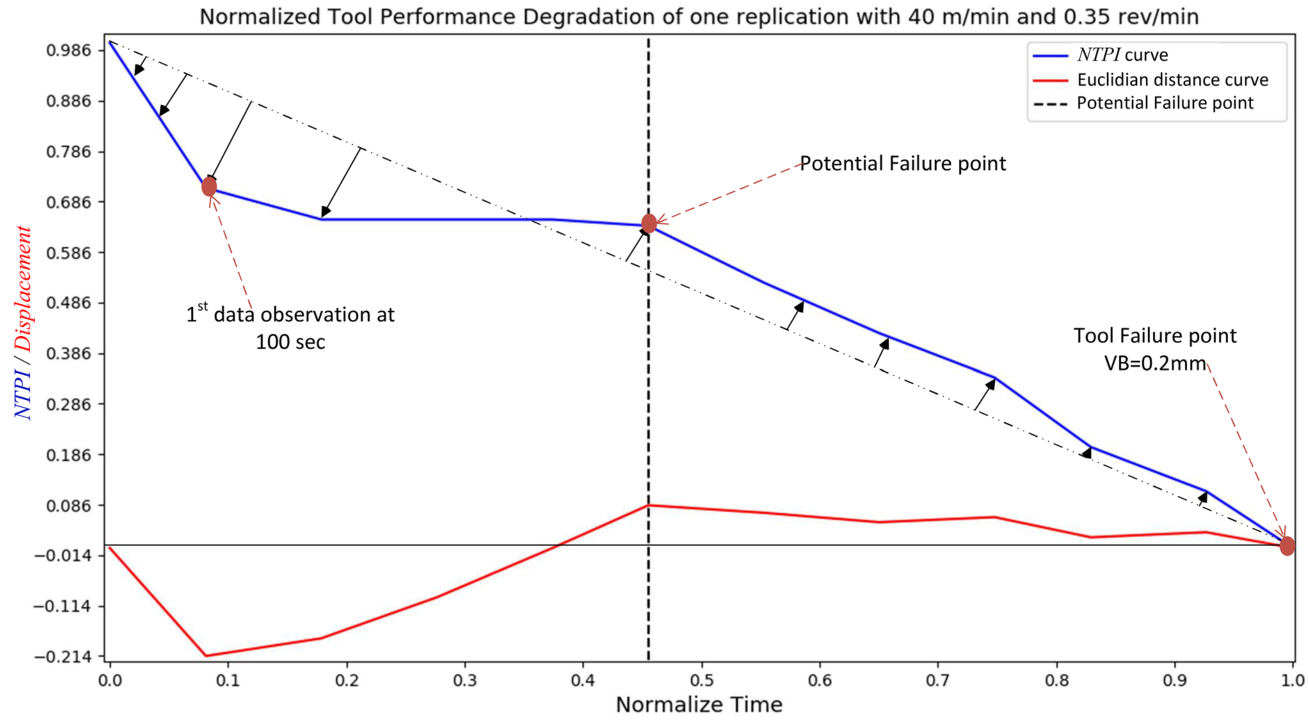


Fig. 5 P-F curve and Euclidian distance curve for P point detection

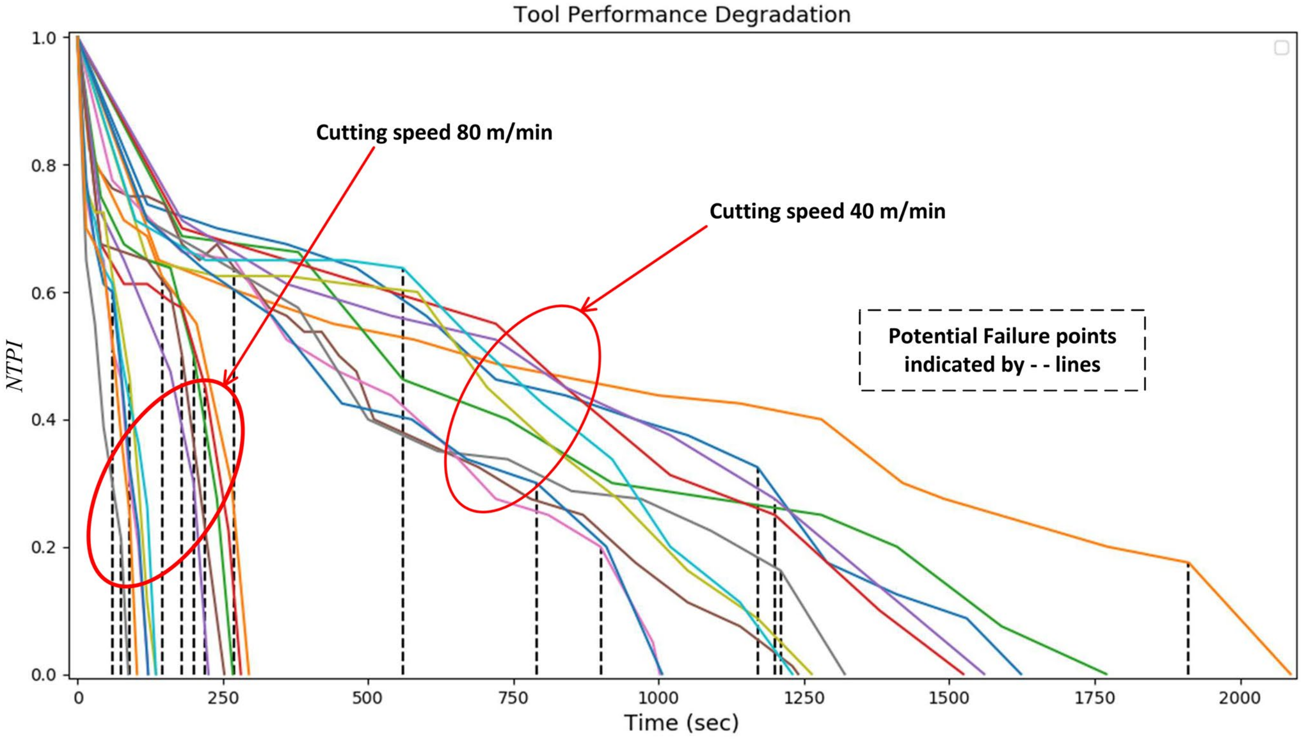


Fig. 6 TiMMC Normalized Tool performance degradation Index *NTPI* for different runs and replications of the experimental data in Table 2

is 0.135 mm. Tool P-F zone is the pre-failure zone at which the correction mechanism is needed.

4.1.2 Tool P-F zone online monitoring and detection

Logical Analysis of Data (LAD) is a non-statistical supervised data mining method. It uses Boolean logic functions and combinatorial optimization for classification [21, 27]. The advantage of LAD over other classification methods is to generate explanatory patterns for each class, which maintains comparative performance in knowledge extraction for supervised and semi-supervised classification problems. The patterns divide the multidimensional space of features into zones that characterize the classes.

cbmLAD solves Mixed-integer Programming (MILP) optimization problems iteratively to find the logical relationships among the input data features by generating patterns that characterize each class of the tool's life [28]. For each pattern, each feature is bounded by a specific range of values. For a new data point, the pattern is satisfied if the value of the measured features lies in its bounded range. In the one versus all (OVA) classification technique, the cbmLAD generates patterns to characterize a specific class from the other classes. From the tool P-F curve, the tool-life consists of three zones, as shown in Fig. 7: (a) New tool class, (b) Pre-failure class, and (c) Failure class. To detect the tool degradation state, cbmLAD divides the classification problem into three sub-problems and finds each class' discrimination function Δ1, Δ2, Δ3.

For a new observation O, the OVA cbmLAD multi-class's discrimination function $\Delta(O)$ is given in Eq. 1 as described in [28].

$$\Delta(O) = arg_i max[\Delta_i(O)]$$
$$\Delta_i(O) = \sum_{P_j} w_j \alpha_{P_j}(O),\ \forall i = 1, 2, 3 \tag{1}$$

Table 3 Potential failure points of the CNC tool experimental data

P-tool wear (mm)	Rep 1	Rep 2	Rep 3	Rep 4	Rep 5	Rep 6	Average
Run 1 **($v1$ = 40 m/min, $f1$ = 0.15 mm/rev)**	0.135	0.165	0.2	0.15	0.145	-	0.159
Run 2 **($v2$ = 80 m/min, $f1$ = 0.15 mm/rev)**	0.075	0.1	0.11	0.14	0.1	-	0.105
Run 3 **($v1$ = 40 m/min, $f2$ = 0.35 mm/rev)**	0.073	0.16	0.168	0.2	0.073	0.14	0.1357
Run 4 **($v2$ = 80 m/min, $f2$ = 0.35 mm/rev)**	0.2	0.158	0.108	0.077	0.2	0.095	0.1397
The average value of the potential tool failure							**0.135 mm**

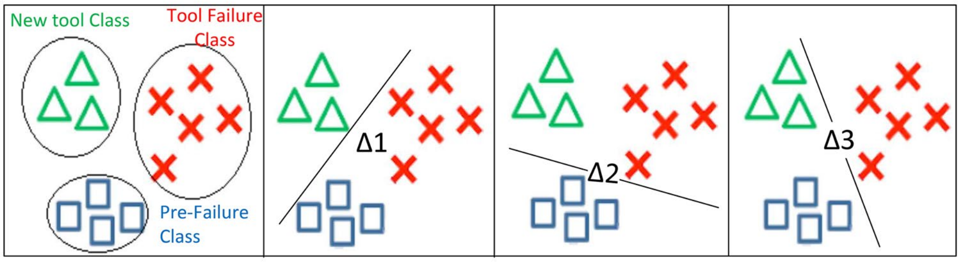


Fig. 7 OVA Technique for the Tool Degradation Performance Classes in Two-Dimensional Space

where P_j is the j^{th} pattern that covers the observation O where j is the number of the pattern that belongs to the class i set of patterns. W_j is the pattern coverage weight, and α is a binary index, which is 1 when the observation O is covered by pattern P_j, and zero otherwise.

In this paper, the cbmLAD one versus all (OVA) technique is applied to solve a multi-classification problem in order to find the tool's state of degradation. In online mode at each time step (t), the P-F monitoring and detection module 2 in Fig. 2 monitors the time-stamped machine's sensors $[t, F_x, F_y, F_z]$ and checks whether the measured observation is covered by any of the patterns that represent the pre-failure or failure zones. The Pre-failure zone's detection signals are used to activate or deactivate the Reinforcement Learning RL module 4 in Fig. 2.

From Table 3, the potential failure VB level is 0.135 mm on average. The P-F zone is defined when the tool wear is $0.135 \leq VB < 0.2$, and the failure level is $VB > 0.2$. The experimental data is categorized into three main classes: (a) $VB < 0.135$, (b) $0.135 \leq VB < 0.2$, and (c) $VB \geq 0.2$ mm. According to the developed tool P-F curve, the data in Table 2 is identified by the classes label and ingested to OVA cbmLAD to generate the tool life's patterns. Table 4 presents the generated patterns that characterize the P-F and the failure classes.

The online P-F monitoring module 2 in Fig. 2 performs two main functions, (a) the monitoring of the potential failure interval with ten patterns, and (b) the detection of failure with five patterns. Each pattern in Table 4 is represented in a multidimensional zone in the features space. At each time step (t), once the measured observation lies in a pattern zone, a signal is sent to activate or deactivate the Pre-Failure agent module 4 in Fig. 2. The P-F monitoring and detection module's scanning cycle is synchronized with the machine module 3 in Fig. 2 and with the pre-failure agent.

4.2 Deep Reinforcement Learning (RL) model

The standard RL is formalized as an agent that interacts with a system's environment, then receives the current system state and instant reward r_t at time t [17, 29]. The RL goal is to find the optimal policy π^* that maximizes the return from the state $R_t = \sum_{t=0}^{\infty} \gamma r_t(s_t, a_t)$, where $\gamma \in [0, 1]$ is the discount factor for future rewards and t is instant of the return

Table 4 Generated patterns of P-F and failure zones for the data of the time-stamped Force

Class	*Pattern*	*t (sec)*	*Fx (N)*	*Fy (N)*	*Fz (N)*
P *re-F ailure*	1	$102.5 < t$	$305.3 < F_x < 685.95$	$126.85 < F_x < 155.85$	$203.3 < F_z$
	2	-	$305.3 < F_x < 542.3$	$117.25 < F_x < 174.5$	$F_z < 250.7$
	3	$85 < t$	$341.35 < F_x < 507.85$	$F_y < 174.5$	-
	4	$685 < t$	$271.2 < F_x < 605.4$	$71.3 < F_x < 174.5$	$175.25 < F_z < 250.7$
	5	$t < 67.5$	$354.1 < F_x < 507.85$	$93.75 < F_y < 174.5$	-
	6	$1085 < t$	$271.2 < F_x < 507.85$	$F_y < 174.5$	-
	7	$805 < t$	$261.9 < F_x < 507.85$	$78.25 < F_y < 114.6$	-
	8	$127.5 < t$	$316.1 < F_x < 507.85$	$F_y < 174.5$	$F_z < 190.25$
	9	$t > 685$	$249.5 < F_x < 488.5$	$71.3 < F_y < 174.5$	$139.7 < F_z < 159.85$
	10	-	$271.2 < F_x$	-	$143.75 < F_z < 150.2$
Class	*Pattern*	*t (sec)*	*Fx (N)*	*Fy (N)*	*Fz (N)*
Failure	1	$t < 327.5$	$F_x > 605.4$	-	-
	2	$1025 < t$	$F_x > 542.3$	-	-
	3	$t < 112.5$	$F_x > 543.3$	-	-
	4	$127.5 < t < 327.5$	$F_x > 488.5$	-	$F_z > 174.85$

[29, 30]. The expected return value of taking action (a_t) in the state s_t under a policy π is called Q-function and it is equal to $Q^{\pi}(s_t, a_t) = E_{r_t, s_{t+1}}[r_t(s_t, a_t) + \gamma E_{a_{t+1}}[Q^{\pi}(s_{t+1}, a_{t+1})]]$. The optimal Q-value $Q^{*}(s_t, a_t) = max(Q^{\pi}(s_t, a_t))$ is the maximum returned value $\forall s_t \in S$ and $\forall a_t \in A$, where S is the state space, and the action space A is limited to discrete actions [29, 30]. The optimal policy is obtained from the optimal Q-value when obtaining the action that maximizes the returned Q-value; in mathematics, this is given by $\mu(s) = argmax_a Q(s_t, a_t)$ [29, 30]. The Q-learning is an off-policy algorithm that uses a greedy policy. It learns the optimal policy π^{*} as it approximates the Q-function by the Q-network parameters θ^{Q}. The optimal Q-function $Q^{*}(s_t, a_t)$ is achieved by obtaining the optimal parameters $\theta^{Q^{*}}$ at which the training loss $L(\theta^{Q}) = E_{s_t, a_t, r_t}[(Q(s_t, a_t|\theta^{Q}) - y_t)^2]$ is the minimum. y_t is the target function with next time step state s_{t+1}, and it is calculated as $y_t = r(s_t, a_t) + \gamma Q(s_{t+1}|\theta^{Q})$. For the Q-learning stability, the target y_t is calculated by another identical Q-network [2, 29]. Practically speaking, it is difficult to apply Q-learning to a continuous action space, and the Q-algorithm is not capable of optimizing an infinite number of actions at each time step. The actor-critic approach is used to solve this problem with the Deterministic Policy Gradient (DPG) algorithm [2, 29]. The critic is an action-value function $Q(s, a|\theta^{Q})$ used to calculate the temporal difference (TD) error to criticize actions made by the actor, and it is updated based on the Q-function. The actor is a deterministic policy function $\mu(s|\theta^{\mu}$ that chooses action a_t given state s_t [2, 29]. The actor's network parameters θ^{μ} are updated according to maximizing the action-value $Q(s, a|\theta^{Q})$, and its training losses are ascending losses given by $\nabla_{Q^{\mu}} J \approx E_{s_t}[\nabla_a . Q(s, a|\theta^{Q})|_{s=s_t, a=\mu(s_t)} \nabla_{Q^{\mu}} \mu(s|\theta^{\mu})|_{s=s_t}]$. A Deep Deterministic policy gradient (DDPG) is an algorithm that implements the deep Q-network on the DPG algorithm. DDPG approximates the Q-function and enables RL in systems that have continuous actions and a large-dimension state [2, 29]. DDPG is a model-free RL algorithm that uses a replay buffer memory to update the system's states, actions, and rewards during agent training [17, 29]. In this paper, the pre-failure agent is an adapted DDPG algorithm to achieve optimal proactive and autonomous feed rate adjustment.

4.2.1 Pre-failure agent for autonomous CNC machine

In the current work, the CNC turning machine pre-failure agent action a_t is performed on the feed rate f_t mm/rev. At each time step t = 1sec, the agent reads the machine sensor data $[F_x, F_y, F_z]$ and scans the P-F monitoring module signal. The agent generates actions to decrease the tool's degradation rate and keep a reasonable productivity limit in the P-F zone, which is given in Eq. 2.

In the training phase, the pre-failure agent interacts with the DT model of the CNC turning machine, and it is rewarded for each action $a_t \rightarrow f_t$ with the reward function r_t. The rewarded value depends on the tool degradation rate and productivity at each time step. The machine's productivity is represented by the Material Removal Rate MRR (mm^3/min) in Eq. 2.

$$MRR = f \times v \times d \tag{2}$$

where f is the feed rate in mm/rev, v is the cutting speed in m/min, and d is the cutting depth in mm. The proposed pre-failure agent interacts with the CNC machine at a different cutting speed, which varies from 25 m/min to 80 m/min. The agent's actions are a wide range of feed rate adjustments from 0.025 mm/rev to 0.35 mm/rev.

Definition of the state The classical RL algorithm is an extension of a Markov Decision Process (MDP) and its assumption of time-independent states [30]. At each time step (t), the RL agent receives state s_t and takes an action $a_t \rightarrow f_t$ according to its learned policy [29]. To learn the optimal policy, the state is assumed to be time-independent, and it describes the system status regardless of the system's historical behavior. In many industrial system applications, the MDP cannot fully describe a system in which the time-independent state is useless in learning the optimal policy [29]. For example, in CNC machining, the instant sensor measurement $[F_x, F_y, F_z]$t at time (t) cannot abstract the tool wear stage, as described in Sect. 4.1. Therefore, it is difficult to take a maintenance decision according to a time-independent instant value of forces. To ensure that the RL agent has the full features to describe the tool wear status, the RL state s_t is extended to include the cutting speed v (m/min), forces measurement at the instant of potential failure detection $[F_x, F_y, F_z]_p$, the sensor measurement deviation at time t from its value when the tool is at the P point $[E_{xyz}]$, and the negative rate of forces $[\Delta F_{xyz}]_t$ over sampling time T. Pre-Failure agent's state s_t is given by Eqs. 3, 4, and 5.

$$[E_{xyz}]_t = [F_x, F_y, F_z]_p - [F_x, F_y, F_z]_t \tag{3}$$

$$[\Delta F_{xyz}]_t = \frac{[F_x, F_y, F_z]_t - [F_x, F_y, F_z]_{t+1}}{T} \tag{4}$$

$$s_t = (v, [F_x, F_y, F_z]_p, [E_{xyz}]_t, [\Delta F_{xyz}]_t) \tag{5}$$

RL agent action The feed rate optimization is the key factor in optimizing the CNC machine tool's performance, as indicated in Sect. 3. At the same cutting speed v (m/min), the tool's degradation rate decreases while the feed rate f (mm/rev) is decreased, and this decreases productivity. The pre-failure RL agent is designed to generate optimal and continuous action at that adjusts the CNC machine feed rate at each time step (t). This action aims at decreasing the tool degradation rate while keeping the productivity within an acceptable limit. In practice, the adjustable feed rate range depends on the tool-material pair, and for products in composite materials, it could be changed from 0.025 mm/rev to 0.35 mm/rev.

Reward function In RL, the reward function $r_t(s_t, f_t, s_{t+1})$ acts as the objective function in mathematical programming. At each time t, the RL agent explores the action space A to find the optimal action at that maximizes its reward according to the given state s_t. The pre-failure agent reward function is designed to minimize the tool degradation rate and to keep the productivity level within acceptable limits. To maximize the tool Time to Failure (T2F), the Pre-Failure agent is designed with a positive reward function $r_t(s_t, f_t, s_{t+1})$ given by Eq. 6. To keep the productivity level relatively high, the agent is rewarded only if its action at minimizes the forces deviation $|E_{xyz}|_t$,which is the absolute difference between the measurement forces $[F_{xyz}]$ in the P-F zone and the detected potential failure forces $[F_{xyz}]_p$ at $t = t_p$ point. The tool degradation rate increases when the deviation decreases. Equation 6 indicates the instant reward $r_t(s_t, f_t, s_{t+1})$ calculation at each time step t.

$$r_t(s_t, f_t, s_{t+1}) = \begin{cases} 1 \text{ if } |E_{xyz}|_{t+1} \leq |E_{xyz}|_t \\ 0 \qquad\quad o.w \end{cases} \tag{6}$$

4.2.2 Pre-failure agent training

The Pre-Failure agent is an adapted DDPG algorithm, and its structure consists of the actor and the Q-function/Critic deep NNs. The actor adjusts the CNC machine feed rate according to the input RL state s_t and learned policy that is criticized by the Q-function network. To improve the DDPG agent training performance, a random noise $N_t \sim N(0, std)$ is added to the actor action [31]. The feed rate f_t to be adjusted at time (t) equals to $a_t = \mu(s_t|\theta^\mu) + N_t$, where $\mu(s_t|\theta^\mu)$ is the output of the actor-network and $\theta^\mu$is the parameters of the actor-network. In the training phase, the hidden layer's parameters of the Pre-Failure agent are updated to minimize the losses function of critics and to maximize the negative losses of the actor-network. To improve the learning stability, the target networks' parameters are updated with soft updates [31]; in other words, $\theta' \leftarrow \tau\theta + (1 - \tau)\theta'$, where the learning rate τ is less than one.

The target network's parameters are $\theta^{Q'}$ and $\theta^{\mu'}$ for the critic and the actor. The pre-failure agent's full training algorithm is given in Algorithm 1, and it is built and trained on a deep learning Pytorch environment.

Algorithm 1 CNC machines' Pre-Failure agent training

```
Initialize Critic $Q(s, a|\theta^Q)$, Actor $\mu(s|\theta^\mu)$
Initialize target networks $Q', \mu'$
for each episode: do
    Reads the cutting speed $v$.
    for each $K$ (min) steps in stimulation-Run: do
        scanning cycle time $t = k/T$
        switch Machine's tool P-F monitoring: do
            case New tool performance:
                Adjust the feed rate to maximum $f = f_{max}$
            case Tool in P-F zone:
                $[F_i]_P = [F_i]_{t_p} \ , \ \forall(i) \ \epsilon \ [x, y, z]$.
                Execute feed rate $f_t \leftarrow \mu(s_t|\theta^\mu) + \mathcal{N}_t$.
                Interact with CNC DT, get $[F_x, F_y, F_z]_{t+1}$.
                Calculate $s_{t+1} = (v, [F_{xyz}]_p, [E_{xyz}]_t, [\Delta F_{xyz}]_t)$
                $r(s_t, f_t, s_{t+1}) = \begin{cases} 0 & \text{if } |E_{xyz}|_{t+1} \leq |E_{xyz}|_t \\ 1 & o.w \end{cases}$
                Set targets $y_t = r_t + \gamma Q'(s_{t+1}, \mu'(s_{t+1}|\theta^{\mu'})|\theta^{Q'})$.
                Update Critic $L = \frac{1}{N}\sum_t (y_t - Q(s_t, a_t|\theta^Q))^2$.
                Update actor $\bigtriangledown_{Q^\mu} J \approx \frac{1}{N}\sum_t \bigtriangledown_a .Q(s, a|\theta^Q) \bigtriangledown_{Q^\mu} .\mu(s|\theta^\mu)$.
                Update $\theta^{Q'} \leftarrow \tau\theta^Q + (1 - \tau)\theta^{Q'}$.
                Update $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1 - \tau)\theta^{\mu'}$.
        if Tool worn then Break
        end if
    end for each
end for each
```

In DRL for continuous control applications, the selections of the agent's hyperparameters are correlated to the complexity of the environment, including numbers of inputs and outputs [29]. Agent complexity increases as hyperparameters are increased, which leads to high training accuracy, model overfitting, and low online accuracy. This issue is known as the bias-variance tradeoff [32]. In this paper, the environment is the CNC turning machine, and the pre-failure agent's action is the feed adjustment while the machine sensors are three F_x, F_y, F_z. The developed Pre-Failure agent architecture and hyperparameters are given in Appendix 1. These hyperparameters are adopted from the best performance DRL model in the literature of the same context. The adopted model's architecture performed a continuous control action on a computer game environment that has the same input-output dimensions as CNC turning machine [29].

Training of pre-failure DRL agent requires a large number of data observations and replications [33]. It is impractical to have massive machining experiments to train the DRL, as the TiMMC and its tool are costly [13]. Therefore, training of the DRL agent in a Digital Twin (DT) environment is proposed to address these limitations. DT is a potential candidate to eliminate or reduce the machining experiments' cost.

4.3 Digital Twin (DT) for CNC turning machine

Recently, the development of Industrial IOT (IIOT), simulation modeling, and Artificial Intelligence (AI) enable the digitalization of the machines, and the Digital Twin (DT) is extracted as a new concept of Cyber-Physical Systems (CPS) [3, 34]. Digital Twin is a model that emulates the Physical CNC machine in the cyber/digital environment, and it has the capability of interacting with the real machine in the Physical environment [34–36]. DT was developed for a system level, and in the case of a single machine, DT is developed on the component level. Figure 8 shows the implementation of DT on machine tool management. The digital environment contains physical data storage, data preprocessing, digital simulation models, and artificial intelligent agents. The digital environment has three main objectives: (1) monitoring the machine's data forces (2) analyzing this data to abstract the health status of the tool, and (3) taking action to improve the tool's performance. There are three methods to model a Digital Twin: Multiphysics modeling using Finite Element Analysis (FEA), Mathematical model-based, and/or Data-driven modeling [31]. In this paper, the experimental data in Table 2 is used to build the machine DT, and a deep artificial Neural Network (NN) is developed to act as a digital twin for the CNC turning machine. This model emulates the CNC turning machine in the digital environment. The DT's outputs are the estimated radial force F_x, feed force F_y, and cutting force F_z measurements at each time step t, and the model inputs are the cutting speed v (m/min) and the feed rate f (mm/rev), and time step t (min).

To minimize the overfitting of the model, a rule of sum for the NN architecture design is provided in Eq. 7 [37, 38]. The numbers of hidden layers are limited to the number of inputs N_i and number of outputs N_o, while the number of hidden neurons N_h for N_s data observations is given by Eq. 7 [37, 38]. β is a scaling factor that represents the prevention

Fig. 8 DT implementation on a machine's tool management

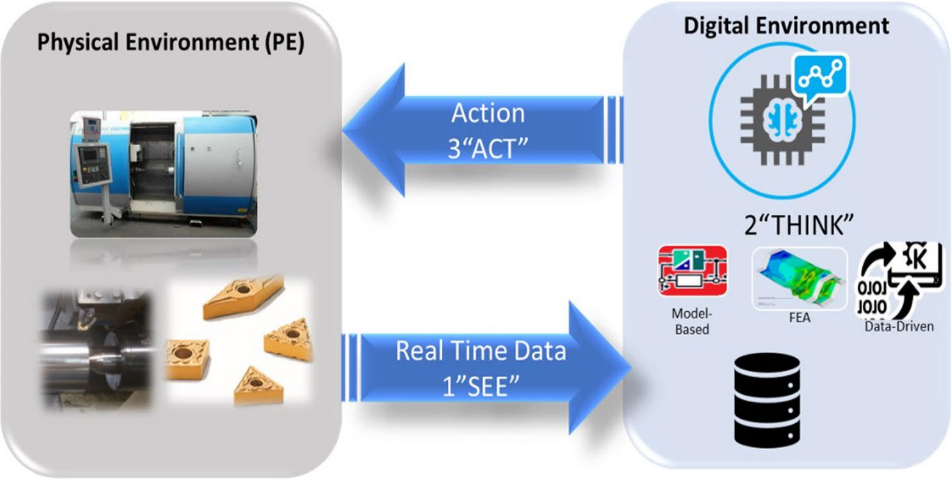


Table 5 Lowest MSE NN models and their hidden layers and neurons

Layers	**1**	**2**		**3**			**4**				**5**				
Layer #	1^{st}	1^{st}	2^{nd}	1^{st}	2^{nd}	3^{rd}	1^{st}	2^{nd}	3^{rd}	4^{th}	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}
Neurons	14	5	19	4	6	19	14	15	18	15	4	10	16	10	14
MSE	**6.63E+04**	**6.44E+04**		**6.26E+04**			**5.53E+04**				**5.88E+04**				

of overfitting in the NN model and it takes a value from 2 to 10 [37, 38].

$$N_h = \frac{N_s}{\beta(N_i + N_o)} \tag{7}$$

The developed digital model has three inputs [t, v, f] and three outputs F_x, F_y, and F_z. For 247 data observations, the number of hidden layers varies from 4 to 20 for $\beta \in [2, 10]$. One of the model's architectures is selected from more than 170 models' architectures. The models' hidden layers vary from single-layer to four-layer models, and each layer's amount of neurons changes from 4 to 20 neurons, and more five-layer models were added to the architecture's comparison. The best model is the model that has the lowest Mean Square Error (MSE) for the unseen testing data. Table 5 demonstrates the lowest MSE Network's architecture among all of the models with the same number of layers. The best model architecture of [14 – 15 – 18 – 15] is selected. To build the CNC machine DT, this model is trained and tested on a deep TensorFlow learning environment [39]. During the testing of the DT model, the testing Mean Absolute Error (MAE) was ±12.8 (e.g., Digital F_x = physical F_x ± 12.8). Figure 9 shows the DT model estimated forces versus the physical CNC turning machine sensors' reading for the unseen testing data of F_x, F_y, and F_z.

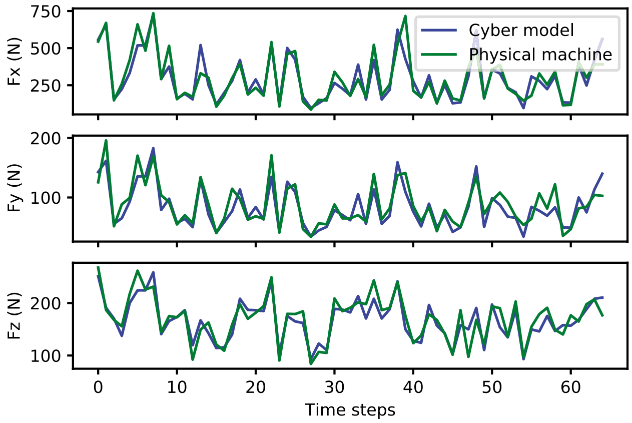


Fig. 9 Sensor data of **a** F_x, **b** F_y, and **c** F_z with the CNC cyber model vs. experimental physical testing data

5 Analysis of the results

This section analyzes the effects of the Pre-Failure agent on the tool performance and the tool Time to Failure (T2F) compared to the standalone CNC machine at different cutting speeds. The performance of the proposed Pre-Failure mechanism is measured by two key indexes: the Tool T2F and the achieved MRR. The P-F monitoring module activates the Pre-Failure agent in the P-F zone, and the agent is deactivated at the instant of tool failure. The closed-loop autonomy enables the Pre-Failure agent to adjust the optimal feed rate according to the estimated machine's forces at time (t + 1). In online mode, the Pre-failure agent interacts with the CNC machine every 1 s, and its sampling time T is selected as T = 60 sec.

DRL Pre-failure agent learned the optimal policy π^* in the training phase, and it figures out the mapping between the machine's sensor measurements that are represented in the state s_t, and the best action $a_t \rightarrow f_t$ at each time t. In online mode, the input state s_t is changed to follow the tool degradation, and accordingly, the developed pre-failure agent interacts with this degradation in the Pre-failure zone, and a new feed rate setting f_t is adjusted as the best action at time t based on the learned optimal policy π^*.

The trained Pre-Failure agent is validated with the CNC Turing machine DT at different cutting speed settings given in Table 6. In each run, the autonomic CNC machine is simulated until the tool failure is detected at VB = 0.2 mm. The

Table 6 Different speeds to validate the trained Pre-Failure agent

Simulation Run	**I**	**II**	**III**	**IV**	**V**
Spindle speed V**(RPM)**	5000	7500	10000	12500	15000
Cutting speed v **(m/min)**	25.12	37.68	50.24	62.8	75.36

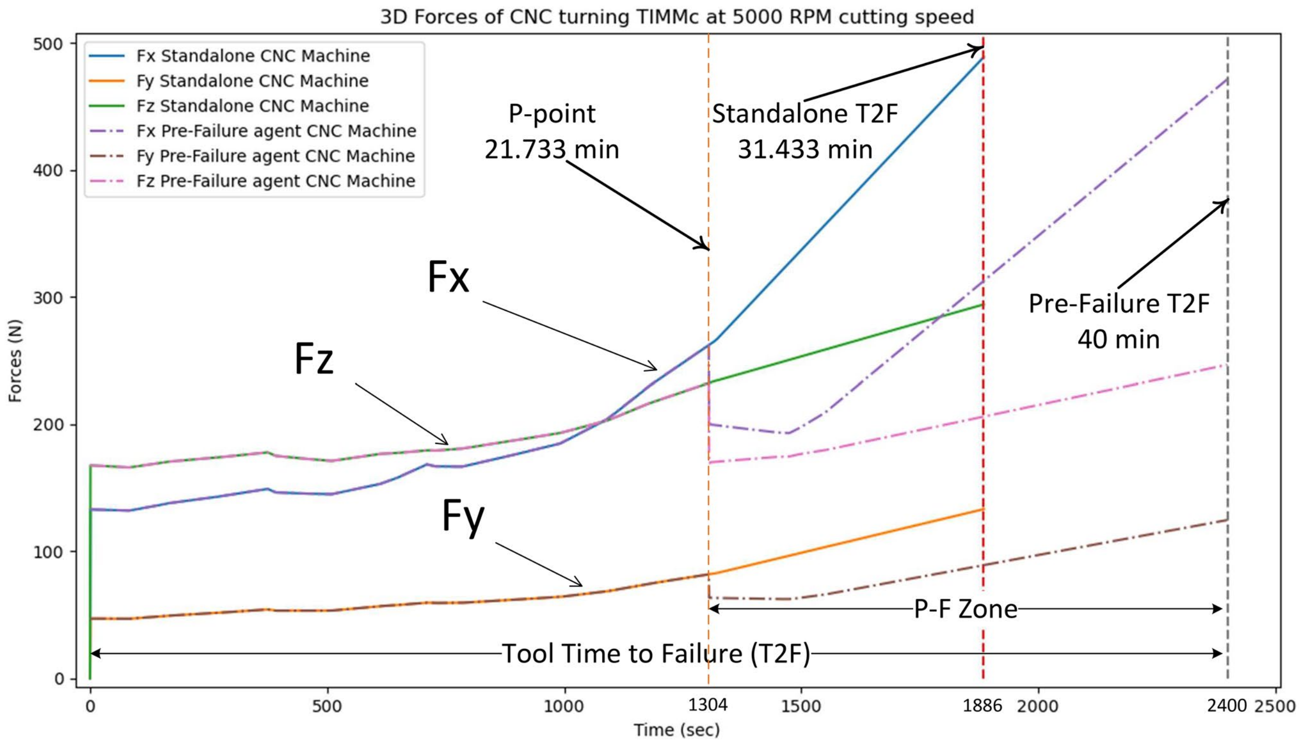


Fig. 10 3D Forces F_{xyz} (N) at 5000 RPM of Run I for the standalone machine and the Pre-Failure machine

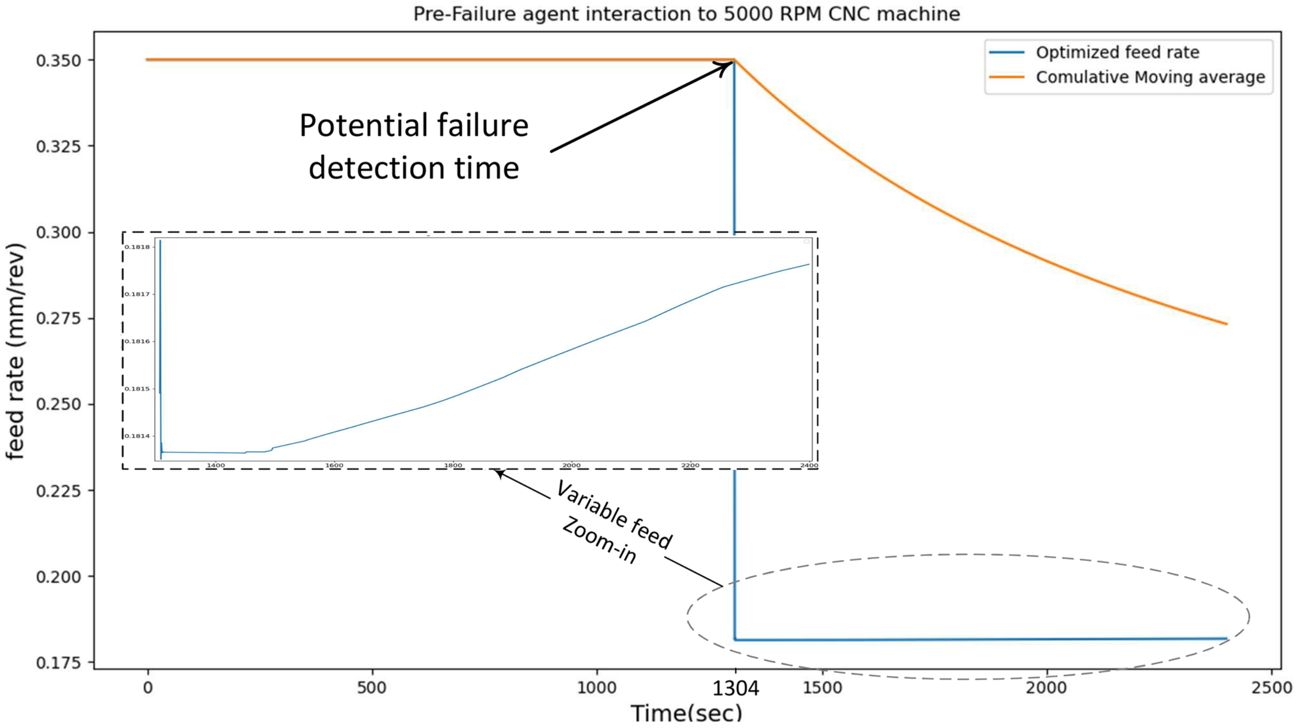


Fig. 11 Pre-Failure feed rate adjustment at a spindle speed of 5000 RPM in Run I

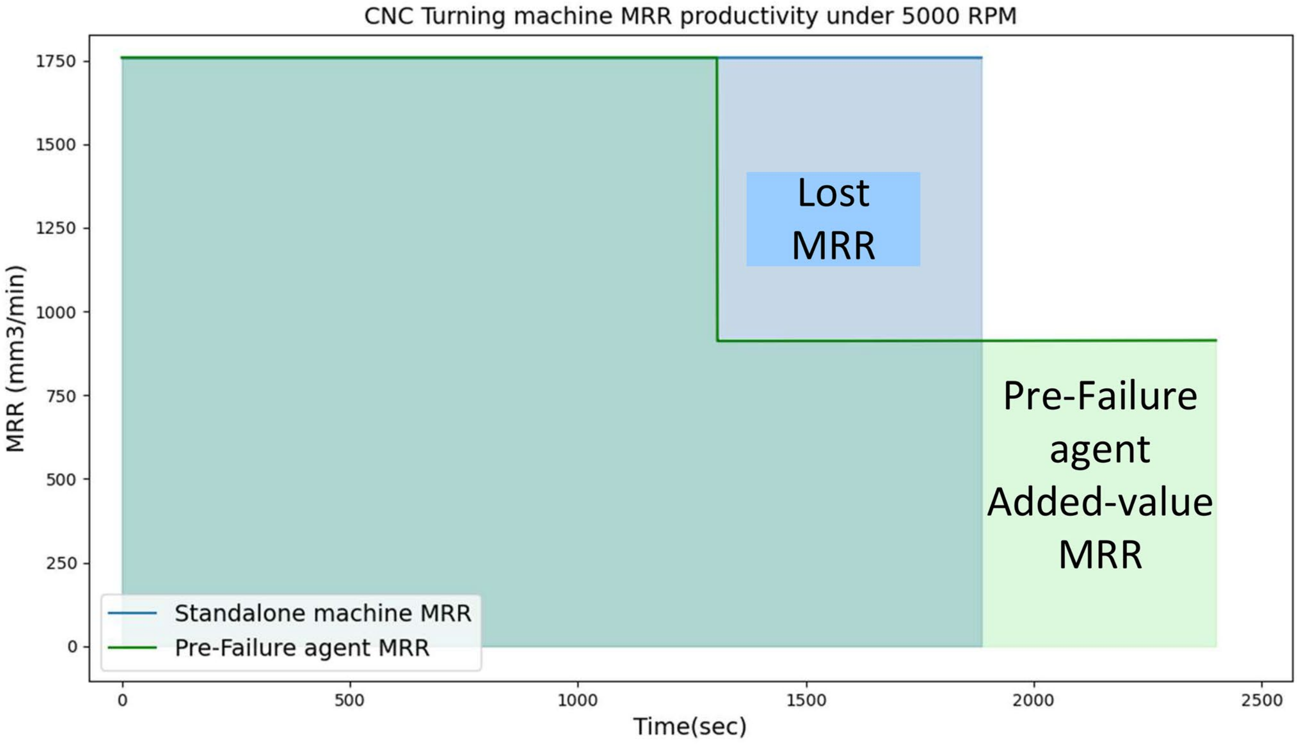


Fig. 12 Pre-Failure Agent Tool MRR VS. Standalone Machine

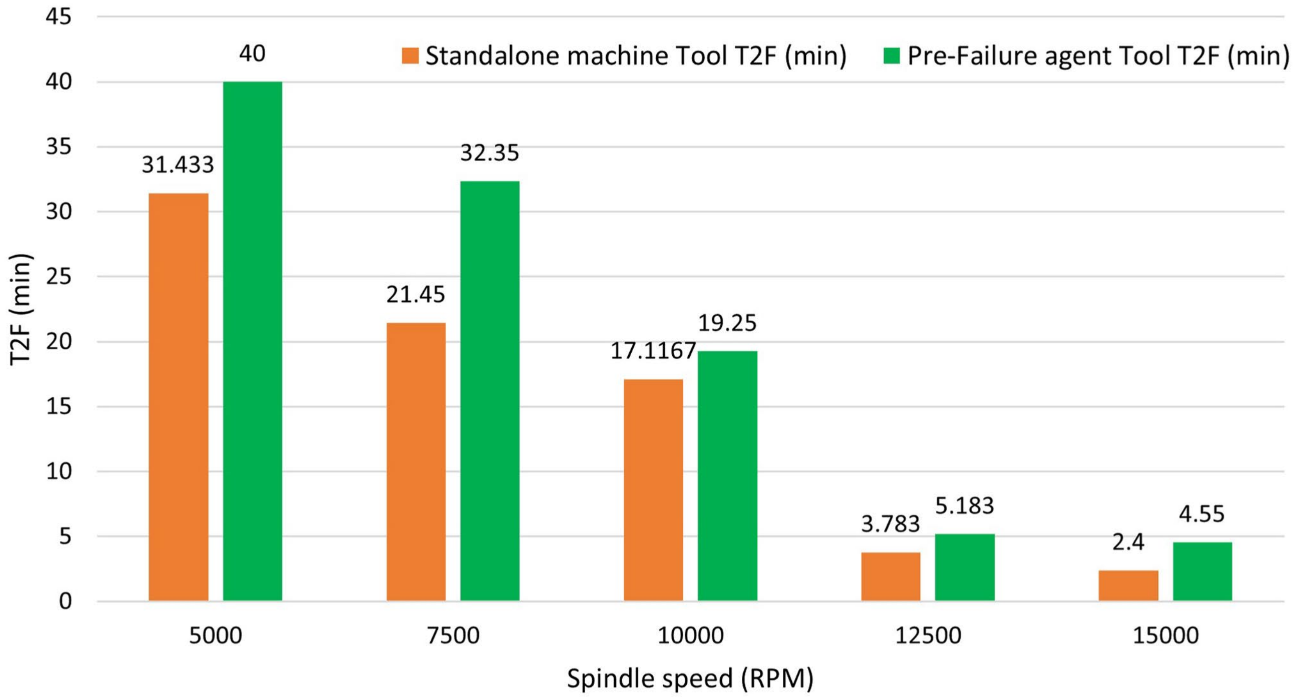
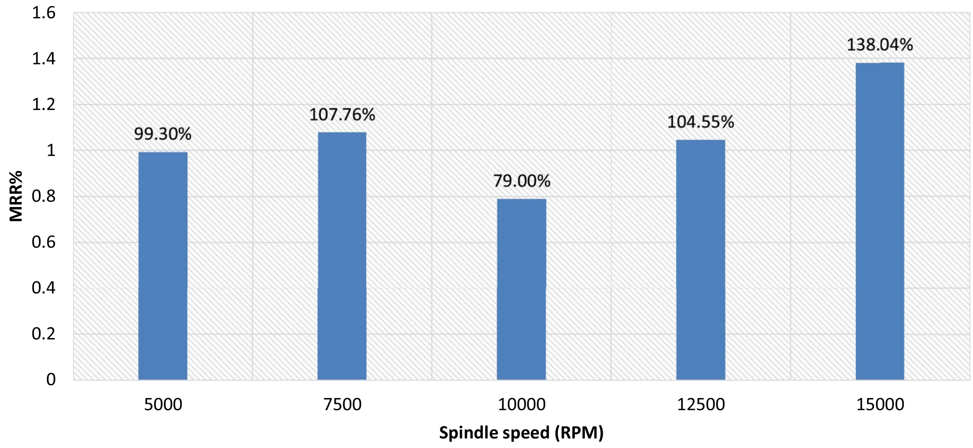


Fig. 13 Tool T2F of the Pre-Failure and the Standalone machine for different RPMs

Fig. 14 *MRR%* for different spindle speeds



machine's tool starts with a maximum feed rate of 0.35 mm/rev and the Pre-Failure enters the machining process when the P-F monitoring module 2 detects the potential failure class and the instant of P point.

In Run I, the Pre-Failure agent increased the tool T2F by almost 27% over the standalone CNC machine. Figure 10 illustrates the force measurements for the standalone CNC machine in solid lines, and the tool Time to Failure (T2F) is 31.433 min (1886 s). With the implementation of the Pre-Failure agent on the CNC machine, the degradation rate of the machine forces decreases, which is indicated by the dashed (− · −) lines in Fig. 10. The tool's T2F increases to 40 min (2400 s).

In the P-F zone, the Pre-Failure agent generates a continuous feed rate adjustment according to the optimal trained policy in Sect. 4.2. At a spindle speed of 5000 RPM, the adjusted feed rate at each time step (1 s) is given with the blue color in Fig. 11, while its accumulated moving average is given in the orange color. At t = 21.733 min (1304 s) the tool's potential

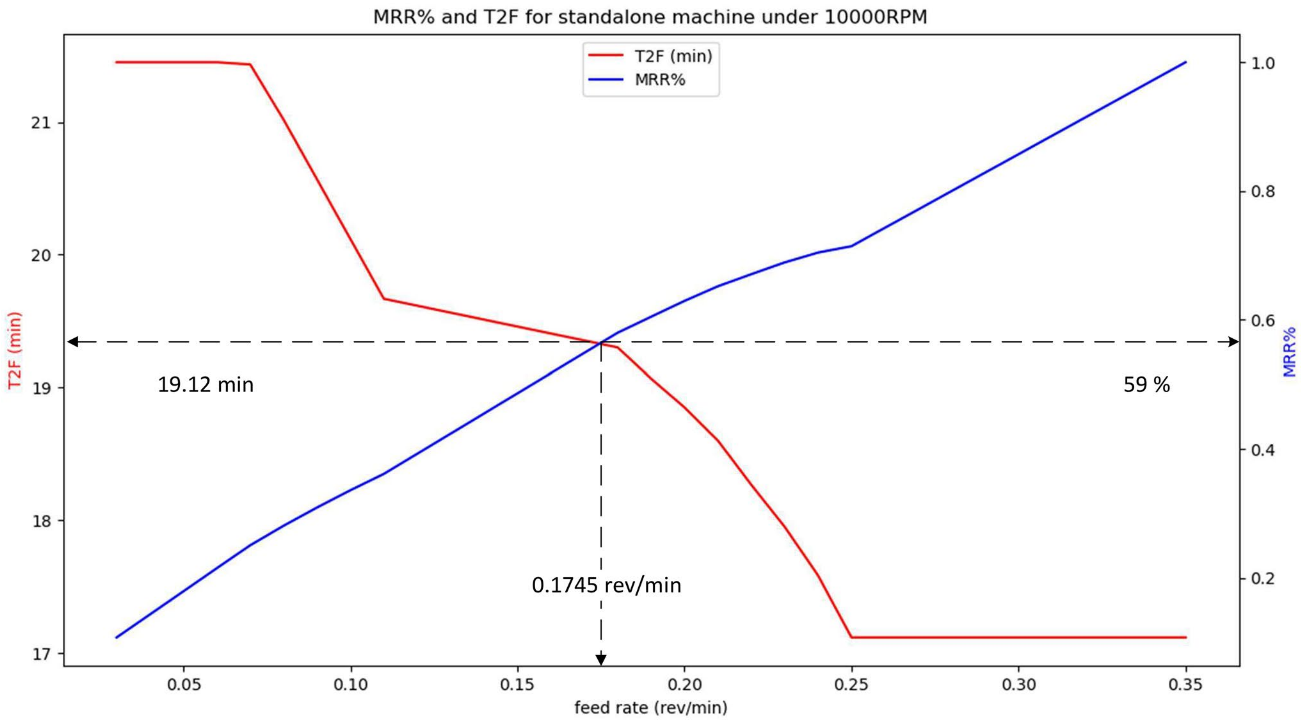


Fig. 15 T2F and MRR% for a standalone machine at a spindle speed of 10000 RPM and different feed rates

failure point (P) is detected, then the feed rate is adjusted according to the learned optimal policy. Cumulative Moving Average (CMA) at each time step is plotted as the average of the Pre-Failure agent's action up to the current time step. The Pre-Failure agent generates a variable feed rate at each second to maximize the T2F within the P-F zone.

The Pre-Failure agent keeps the productivity of the CNC machine at an acceptable limit, as the minimum CMA feed rate is kept at 0.2665 mm/rev. The machine's productivity index is the Material Removal Rate MRR (mm^3/min) in Eq. 2. For a 0.2 mm depth of cut and 25120 mm/min cutting speed, the change of the MRR with the Pre-Failure agent is indicated by the green shaded area in Fig. 12. $MRR\%$ is the Pre-Failure agent's overall productivity MRR_{PF} relative to the standalone machine $MRR_{maxfeed}$ at the maximum feed rate given by Eq. 8. $T2F_{st}$ is the standalone machine's T2F, and $T2F_{PF}$ is the T2F that includes the Pre-Failure agent.

$$MMR\% = \frac{\sum_{t=0}^{T2F_{PF}} MMR_{PF}}{\sum_{t=0}^{T2F_{st}} MRR_{maxfeed}} = \frac{\sum_{t=0}^{T2F_{PF}} f(t)}{f_{max} \times T2F_{st}} \tag{8}$$

In Run I, the Pre-Failure machine productivity is almost the same as the standalone machine with the maximum feed rate, and the $MRR\%$ equals 99.3%. The extension in T2F enables the machine to produce more within the added time. Figure 12 shows that the Pre-Failure added-value MRR recovers the lost MRR.

Figure 13 concludes the extension of the tool's Time to Failure (T2F) by the Pre-Failure agent over the standalone machine. The tool's added lifetime is high with relatively low spindle speeds, and it is small with high speeds. The lowest T2F added time is 1.4 min (37%) for the tool that works on 12500 RPM and the highest added time is 10.9 min (50%) with 7500 RPM. At higher cutting speeds, the tool degrades faster, the P-F interval is smaller, and the Pre-Failure agent has a smaller time to interact with the CNC machine environment. In the meantime, the T2F added time adds more valuable MRR at high speeds, as stated in Fig. 14. The Pre-Failure agent keeps the level of productivity high, as given in Fig. 14, while the tool deviation from the potential failure level is considered as discussed in Sect. 4.2. The lowest Pre-Failure agent's $MRR\%$ is 79%, which is achieved at a spindle speed of 10000 RPM, as given in Fig. 14. At 10000 RPM, the tool T2F increases to 2.133 min (128 sec), and the added-value MRR with the Pre-Failure agent is lower than the lost one due to a decrease in the feed rate value. Figure 15 depicts the T2F and $MRR\%$ of the standalone machine at different static feed rates and a spindle speed of 10000 RPM. The best standalone static feed rate setting is 0.1745 mm/rev, which aims to increase both MRR and T2F; the achieved MRR is 59% of that at fmax, and tool T2F is 19.12 min. The proposed Pre-Failure mechanism outperforms the standalone machine with static settings, and the online agent's optimal policy achieves more T2F extension and $MRR\%$, as given in Figs. 13 and 14.

Implementation of the developed Pre-Failure agent improves the tool's performance in the P-F zone. The online optimal feed rate continuous adjustment adds on average of 5 min to the tool T2F and 5% to the MRR over the classical machining system. The detailed experimental results for each run in Table 6 are provided in Appendix 2.

6 Conclusion

In this paper, the developed Pre-Failure approach improves the tool's performance in the Pre-Failure zone based on Deep Reinforcement Learning (DRL) during machining processes. The proposed Pre-Failure agent increases the tool Time to Failure (T2F) while maintaining the Material Removal Rate (MRR) at an acceptable limit. The machine tool's P-F curves and Logical Analysis of Data (LAD) are implemented to monitor and detect the potential failure level of the machine's tool. In the P-F zone, Pre-Failure model-free agent interacts with the CNC machine and adjusts its feed rate according to the estimated machine's forces at time (t+1). This method decreases the tool's degradation rate in the P-F zone before the tool is worn out, at VB = 0.2mm. The Pre-Failure mechanism also keeps the forces at a relatively high level. To train the Pre-Failure agent, a machine Digital Twin (DT) was developed and validated with the physical machine data. The Pre-Failure agent is validated at different spindle speeds, starting from 5000 RPM to 15,000 RPM. By implementing the proposed Pre-Failure approach, the tool T2F increases over the classical machining approach. The value-added time is high at relatively low spindle speeds. It was found that the maximum added time is 10.9 min, which is achieved with 7500 RPM. Meanwhile, at 15,000 RPM, the tool T2F equals 4.55 min, which is almost double the standalone machine. In the P-F zone, the Pre-Failure agent adds more MRR that recovers the lost MRR due to decreasing the adjusted feed rate to be lower than its maximum value. At high speeds, the added MRR is higher than the lost ones. The Pre-Failure agent's MRR reaches 138.04% of that achieved with a static maximum feed rate under 15,000 RPM spindle speed. At 1000 RPM, the Pre-Failure agent gets the lowest MRRR of 79%, relative to the standalone machine, and it adds 12% of the tool T2F. However, the added time is not enough to recover the lost MRR. The developed dynamic Pre-Failure agent outperforms the best static adjustment for standalone machine runs at 10,000 RPM from the perspective of tool life and productivity. The current work can be extended in the future by including electrical power consumption, the material type, and other machining quality characteristics (e.g., surface roughness and residual stresses).

Appendix 1

Section title of first appendix

The proposed pre-failure DDPG architecture consists of two deep actor-critic networks with two hidden layers and the hyperparameters in Table A. The DRL performance depends on its hyperparameters, and it is related to the environment/application dimension space of actions and state [2, 17, 29, 40]. These hyperparameters are adopted from literature on computer game applications that have the same data dimensions of actions and sensors as the CNC turning machine [29]. Figure 16 shows the developed Pre-Failure agent architecture.

Hidden neurons	Discount factor	Batch size	Learning rate critic	Learning rate actor	Target Update rate	Memory size
256/128	0.995	128	1e-4	1e-4	1e-3	1e6

Appendix 2

Other runs figures

In this section, the detailed results of runs II, III, IV, and V are stated.

Run II, the spindle speed is 7500 RPM

The Pre-Failure agent interaction in this run is demonstrated by Fig. 17.

In Run III, the spindle speed is 10000 RPM

The Pre-Failure agent interaction in this run is given by Fig. 18.

In Run IV, spindle speed is 12,500 RPM

The Pre-Failure agent interaction in this run is depicted by Fig. 19.

In Run V, the spindle speed is 15,000 RPM

The Pre-Failure agent interaction in this run is shown by Fig. 20.

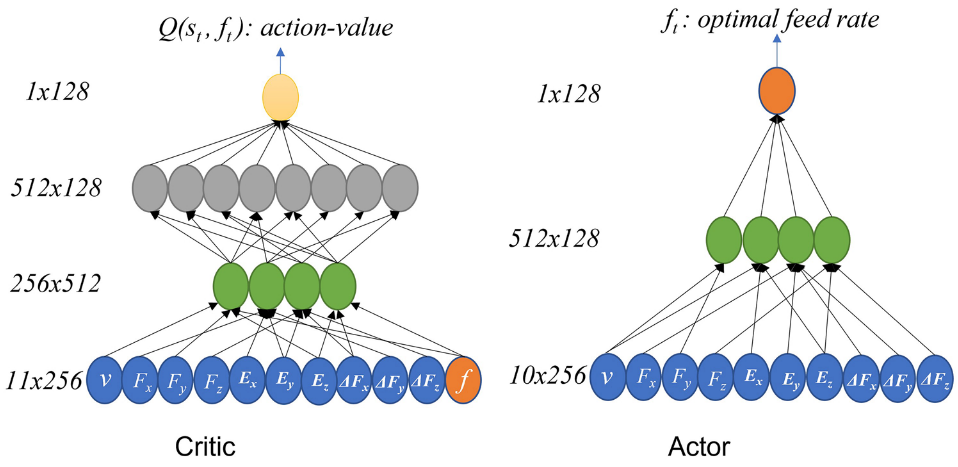


Fig. 16 Pre-failure DDPG agent architecture for CNC tool performance

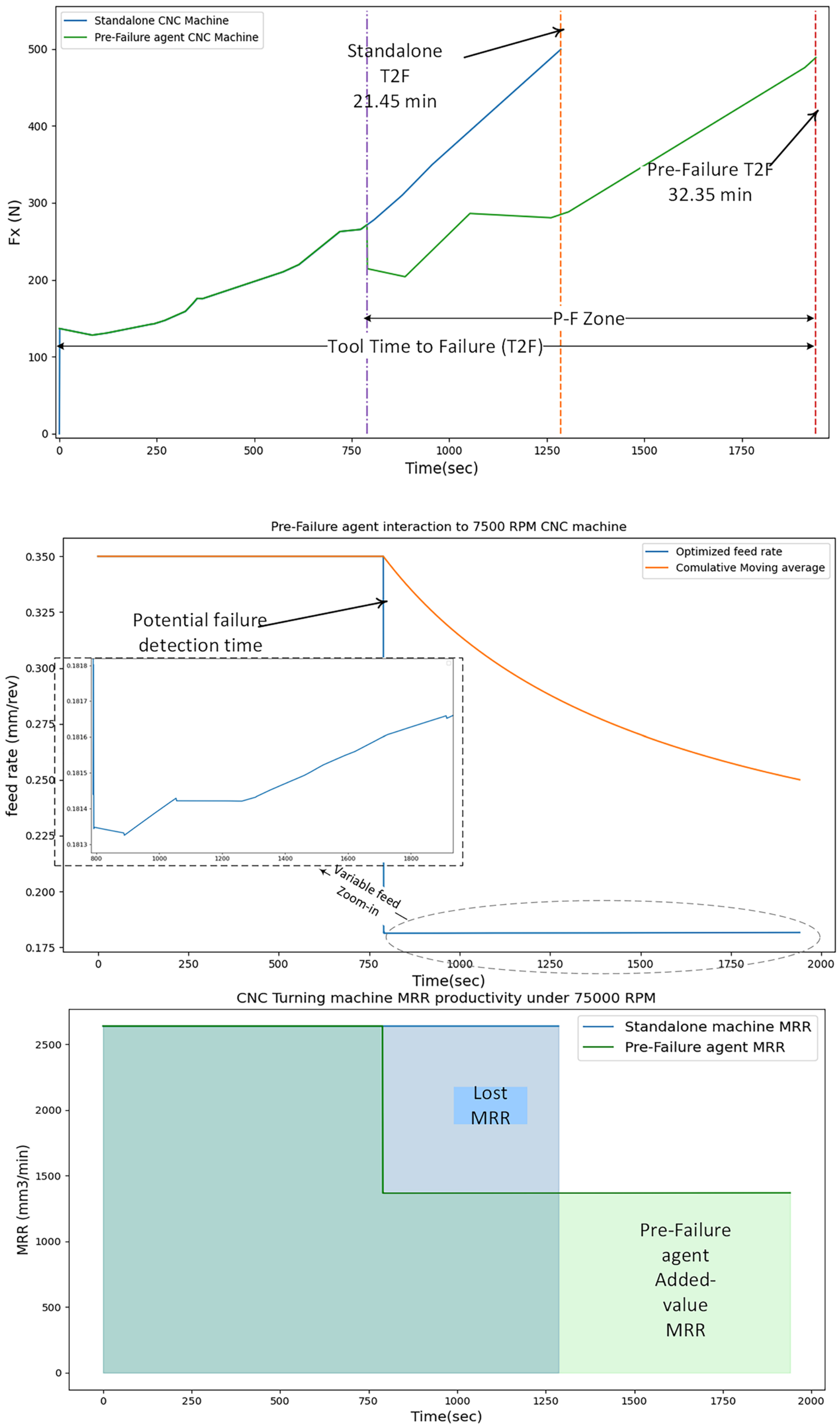
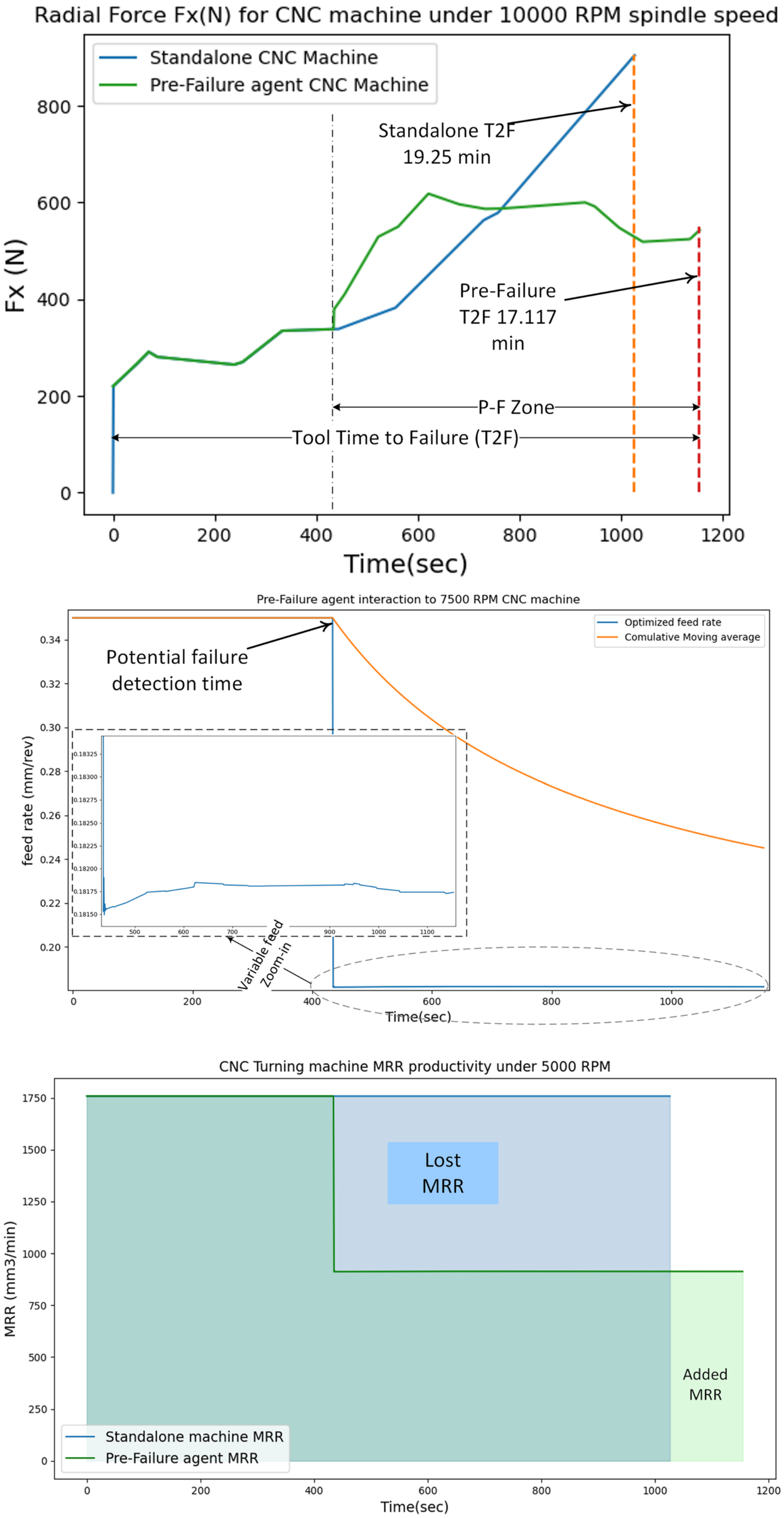


Fig. 17 Pre-Failure agent interaction in Run II, the spindle speed is 7500 RPM

Fig. 18 Pre-Failure Agent Interaction in Run III, the spindle speed is 10000 RPM



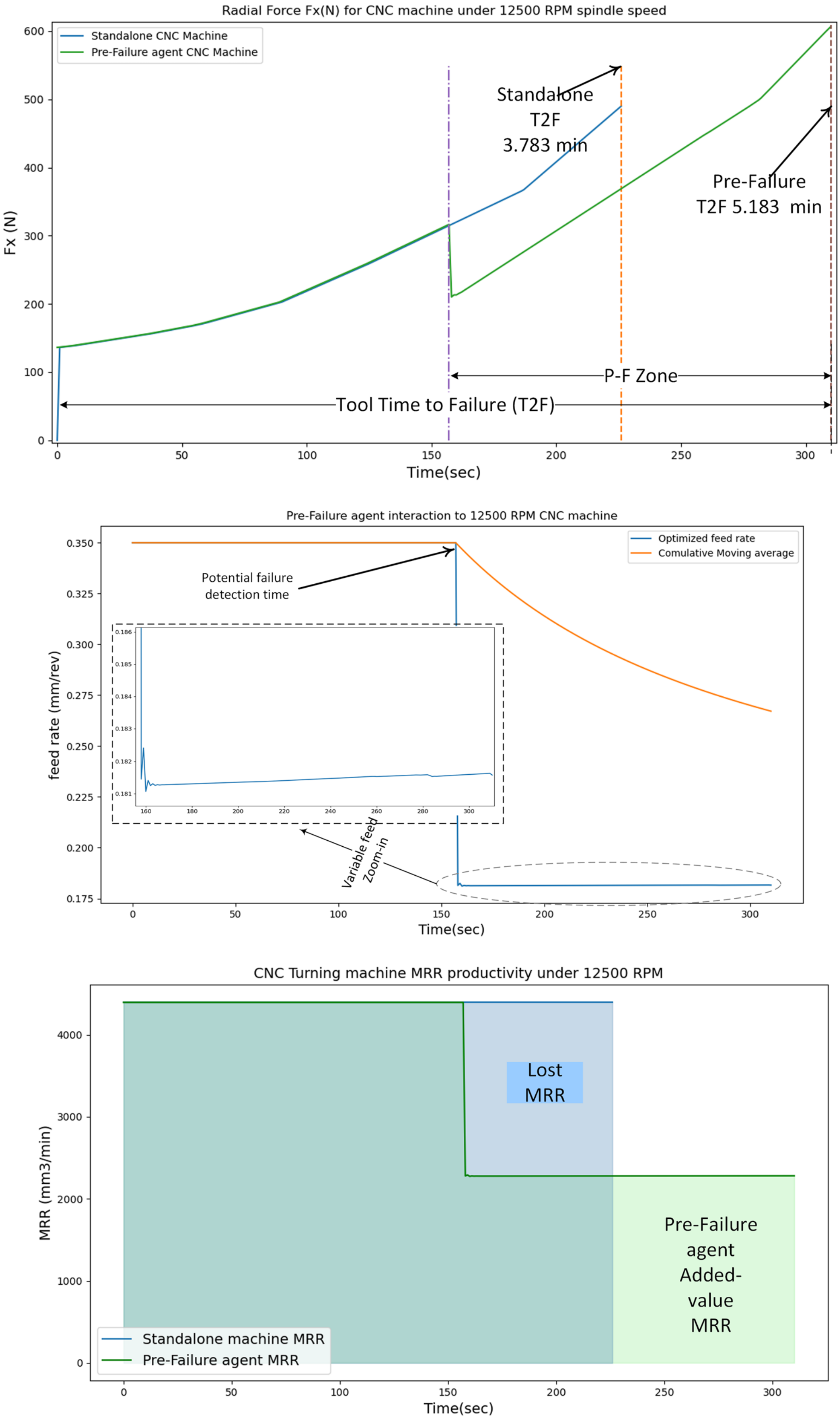
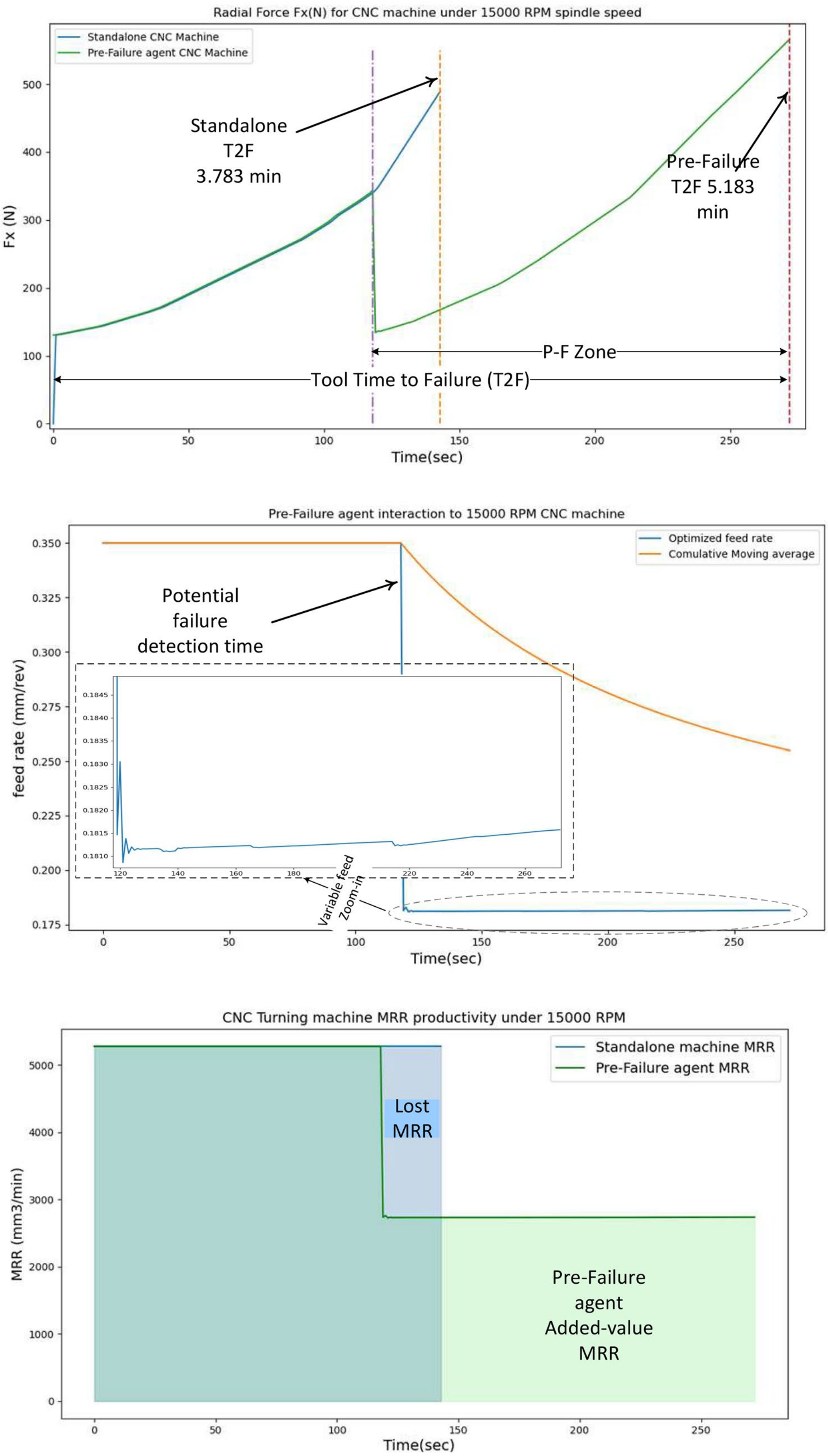


Fig. 19 Pre-Failure Agent Interaction in Run IV, spindle speed is 12,500 RPM

Fig. 20 Pre-Failure Agent Interaction in Run V, the spindle speed is 15,000 RPM



Radial Force Fx(N) for CNC machine under 15000 RPM spindle speed

Pre-Failure agent interaction to 15000 RPM CNC machine

CNC Turning machine MRR productivity under 15000 RPM

Acknowledgements The authors acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would like to thank Dr. Hussien Hegab for his valuable discussions and his support for this research perspective.

Funding This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), Prof. Soumaya Yacout, grant reference RGPIN-05785-2017.

Availability of data and materials Data are available.

Code availability Code is available.

Declarations

Ethics approval The authors confirm that this work does not contain any studies with human participants performed by any of the authors.

Consent to participate Not applicable.

Consent for publication The author grants the Publisher the sole and exclusive license of the full copyright in the Contribution, which license the Publisher hereby accepts.

Conflict of interest The authors declare no competing interests.

References

1. Lee J, Ardakani HD, Yang S, Bagheri B (2015) Industrial big data analytics and cyber-physical systems for future maintenance & service innovation. Procedia CIRP 38:3–7. https://doi.org/10.1016/j.procir.2015.08.026. https://www.sciencedirect.com/science/article/pii/S2212827115008744. Proceedings of the 4th International Conference on Through-life Engineering Services
2. Spielberg S, Tulsyan A, Lawrence NP, Loewen PD, Bhushan Gopaluni R (2019) Toward self-driving processes: a deep reinforcement learning approach to control. AIChE Journal 65(10):e16689. https://doi.org/10.1002/aic.16689. https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.16689
3. Yacout S (2019) Industrial value chain research and applications for industry 4.0. In: In 4th North America Conference on Industrial Engineering and Operations Management, Toronto, Canada
4. Elsheikh A, Yacout S, Ouali MS, Shaban Y (2020) Failure time prediction using adaptive logical analysis of survival curves and multiple machining signals. J Intell Manuf 31(2):403–415. https://doi.org/10.1007/s10845-018-1453-4
5. Xiong G, Li ZL, Ding Y, Zhu L (2020) Integration of optimized feedrate into an online adaptive force controller for robot milling. Int J Adv Manuf Technol 106(3):1533–1542. https://doi.org/10.1007/s00170-019-04691-1
6. Abbas AT, Abubakr M, Elkaseer A, Rayes MME, Mohammed ML, Hegab H (2020) Towards an adaptive design of quality, productivity and economic aspects when machining aisi 4340 steel with wiper inserts. IEEE Access 8:159206–159219. https://doi.org/10.1109/ACCESS.2020.3020623
7. Abbas AT, Sharma N, Anwar S, Hashmi FH, Jamil M, Hegab H (2019) Towards optimization of surface roughness and productivity aspects during high-speed machining of Ti-6Al-4V. Materials 12(22):3749. https://doi.org/10.3390/ma12223749
8. Park HS, Tran NH (2014) Development of a smart machining system using self-optimizing control. Int J Adv Manuf Technol 74(9–12):1365–1380. https://doi.org/10.1007/s00170-014-6076-0
9. Ridwan F, Xu X, Liu G (2012) A framework for machining optimisation based on STEP-NC. J Intell Manuf 23(3):423–441. https://doi.org/10.1007/s00170-014-6076-0
10. Stemmler S, Abel D, Adams O, Klocke F (2016) Model predictive feed rate control for a milling machine. IFAC-PapersOnLine 49(12):11–16. https://doi.org/10.1016/j.ifacol.2016.07.542
11. Shaban Y, Aramesh M, Yacout S, Balazinski M, Attia H, Kishawy H (2014) Optimal replacement of tool during turning titanium metal matrix composites. In: Proceedings of the 2014 Industrial and Systems Engineering Research Conference
12. Shaban Y, Meshreki M, Yacout S, Balazinski M, Attia H (2017) Process control based on pattern recognition for routing carbon fiber reinforced polymer. J Intell Manuf 28(1):165–179. https://doi.org/10.1007/s10845-014-0968-6
13. Shaban Y, Yacout S, Balazinski M (2015) Tool wear monitoring and alarm system based on pattern recognition with logical analysis of data. J Manuf Sci Eng 137(4). https://doi.org/10.1115/1.4029955
14. Sadek A, Hassan M, Attia M (2020) A new cyber-physical adaptive control system for drilling of hybrid stacks. CIRP Ann 69(1):105–108. https://doi.org/10.1016/j.cirp.2020.04.039
15. Shaban Y, Aramesh M, Yacout S, Balazinski M, Attia H, Kishawy H (2017) Optimal replacement times for machining tool during turning titanium metal matrix composites under variable machining conditions. Proc Inst Mech Eng B J Eng Manuf 231(6):924–932. https://doi.org/10.1177/0954405415577591
16. Taha HA, Yacout S, Shaban Y (2022) Autonomous self-healing mechanism for a CNC milling machine based on pattern recognition. J Intell Manuf 1–21. https://doi.org/10.1007/s10845-022-01913-4
17. Ma Y, Zhu W, Benton MG, Romagnoli J (2019) Continuous control of a polymerization system with deep reinforcement learning. J Process Control 75:40–47. https://doi.org/10.1016/j.jprocont.2018.11.004
18. Zhang Y, Li Y, Xu K (2022) Reinforcement learning-based tool orientation optimization for five-axis machining. Int J Adv Manuf Technol 119(11):7311–7326. https://doi.org/10.1007/s00170-022-08668-5
19. Xiao Q, Li C, Tang Y, Li L (2021) Meta-reinforcement learning of machining parameters for energy-efficient process control of flexible turning operations. IEEE Trans Autom Sci Eng 18(1):5–18. https://doi.org/10.1109/TASE.2019.2924444
20. Ochella S, Shafiee M, Sansom C (2021) Adopting machine learning and condition monitoring pf curves in determining and prioritizing high-value assets for life extension. Expert Syst Appl 176. https://doi.org/10.1016/j.eswa.2021.114897
21. Bennane A, Yacout S (2012) LAD-CBM; new data processing tool for diagnosis and prognosis in condition-based maintenance. J Intell Manuf 23(2):265–275. https://doi.org/10.1007/s10845-009-0349-8
22. Singh G, Gupta MK, Mia M, Sharma VS (2018) Modeling and optimization of tool wear in MQL-assisted milling of Inconel 718 superalloy using evolutionary techniques. Int J Adv Manuf Technol 97(1):481–494. https://doi.org/10.1007/s00170-018-1911-3
23. M'Saoubi R, Axinte D, Soo SL, Nobel C, Attia H, Kappmeyer G, Engin S, Sim WM (2015) High performance cutting of advanced aerospace alloys and composite materials. CIRP Annals 64(2):557–580. https://doi.org/10.1016/j.cirp.2015.05.002. https://www.sciencedirect.com/science/article/pii/S0007850615001419
24. Aramesh M, Shaban Y, Yacout S, Attia M, Kishawy H, Balazinski M (2016) Survival life analysis applied to tool life estimation with variable cutting conditions when machining titanium metal matrix composites (TI-MMCS). Mach Sci Technol 20(1):132–147. https://doi.org/10.1080/10910344.2015.1133916
25. Montgomery DC (2007) Introduction to statistical quality control. John Wiley & Sons
26. Satopaa V, Albrecht J, Irwin D, Raghavan B (2011) Finding a "kneedle" in a haystack: detecting knee points in system behavior.

In: 2011 31st International Conference on Distributed Computing Systems Workshops. pp 166–171. https://doi.org/10.1109/ICDCSW.2011.20

27. Lejeune M, Lozin V, Lozina I, Ragab A, Yacout S (2019) Recent advances in the theory and practice of logical analysis of data. Eur J Oper Res 275(1):1–15. https://doi.org/10.1016/j.ejor.2018.06.011
28. Shaban Y, Yacout S, Balazinski M, Jemielniak K (2017) Cutting tool wear detection using multiclass logical analysis of data. Mach Sci Technol 21(4):526–541. https://doi.org/10.1080/10910344.2017.1336177
29. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2015) Continuous control with deep reinforcement learning. arXiv preprint arXiv:150902971
30. Barde SR, Yacout S, Shin H (2019) Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks. J Intell Manuf 30(1):147–161. https://doi.org/10.1007/s10845-016-1237-7
31. Shafto M, Conroy M, Doyle R, Glaessgen E, Kemp C, LeMoigne J, Wang L (2012) Modeling, simulation, information technology & processing roadmap. National Aeronautics and Space Administration 32(2012):1–38
32. Taha HA, Sakr AH, Yacout S (n.d.) Aircraft engine remaining useful life prediction framework for industry 4.0
33. Yao J, Lu B, Zhang J (2022) Tool remaining useful life prediction using deep transfer reinforcement learning based on long short-term memory networks. Int J Adv Manuf Technol 118(3):1077–1086. https://doi.org/10.1007/s00170-021-07950-2
34. Glaessgen E, Stargel D (2012) The digital twin paradigm for future NASA and US Air Force vehicles. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA. p 1818. https://doi.org/10.2514/6.2012-1818
35. AboElHassan A, Sakr A, Yacout S (2021) A framework for digital twin deployment in production systems. In: Weißgraeber P, Heieck F, Ackermann C (eds) Advances in Automotive Production Technology - Theory and Application. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 145–152
36. Qi Q, Tao F, Hu T, Anwer N, Liu A, Wei Y, Wang L, Nee A (2021) Enabling technologies and tools for digital twin. J Manuf Syst 58:3–21. https://doi.org/10.1016/j.jmsy.2019.10.001
37. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT press
38. Hagan Martin T, Demuth Howard B, Beale Mark H et al (2002) Neural network design. University of Colorado at Boulder
39. Brownlee J (2016) Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras. Machine Learning Mastery
40. Traue A, Book G, Kirchgässner W, Wallscheid O (2022) Toward a reinforcement learning environment toolbox for intelligent electric motor control. IEEE Transactions on Neural Networks and Learning Systems 33(3):919–928. https://doi.org/10.1109/TNNLS.2020.3029573

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