Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Learning to tune: AI-enabled virtual commissioning for CNC interpolator parameter optimization

You-Cheng Chen
National Formosa University
Yong-Zhong Li
National Formosa University
Ming-Tsung Lin
mtlin@nfu.edu.tw

National Formosa University https://orcid.org/0000-0002-9243-7896

Research Article

Keywords: Virtual commissioning, Reinforcement learning, CNC interpolator parameter optimization, Surrogate model, Contouring error

Posted Date: November 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11533444/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Learning to tune: AI-enabled virtual commissioning for CNC interpolator parameter optimization

You-Cheng Chen [1], Yong-Zhong Li [1], Ming-Tsung Lin [1,2,*]

[1] Department of Mechanical Design Engineering, National Formosa University, No. 64, Wenhua Rd., Huwei, Yunlin 632301, Taiwan

[2] Smart Machine and Intelligent Manufacturing Research Center, National Formosa University, No. 64, Wenhua Rd., Huwei, Yunlin 632301, Taiwan, R.O.C.

*Corresponding author. E-mail: mtlin@nfu.edu.tw (M.-T. Lin);

Contributing authors: 50733143@gm.nfu.edu.tw (Y.-C. Chen); 50733140@gm.nfu.edu.tw (Y.-Z. Li).

Abstract

CNC interpolator commissioning still relies heavily on trial-and-error manual tuning, limiting repeatability and efficiency. This study presents an AI-enabled virtual commissioning framework that integrates a back-propagation neural network (BPNN) surrogate with CrossQ reinforcement learning for interpolator parameter optimization. Trained on 729 full-factorial design-of-experiments samples, the surrogate is evaluated using ten-fold out-of-fold validation and 600 independent off-grid samples, enabling continuous parameter search without repeated on-machine trials. Pearson correlation and SHAP analyses identify influential interpolator parameters. A range-normalized reward balances cycle time against soft penalties for contouring error and jerk. Sensitivity and ablation analyses across ten random seeds demonstrate that stronger jerk penalization reduces jerk violations with less than a 0.5% increase in cycle time. Compared with BP-GA and BP-TD3, BP-CrossQ is the only surrogate-assisted method satisfying both contouring accuracy and jerk targets on the KAKINO benchmark under semi-finish and finishing. A frozen, condition-specific policy attains 100% surrogate-predicted feasibility

from 50 initial settings per condition without retraining, showing consistent reusable parameter refinement. Transferring the optimized settings to a FANUC test mold on the same machine reduces maximum contouring error by 30.97% and 12.61% and cycle time by 1.20% and 1.43%, respectively, relative to manual tuning. Total commissioning time decreases from 24 h to about 11 h, supporting efficient, reusable virtual commissioning and cross-trajectory performance improvement within the evaluated machine and conditions.

Keywords: Virtual commissioning; Reinforcement learning; CNC interpolator parameter optimization; Surrogate model; Contouring error

1. Introduction

1.1. Background and motivation

The rapid development of Industry 4.0 has accelerated the integration of artificial intelligence (AI), industrial internet of things (IIoT), edge computing, and digital twin (DT) technologies into smart manufacturing systems [1]-[8]. In CNC machining, these technologies have created new opportunities for data-driven process monitoring, prediction, optimization, and intelligent control. CNC machine tools are increasingly equipped with edge-computing devices that acquire high-frequency drive and motion signals, providing a foundation for extending AI-based methods from process-level applications to controller-level decision-making. Although AI has been widely applied to cutting-force modeling, tool-wear prediction, and surface-quality prediction [3]-[6], virtual commissioning of CNC controllers, particularly the automated tuning of interpolator parameters, remains underexplored.

In practice, CNC controller commissioning is iterative because strongly coupled parameters must be tuned to satisfy accuracy and efficiency requirements under diverse machining conditions, feedrates,

and toolpath geometries. Field engineers perform repeated dynamics analysis, cutting tests, and manual tuning, a procedure that typically requires approximately three working days (24 h). Because the resulting settings rely heavily on individual experience, they are difficult to reproduce and transfer across operating conditions. These challenges motivate an AI-enabled virtual commissioning approach in which the physical machine is used primarily for data acquisition, while parameter optimization is conducted within a surrogate virtual environment under prescribed machining performance and kinematic constraints.

1.2. Related work and research gaps

Research on intelligent CNC interpolation falls into four categories: interpolator modeling, learning-based motion optimization, data-driven error prediction and compensation, and interpolator parameter optimization. Studies in the first category show that the internal behavior and parameter settings of commercial interpolators substantially affect machining performance. Petráček et al. [9] quantified the effects of contour tolerance, limiting frequency, and jerk-related settings on toolpath precision, particularly in corner regions of a Heidenhain interpolator. Ward et al. [10] showed that explicitly modeling interpolation dynamics, including feedrate planning and corner blending, is essential for accurately predicting the feedrate profile and cycle time of complex toolpaths. However, these studies focus primarily on modeling and prediction rather than automated commissioning.

Model-based and learning-based approaches have subsequently been applied to improve motion optimization. Kim et al. [11] combined an uncertainty-aware DT with model predictive control to maximize feedrate while satisfying contouring error constraints. Lee et al. [12] optimized interpolator

parameters using a feed-drive simulation model and a genetic algorithm (GA), reducing trajectory error without increasing cycle time. Within reinforcement learning (RL) [13], constraint-aware motion planning has been realized by integrating velocity, acceleration, and jerk limits into the learning framework [14]. In CNC applications, Li et al. [15] adopted RL for direct trajectory smoothing in which the agent generates servo commands from the current toolpath and motion state, replacing part of the conventional interpolation procedure. In contrast, Jiang et al. [16] retained the conventional CNC architecture and adopted a time-series deep Q-network to tune contouring error compensation parameters rather than the low-level control law. Parameter-level RL has proven effective, but direct interaction with physical machines remains time-consuming and may involve undesirable exploratory actions.

Data-driven models provide an alternative means of representing nonlinear machine behavior. GRU- and LSTM-based models have been used for tracking error and contouring error prediction, feedforward compensation, and reference pre-compensation [17]-[20]. Ji et al. [21] proposed a DT-based hybrid model in which XGBoost learns the residual of a parametric feed-drive model from sparse time series, enabling adaptive model updating. However, these models are used primarily for error prediction or trajectory compensation rather than interactive interpolator commissioning.

Surrogate-assisted optimization has also been applied to CNC controller tuning. Chiu and Lee [22] combined ANFIS with particle swarm optimization (PSO) using machining time, contouring error, and tracking error as performance metrics. Tseng et al. [23] employed BPNN-GA with corner vibration considerations. Lin et al. [24] adopted BPNN-NSGA-II for multiple performance objectives. Jhuang

et al. [25] combined GPR and XGBoost predictors with region partitioning and PSO to minimize machining time under a contouring-error limit. More recently, Chen et al. used a pre-trained BPNN as a virtual environment for DQN-based interpolator tuning [26] and extended the approach to CrossQ with soft contouring error and jerk constraints [27]. These studies demonstrate the feasibility of surrogate-assisted optimization for CNC interpolator commissioning.

Table 1 Comparison of representative data-driven approaches for CNC interpolator optimization.

Study	Optimization framework	Constraint handling	Interpretability and robustness	Physical validation and transferability
Chiu and Lee [22]	ANFIS + PSO	Weighted multi-objective optimization	None reported	Single workpiece (no transfer)
Tseng et al. [23]	BPNN + GA	Weighted multi-objective optimization	None reported	Square → KAKINO
Chen et al. [27]	BPNN + CrossQ	GMN-based soft penalties	Sensitivity	KAKINO → FANUC mold
Proposed method	BPNN + CrossQ with VC evaluation	RNRF with sensitivity and ablation	Off-grid validation SHAP + ablation + initial-state evaluation	KAKINO → FANUC mold semi-finish/finishing GA/TD3 comparison commissioning cost

Table 1 compares representative data-driven approaches in terms of optimization framework, constraint handling, interpretability, robustness, and physical validation. Although the preliminary CrossQ study [27] included sensitivity analysis and cross-trajectory testing, three research gaps remain. First, independent off-grid validation and near-boundary feasibility assessment are limited. Second, the individual effects of reward penalties, feature-level interpretation, and policy consistency across machining conditions and initial parameter settings require systematic investigation. Third, evidence of cross-condition physical performance and practical commissioning cost remains limited. Therefore,

this study integrates independent off-grid testing, interpretable constraint-aware policy optimization, and physical validation with commissioning-cost analysis.

1.3. Contributions

Building on the preliminary BP-CrossQ study [27], this work extends AI-enabled virtual commissioning for CNC interpolator parameter optimization through three contributions:

(1) Validated surrogate-based virtual commissioning: A BPNN surrogate trained on full-factorial DOE data is evaluated through ten-fold out-of-fold validation and independent off-grid testing, including near-boundary feasibility analysis, to support continuous parameter optimization.

(2) Interpretable and constraint-aware policy learning: SHAP analysis, reward-weight sensitivity, penalty ablation, and ten-seed evaluations under semi-finish and finishing conditions quantify how contouring error and jerk penalties shape policy behavior and constraint satisfaction.

(3) Comparative and deployment validation: BP-CrossQ is compared with manual tuning, BP-GA, and BP-TD3 under identical parameter bounds and constraint settings. Initial-state tests assess surrogate-predicted feasibility consistency and policy reuse without retraining, while physical machining and commissioning-time analyses evaluate cross-trajectory performance transfer and practical efficiency.

The remainder of this paper is organized as follows. Section 2 formulates the CNC interpolator optimization problem. Section 3 describes the AI-enabled virtual commissioning framework. Section 4 presents the BP-CrossQ agent design. Section 5 reports simulation and experimental results, followed by conclusions and future work in Section 6.

2. CNC interpolation system and problem formulation

This section introduces CNC controller architecture, benchmark trajectories, AICC interpolator parameters, and motion performance metrics, and formulates the parameter optimization problem.

2.1. Architecture of CNC control system

This study employs a FANUC 0i-MF Plus controller to investigate the dynamic effects of AI Contour Control (AICC) interpolator parameters on motion performance. Fig. 1 illustrates the CNC controller architecture, which comprises an interpolator and a cascaded servo control scheme. The interpolator includes the functions of lookahead, acceleration/deceleration before interpolation (ADBI), and acceleration/deceleration after interpolation (ADAI). The reference commands generated by the interpolator are fed to the servo loops of each motion axis in feed drive systems. The servo parameters [28][29] are maintained at the nominal settings so that the investigation focuses exclusively on AICC interpolator commissioning.

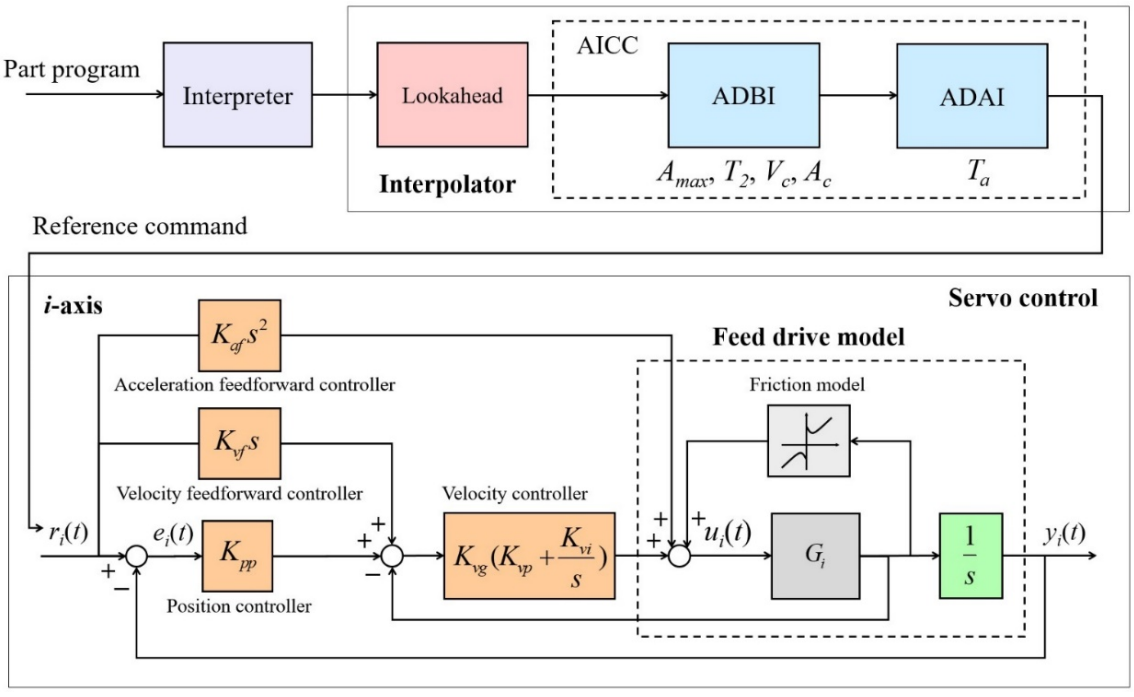


Fig. 1. Architecture of the CNC controller.

2.2. Benchmark and target trajectories

The KAKINO test path [30], illustrated in Fig. 2(a), is a standardized benchmark trajectory that manufacturers typically apply on the XY, YZ, and XZ planes to evaluate the dynamic performance of CNC machine tools. It comprises six geometric features representative of practical machining operations: three corner geometries (45°, 90°, and 135°) and three curvature transitions (straight-to-arc, arc-to-straight, and arc-to-arc). These transitions impose abrupt changes in curvature and motion direction, challenging feedrate planning and path-tracking capabilities of the interpolator. In this study, the KAKINO path is adopted for optimizing the AICC interpolator parameters. The FANUC test mold in Fig. 2(b) is used as the target workpiece for independent validation. This two-stage arrangement allows the optimized parameters identified from the benchmark trajectory to be reevaluated in practical mold machining, assessing their cross-trajectory transferability.

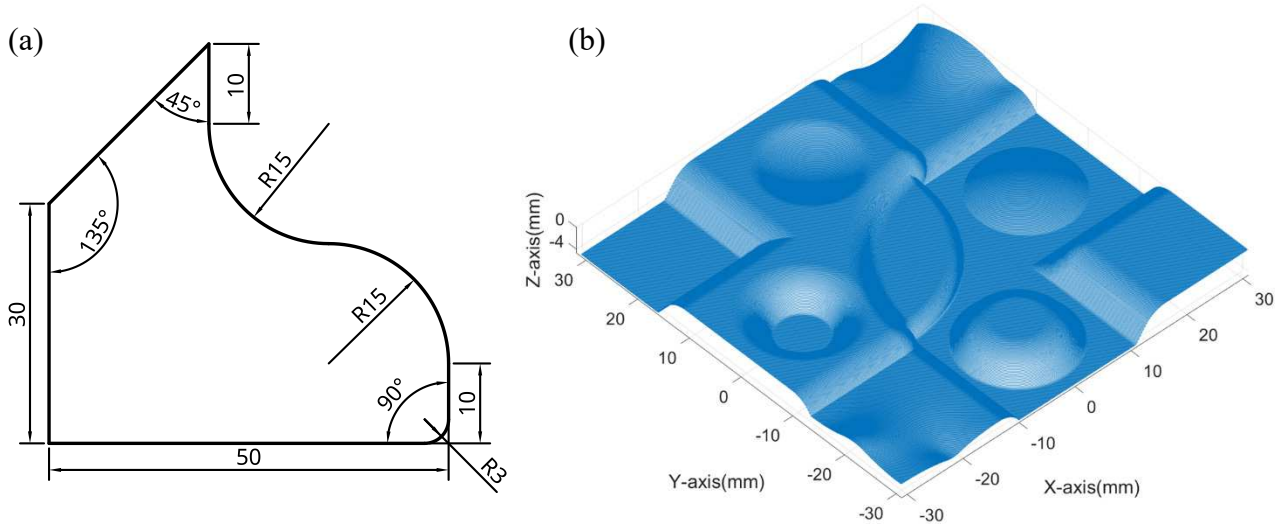


Fig. 2. Standard benchmark and target trajectories (a)KAKINO, (b)FANUC test mold.

2.3. AICC interpolator parameters

The AICC function [28] is designed to shorten cycle time while sustaining the accuracy required for high-speed, high-precision machining. The relevant AICC ACC/DEC interpolator parameters are

summarized in Table 2, and their roles in feedrate planning are illustrated in Fig. 3.

For a linear segment commanded by G01, trajectory interpolation is generated using the jerk-limited bell-type velocity profile [29][31] shown in Fig. 3(a). Each motion axis is subject to a maximum allowable acceleration A_{max} determined by the driving capability of the servo motor. The parameter T_2 defines the time constant in the ADBI stage and sets the jerk level during motion transitions, with the nominal jerk given by $J=1000A_{max}/T_2$. A smaller T_2 value results in faster acceleration changes, which may induce structural vibration and degrade surface quality. At sharp corners, the corner feedrate difference V_c, depicted in Fig. 3(b), regulates the allowable velocity transition between consecutive motion segments. Increasing V_c can shorten cycle time by permitting more aggressive corner transitions, but excessive values may amplify dynamic excitation and contour deviation, particularly at high feedrates. To mitigate these effects, the time constant T_a of the ADAI filters the velocity command and smooths tangential discontinuities shown in Fig. 3(c). However, larger T_a improves motion smoothness at the expense of contouring accuracy and cycle time. For circular interpolation, A_c specifies the allowable centripetal acceleration and constrains the tangential velocity V_{tc} through the local curvature κ as $V_{tc}=\mathrm{sqrt}(A_c/\kappa)$, shown in Fig. 3(d). This constraint is essential for small-radius arcs and high-speed circular motion, where excessive acceleration deteriorates contour accuracy.

Machining performance is not determined by any single parameter, but by the coupled effects of multiple interpolator settings. Aggressive ACC/DEC settings reduce cycle time while increasing jerk, tracking error, and contouring error, whereas conservative settings reverse this relationship. Because

these interactions are nonlinear and path-dependent, manual tuning becomes increasingly difficult as machining conditions and geometric features vary. The resulting trade-offs among efficiency, accuracy, and motion smoothness are summarized qualitatively in Section 2.4.

Table 2 AICC ACC/DEC interpolator parameters.

Parameter	Symbol	Unit	Definition
P1660	A_{max}	mm/s^2	Acceleration rate for ACC/DEC before interpolation
P1735	A_c	mm/s^2	Allowable acceleration rate for circular interpolation
P1769	T_a	ms	Time constant for ACC/DEC after interpolation
P1783	V_c	mm/min	Corner feedrate difference
P1772	T_2	ms	Time constant for bell-type acc/dec before interpolation

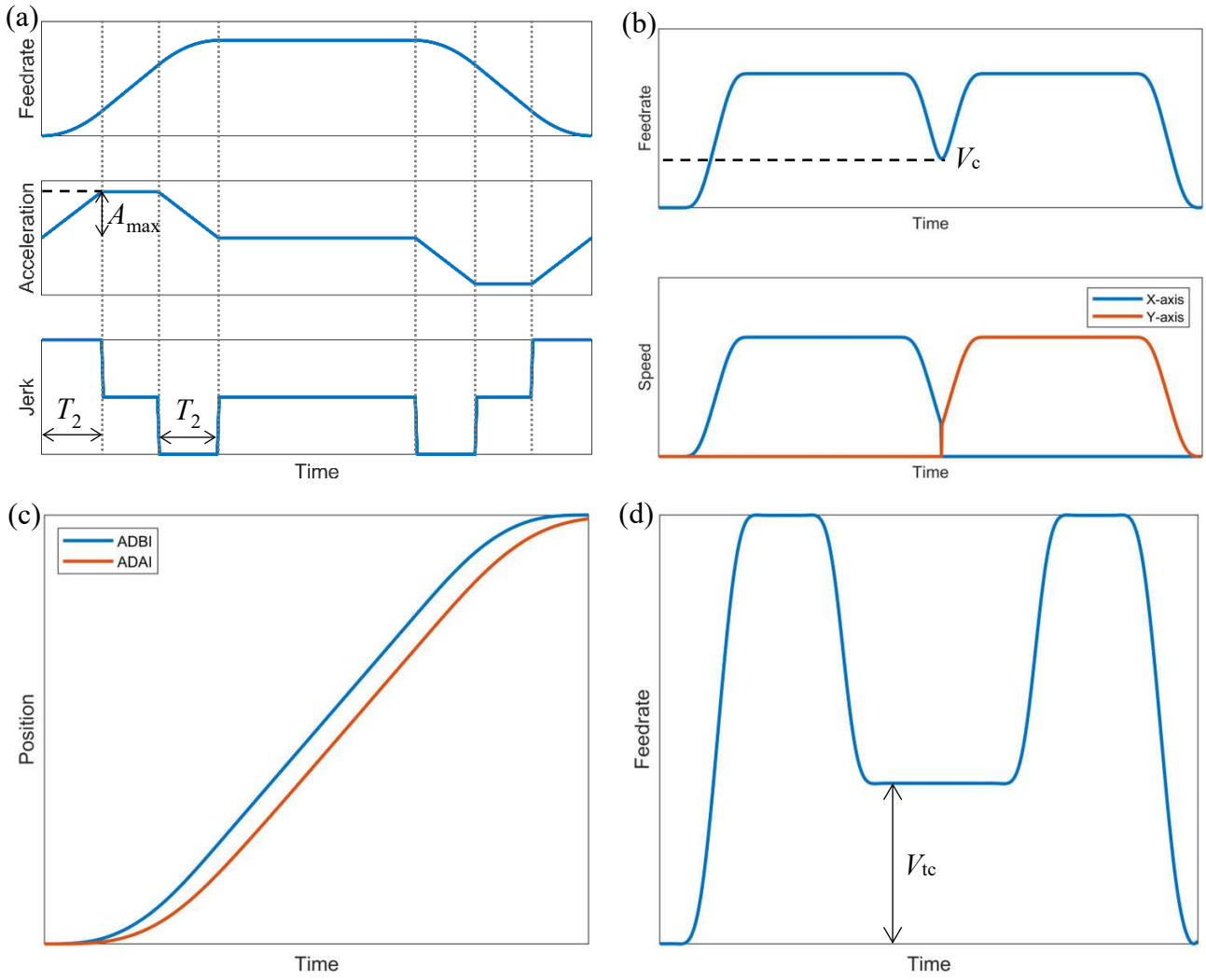


Fig. 3. Feedrate planning based on AICC ACC/DEC interpolator parameters. (a)Bell-type velocity profile, (b)Corner interpolation, (c)ADAI, (d)Circular interpolation.

2.4. Motion performance metrics

Machining performance is commonly evaluated in terms of speed, accuracy, and surface quality, which in CNC motion systems depend on the dynamic behavior of the interpolator, servo drives, and mechanical structure. Accordingly, this study defines cycle time (CT), contouring error (CE), and tracking error (TE) as motion performance metrics, representing machining efficiency, contouring accuracy, and servo-motion quality, respectively. Tracking error is adopted as an indicator for servo-motion quality rather than a direct measure of surface finish, since both are influenced by dynamic phenomena such as structural resonance, stick-slip, friction, and backlash, and their relationship is investigated in [22]. Table 3 summarizes the qualitative effects of decreasing individual AICC interpolator parameters on these metrics, following the tuning guidance in [29]. The trends illustrate that no single parameter adjustment can optimize all performance metrics simultaneously.

Table 3 Qualitative effects of decreasing individual AICC parameters on performance metrics.

Parameter	Symbol	Cycle time	Contouring error	Tracking error
P1660	A_{max}	↑ *	↓	↓
P1735	A_c	↑	↓	↓
P1769	T_a	↓	↓ **	↓ **
P1783	V_c	↑ *	↓	↓
P1772	T_2	↓ **	↑	↑

Note: ↑ and ↓ indicate an increase and decrease, respectively. Tuning priority recommended by FANUC: **high, *medium.

The tracking error of a single feed axis i shown in Fig. 1 is defined as $e_i=r_i-y_i$, where r_i and y_i denote the reference and actual axis positions, respectively. It reflects the dynamic response of the feed drive and is influenced by the commanded trajectory, servo control bandwidth, and machine dynamics noted above. For the three-axis machine tool considered here, the axial tracking errors are e_x, e_y, and

e_z. For multi-axis motion, the contouring error is illustrated in Fig. 4, where R and P represent the reference and actual positions, respectively, and R_f is the orthogonal projection of P onto the reference contour. The contouring error ε_c is defined as the shortest orthogonal distance between the actual position and reference trajectory [32]. It quantifies the geometric deviation of the tool-center path directly and is the primary measure of machining accuracy, particularly in high-speed machining of complex geometries. Cycle time t_{cyc} represents the total time required to traverse the prescribed toolpath and serves as the measure of machining efficiency.

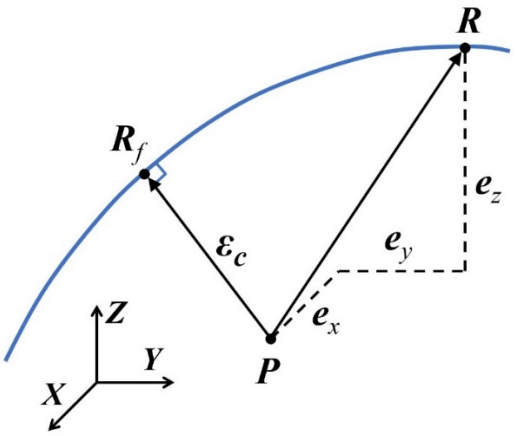


Fig. 4. Illustration of the tracking and contouring errors.

2.5. Optimization problem formulation

In practical CNC commissioning, the AICC interpolator parameters must be determined to minimize machining cycle time while satisfying the prescribed accuracy and dynamic performance requirements defined in Section 2.4. Hence, the AICC interpolator parameter optimization problem is formulated as:

$$\begin{aligned} &\min_{P} \ t_{cyc}(P) \\ &s.t. \quad g_l(P) \le 0, \quad g_u(P) \le 0, \quad g_c(P) \le 0, \quad g_j(P) \le 0 \end{aligned} \tag{1}$$

where P denotes the AICC interpolator parameter vector. The lower- and upper-bound constraints, g_l and g_u, restrict the parameters to the operating ranges permitted by the machine tool manufacturer. The

accuracy constraint, g_c, keeps the maximum contouring error below the prescribed tolerance, and the kinematic constraint, g_j, limits the nominal jerk, suppressing dynamic excitation that degrades surface quality. The latter two are engineering requirements rather than physical limits of the machine and are therefore enforced in the proposed RL implementation through soft reward penalties rather than hard projection. Because the mapping from P to the performance metrics is nonlinear, toolpath-dependent, and unavailable in closed form for the commercial controller, the optimization is performed within the surrogate-assisted RL framework introduced in Section 3, avoiding repeated on-machine trials of candidate parameter settings.

3. AI-enabled virtual commissioning framework

This section presents the AI-enabled virtual commissioning (VC) framework including the overall architecture, experimental design, statistical analysis, BPNN surrogate modeling, and SHAP-based feature interpretation.

3.1. Overview of virtual commissioning framework

Fig. 5 illustrates the proposed VC framework which consists of a physical machine tool, a FANUC Tuner, an industrial PC, a BPNN surrogate model, and a CrossQ agent. Machining data acquired through the FANUC Tuner are used to train the BPNN surrogate, which approximates the nonlinear mapping between interpolator parameters and motion performance metrics. The trained surrogate serves as a virtual machine environment in which the CrossQ agent learns a parameter-selection policy through reward-guided exploration. The physical machine is consequently required only for the initial data acquisition and the final validation, not for every candidate evaluation. The framework realizes a

learning-to-tune procedure, in which interpolator parameters are determined offline within the virtual environment before deployment to the physical machine.

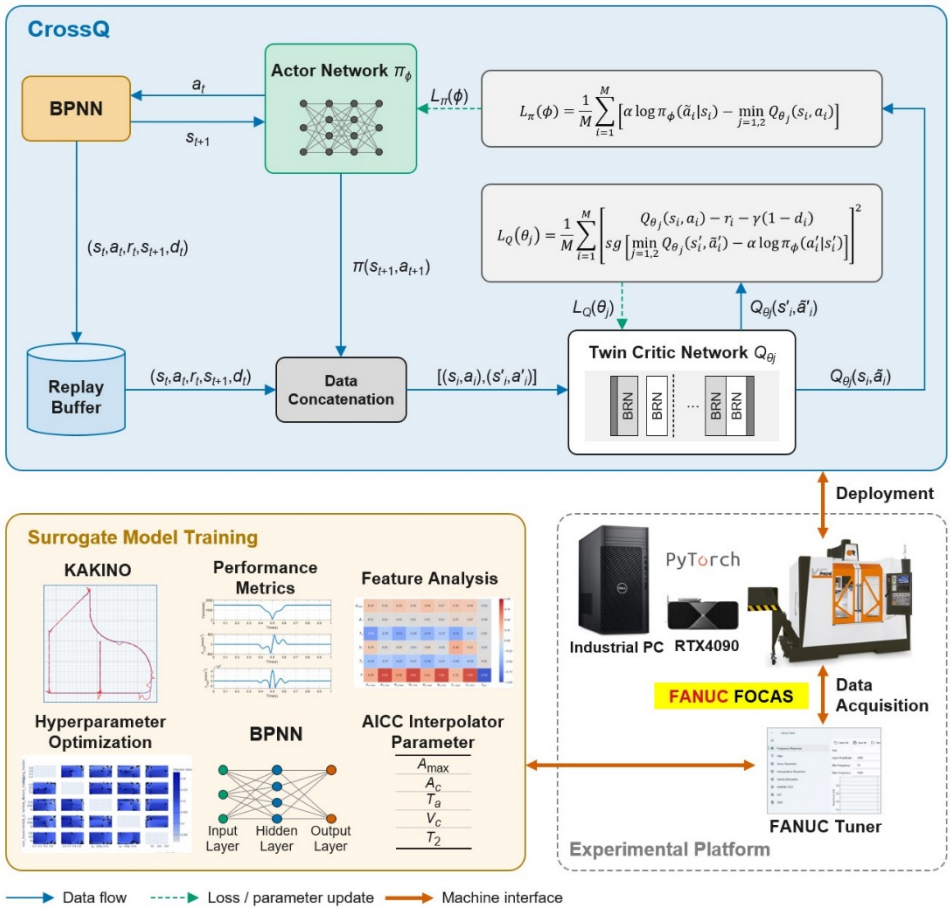


Fig. 5. AI-enabled virtual commissioning framework.

3.2. Design of experiments and dataset generation

A full-factorial Design of Experiments (DOE) approach was employed to investigate the influence of AICC interpolator parameters on motion performance and to generate the dataset required for surrogate model development. Six experimental factors including five AICC interpolator parameters and feedrate were considered, and each evaluated at three levels, as summarized in Table 4. The parameter levels span the operating ranges specified by the machine tool manufacturer, and the feedrate F of 3000, 1500, and 750 mm/min corresponds to roughing, semi-finish, and finishing conditions, respectively. Following recommendations from field engineers, identical AICC parameter settings

were applied to the *X*-, *Y*-, and *Z*-axes to ensure consistent motion behavior across all servo axes. The three-level full-factorial design yields 729 (3^6) experimental combinations, which capture both the individual and combined effects of the factors.

Nine performance metrics were selected as response variables: the maximum (MAX) and root-mean-square (RMS) tracking errors of three motion axes, MAX and RMS contouring errors, and cycle time. During data acquisition, interpolation commands and position feedback signals were collected through the FANUC FOCAS library interface at a sampling rate of 1 kHz, from which the tracking and contouring errors were computed.

Table 4 Full factorial design for factors and variables.

Factor	Level			Unit	Variable (MAX/RMS)	Unit
	1	2	3			
A_{max}	800	1500	3000	mm/s^2	$e_{x,max}$/$e_{x,rms}$	μm
A_c	240	1000	2000	mm/s^2	$e_{y,max}$/$e_{y,rms}$	μm
T_a	16	32	64	ms	$e_{z,max}$/$e_{z,rms}$	μm
V_c	60	320	640	mm/min	$\varepsilon_{c,max}$/$\varepsilon_{c,rms}$	μm
T_2	16	32	64	ms	t_{cyc}	s
F	750	1500	3000	mm/min		

3.3. Statistical and correlation analysis

Statistical analysis was first performed to verify data quality and identify the key factors affecting the motion performance responses. The 729 DOE samples were screened using the interquartile range (IQR) criterion and no outliers were detected. All 729 samples were retained for surrogate modeling. Pearson correlation analysis [33] was then performed to quantify the linear correlations between interpolator parameters and motion performance metrics. The Pearson correlation coefficient (PCC), ρ, is defined as

$$\rho = \frac{\mathrm{cov}(x,y)}{\sigma_x \sigma_y} = \frac{E\left[(x-\mu_x)(y-\mu_y)\right]}{\sigma_x \sigma_y} \tag{2}$$

where cov(x,y) denotes the covariance between variables x and y. σ_x, σ_y, μ_x, and μ_y are their standard deviations and means, respectively. The coefficient ρ ranges from -1 to 1, with larger absolute values indicating stronger linear correlation. Fig. 6 presents the correlation heatmap. For the maximum CE, T_a, V_c, F, T_2, and A_{max} show the correlations of ρ=-0.51, 0.38, 0.33, -0.3, and 0.22, respectively. Similar trends are observed for the maximum TEs of the three axes. In contrast, cycle time is dominated by the feedrate (ρ=-0.93), confirming its primary influence on machining efficiency. The results demonstrate that the interpolator parameters affect motion accuracy and efficiency in different ways and motivate the subsequent nonlinear surrogate and SHAP analyses.

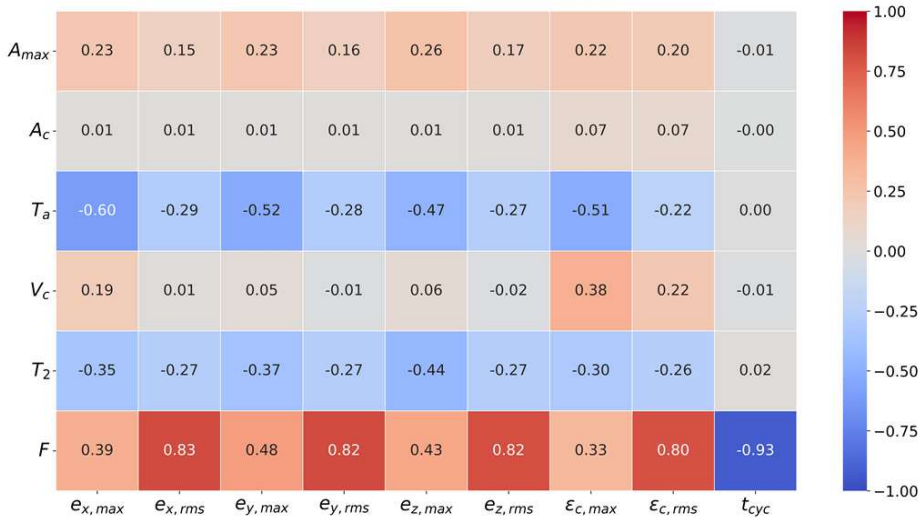


Fig. 6. Heatmap of Pearson correlation coefficients.

3.4. Surrogate modeling and accuracy validation

A back-propagation neural network (BPNN) is employed as the surrogate model to establish the nonlinear mapping between interpolator parameters and motion performance metrics, avoiding explicit analytical modeling of CNC interpolation process. The inputs consist of five interpolator parameters and feedrate, and the outputs contain nine performance metrics defined in Section 3.2. All inputs were

standardized using Z-score normalization before training, and the 729 DOE samples were divided by stratified sampling into a training set (80%) and a validation set (20%). In addition, 600 parameter combinations obtained from historical field-engineering data that did not coincide with the DOE grid were reserved as an independent off-grid testing set. These samples were excluded from both training and hyperparameter selection and used to evaluate the interpolation capability required by the subsequent continuous-action CrossQ optimization.

Table 5 Hyperparameter optimization of the BPNN model.

Hyperparameter	Minimum	Maximum	Step	Optimization
Learning rate	0.0001	0.01	auto	1.86e-3
Layer	1	10	1	3
Unit	32	512	16	192/96/448
Kernel regularizer	0.00001	0.1	auto	1.65e-4

Because BPNN performance is sensitive to network architecture and training hyperparameters, Hyperband was employed for efficient hyperparameter optimization [34]. The network was trained for 1500 epochs with a batch size of 64 using the Adam optimizer and an MSE loss. The resulting configuration is summarized in Table 5, giving a final network of six input neurons, three hidden layers, and nine output neurons. Prediction accuracy is evaluated using the mean absolute error (MAE), root mean square error (RMSE), maximum absolute error (MaxAE), normalized root mean square error (NRMSE), and coefficient of determination (R^2), defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_{p,i} - y_{a,i}\right|,\quad RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{p,i} - y_{a,i})^2},\quad MaxAE = \max_{i=1,\ldots,n}\left(\left|y_{p,i} - y_{a,i}\right|\right)$$
$$NRMSE = \frac{RMSE}{y_{a,\max} - y_{a,\min}},\quad R^2 = 1 - \frac{\sum_{i=1}^{n}(y_{p,i} - y_{a,i})^2}{\sum_{i=1}^{n}(y_{a,i} - y_m)^2} \tag{3}$$

where n is the number of samples. $y_{p,i}$ and $y_{a,i}$ denote the predicted and actual values of the ith sample,

respectively, and y_m is the mean of the actual outputs. $y_{a,max}$ and $y_{a,min}$ are their maximum and minimum values. NRMSE is normalized by the output range, allowing prediction accuracy to be compared across metrics of different physical units.

To assess robustness with respect to data partitioning, ten-fold cross-validation was performed using out-of-fold (OOF) predictions on the DOE dataset. As listed in Table 6, the OOF MAE and RMSE remained consistently low across the nine outputs, with MaxAE below 0.5301 μm for the error-related outputs and 0.0178 s for cycle time. The corresponding NRMSE ranged from 0.45% to 1.60%, with R^2 values of 99.46-99.96%. The consistency of the OOF results indicates that the predictive performance does not depend on a particular training-validation split and provides limited evidence of overfitting.

Table 6 Ten-fold out-of-fold prediction performance of the BPNN surrogate model.

Output	MAE	RMSE	MaxAE	NRMSE (%)	R^2 (%)
Tracking error (μm)					
$e_{x,max}$	0.0743	0.1003	0.3724	1.15	99.78
$e_{x,rms}$	0.0028	0.0037	0.0112	0.70	99.92
$e_{y,max}$	0.0973	0.1309	0.5301	1.05	99.76
$e_{y,rms}$	0.0034	0.0045	0.0210	0.66	99.92
$e_{z,max}$	0.0876	0.1152	0.4345	1.60	99.46
$e_{z,rms}$	0.0028	0.0037	0.0182	0.81	99.88
Contouring error (μm)					
$\varepsilon_{c,max}$	0.0866	0.1136	0.3914	1.24	99.69
$\varepsilon_{c,rms}$	0.0026	0.0034	0.0127	0.65	99.91
Cycle time (s)					
t_{cyc}	0.0055	0.0068	0.0178	0.45	99.96

The trained BPNN was further evaluated using the independent 600-sample off-grid testing set. As summarized in Table 7, the MAE and RMSE remained low across all nine outputs, with the NRMSE ranging from 0.06% to 2.91%, and R^2 from 91.98% to 99.22%. Although the maximum CE had the highest NRMSE, its R^2 was 96.14%, and its MaxAE did not exceed 1.3725 μm. These results

demonstrate that the surrogate retains satisfactory predictive accuracy for parameter combinations that are not located on the discrete DOE grid.

Table 7 Prediction accuracy of the BPNN surrogate model on the testing set.

Output	MAE	RMSE	MaxAE	NRMSE (%)	R^2 (%)
Tracking error (μm)					
$e_{x,max}$	0.1305	0.1688	0.6833	1.08	98.92
$e_{x,rms}$	0.0097	0.0112	0.0381	0.65	98.80
$e_{y,max}$	0.2341	0.2977	1.0596	1.27	97.99
$e_{y,rms}$	0.0138	0.0160	0.0472	0.73	98.47
$e_{z,max}$	0.2330	0.3092	1.0254	2.27	91.98
$e_{z,rms}$	0.0097	0.0113	0.0297	0.76	98.21
Contouring error (μm)					
$\varepsilon_{c,max}$	0.2293	0.2712	1.3725	2.91	96.14
$\varepsilon_{c,rms}$	0.0120	0.0148	0.0452	1.24	96.56
Cycle time (s)					
t_{cyc}	0.0181	0.0232	0.0552	0.06	99.22

To assess constraint reliability, the 600 off-grid samples were analyzed using the condition-specific predicted CE deviation ($d=\hat{\varepsilon_{c,max}}-\varepsilon_{tar}$) and prediction error ($\delta_c=\varepsilon_{c,exp}-\hat{\varepsilon_{c,max}}$). For 275 samples near the constraint boundary, $|d| \leq 0.3$μm, the MAE was 0.227 μm, with a positive mean bias of 0.115 μm and a 95th-percentile signed error of 0.509 μm. Among 240 samples predicted feasible within 0.3 μm below the target, 61 (25.4%) exceeded the experimental CE limit. These findings indicate that satisfactory overall prediction accuracy does not guarantee near-boundary feasibility, emphasizing the importance of constraint margins and physical verification of optimized settings.

Overall, the OOF and off-grid evaluations support surrogate-based continuous parameter search within the investigated domain, while the near-boundary feasibility misclassifications highlight the need for constraint-margin assessment and physical verification.

3.5. Explainable feature analysis based on SHAP

Although the BPNN surrogate model achieves high predictive accuracy, its nonlinear, multilayer

structure limits direct interpretation of the learned input-output relationships. SHapley Additive exPlanations (SHAP)[35] was therefore applied to quantify the contribution of each input feature to the surrogate predictions. Fig. 7 presents the SHAP beeswarm plots for the nine performance metrics, where the magnitude of the SHAP value reflects the strength of a feature's contribution and its sign indicates whether the feature increases or decreases the predicted response.

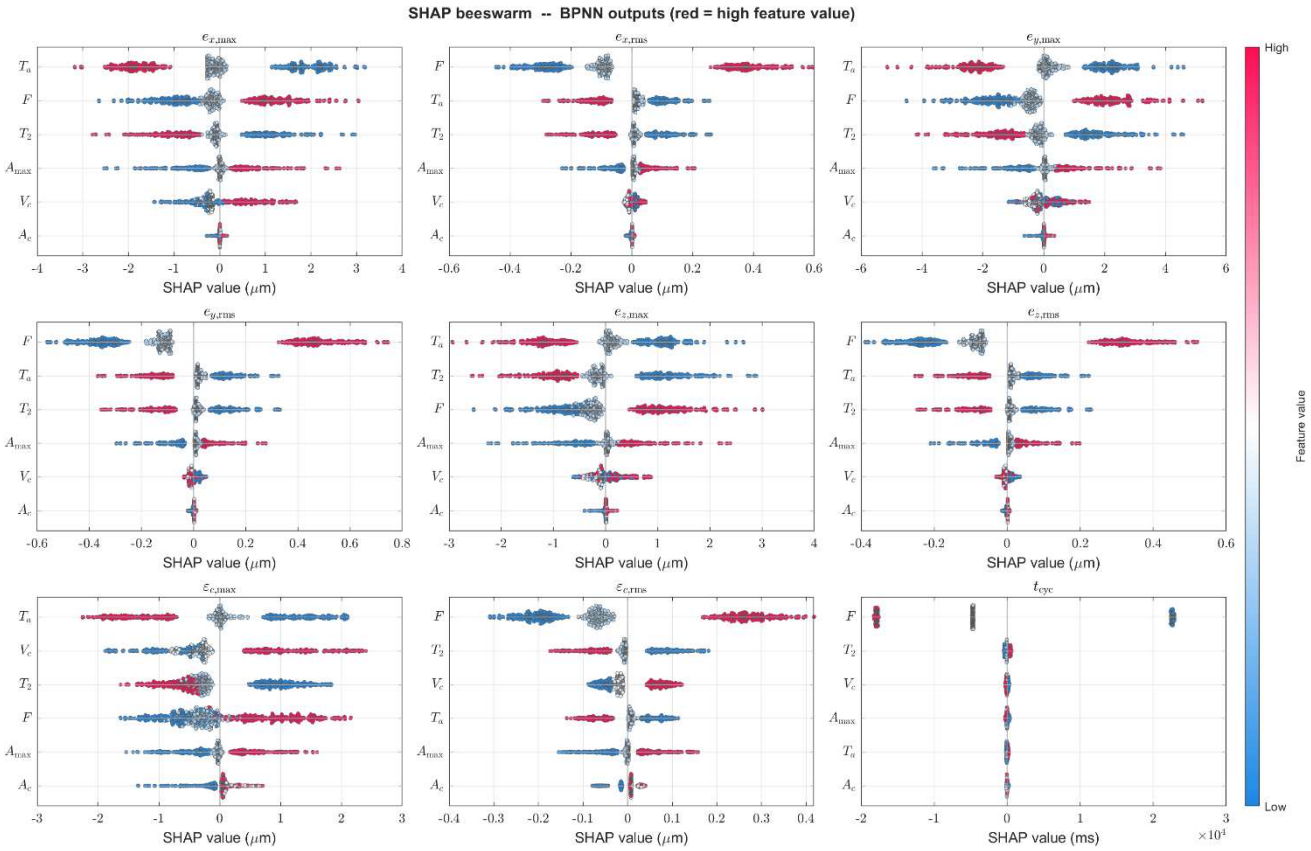


Fig. 7. SHAP beeswarm plots of feature contributions to BPNN predictions. (red: high feature value; blue: low feature value)

For the TE outputs, the ADAI time constant T_a, feedrate F, and ADBI time constant T_2 are generally dominant, although their relative importance varies among the three axes and between the RMS and maximum errors. T_a exerts a pronounced influence on the maximum TEs, indicating the importance of ACC/DEC dynamics in transient tracking performance. For the maximum CE, T_a again ranks highest, while the corner feedrate difference V_c and T_2 also contribute substantially. These trends

indicate that contouring accuracy is influenced by the combined effects of interpolation dynamics and feedrate-dependent motion behavior. This ranking is consistent with the Pearson correlations in Section 3.3 and with the FANUC tuning guidelines in Table 3, suggesting that the surrogate captures physically plausible parameter-performance relationships.

In contrast, feedrate dominates cycle time; its SHAP distribution is wider than those of the five AICC parameters, indicating that their direct effect on machining efficiency is limited. The SHAP analysis therefore reveals a clear functional distinction: feedrate primarily dominates machining efficiency, whereas the AICC parameters contribute more prominently to tracking and contouring accuracy. This interpretation provides a physical basis for the constraint-aware reward design in Section 4.2, in which cycle time forms the optimization objective while violations of contouring-error and jerk constraints are penalized.

4. BP-CrossQ framework for interpolator parameter optimization

As illustrated in Fig. 5, the BP-CrossQ framework performs continuous optimization of AICC interpolator parameters through interactions between CrossQ agent and BPNN surrogate environment. The following subsections describe the state and action spaces, reward function, and agent training procedure.

4.1. State and action definition

The state combines the current motion performance responses with the corresponding interpolator settings. At each time step t, the BPNN surrogate predicts the tracking error, contouring error, and cycle time for the current parameter vector. Because the feedrate is fixed for each machining mode, it

is treated as an environment condition rather than a state variable. The state is defined as

$$s_t = \left[e_t \quad \varepsilon_{c,t} \quad t_{cyc,t} \quad P_t \right] \tag{4}$$

where

$$\begin{aligned} e_t &= \left[e_{x,\max} \quad e_{x,rms} \quad e_{y,\max} \quad e_{y,rms} \quad e_{z,\max} \quad e_{z,rms} \right] \\ \varepsilon_{c,t} &= \left[\varepsilon_{c,\max} \quad \varepsilon_{c,rms} \right] \\ P_t &= \left[A_{\max,t} \quad A_{c,t} \quad T_{a,t} \quad V_{c,t} \quad T_{2,t} \right] \end{aligned} \tag{5}$$

and P_t denotes the AICC interpolator parameter vector at each time step t. The action is defined as the incremental adjustment

$$a_t = \Delta P_t = \left[\Delta A_{\max} \quad \Delta A_c \quad \Delta T_a \quad \Delta V_c \quad \Delta T_2 \right] \tag{6}$$

which modifies the parameter component of the state. The updated parameter vector is obtained as

$$P_{t+1} = \mathrm{clip}(P_t + a_t, P_{low}, P_{upr}) \tag{7}$$

where P_{upr} and P_{low} represent the upper and lower bounds of the AICC interpolator parameter vector, respectively, corresponding to the bound constraints g_{l} and g_{u} of Eq. (1). The clipping operation enforces these bounds during action execution, confining the search to the feasible domain. In contrast, the contouring error and jerk requirements are imposed as soft penalties through the reward function of Section 4.2. The BPNN surrogate then evaluates the updated parameter vector to generate the corresponding motion performance responses and the next state. Because the surrogate maps each parameter vector to performance metrics under fixed machining conditions, the sequential transitions represent iterative parameter refinement rather than the physical time evolution of the CNC system. This formulation enables the agent to learn a state-dependent adjustment policy that can be reused across different initial parameter settings.

4.2. Constraint-aware reward design

The reward function guides the CrossQ agent toward interpolator settings that balance machining efficiency, contouring accuracy, and motion smoothness. To express these competing terms on comparable dimensionless scales, a range-normalized reward function (RNRF) is formulated as

$$\begin{aligned} r_t &= r_{cyc} - w_c r_c - w_j r_j \\ r_{cyc} &= \frac{t_{\max} - t_{cyc}}{t_{\max} - t_{\min}}, \quad r_c = \max(0, \frac{\varepsilon_{c,\max} - \varepsilon_{tar}}{\varepsilon_{\max} - \varepsilon_{\min}}), \quad r_j = \max(0, \frac{J - J_{tar}}{J_{\max} - J_{\min}}) \end{aligned} \tag{8}$$

where $t_{\max}$, $t_{\min}$, $\varepsilon_{\max}$, $\varepsilon_{\min}$, $J_{\max}$ and $J_{\min}$ denote the condition-specific response bounds derived from the DOE dataset. The BPNN surrogate predicts the cycle time t_{cyc} and maximum contouring error $\varepsilon_{\mathrm{c,max}}$. The targets $\varepsilon_{\mathrm{tar}}$ and J_{tar} specify the allowable contouring accuracy and jerk, respectively. The nominal jerk is computed as $J=1000A_{max}/T_2$. The term r_{cyc} rewards shorter cycle time, whereas r_{c} and r_{j} impose soft penalties only when their respective constraint targets are exceeded. The fixed, condition-specific normalization ranges allow the penalty weights w_{c} and w_{j} to regulate the relative contributions of the two constraint penalties. Because r_{cyc} is bounded by unity within the normalization range, the penalty weights can be interpreted in terms of normalized constraint exceedances. Let δ_{c} and δ_{j} denote prescribed normalized exceedances of contouring error and jerk, respectively. For a penalty to offset the maximum possible cycle time reward gain, the weights should satisfy

$$w_q \delta_q \geq 1 \quad \Rightarrow \quad w_q \geq \frac{1}{\delta_q}, \quad q \in \{c, j\} \tag{9}$$

A normalized exceedance of 5% corresponds to a reference penalty weight of 20. This value is used as a reference scale rather than a theoretically optimal setting. Because contouring accuracy is assigned higher priority in precision machining, $w_{\mathrm{c}}=20$ is retained, while w_{j} is systematically varied in

Section 5.2 to evaluate the trade-off between machining efficiency and jerk violation rate.

4.3. CrossQ agent architecture and training

The CrossQ agent learns a state-dependent policy for sequential AICC interpolator parameter tuning through interactions with the BPNN surrogate environment. At each time step t, the stochastic actor π_φ selects an incremental parameter adjustment a_t from the current state s_t; the BPNN surrogate predicts the motion performance after enforcing parameter bounds of Eq. (7) to yield the reward r_t and next state s_{t+1}, and the transition (s_t, a_t, r_t, s_{t+1}, d_t) is stored in the replay buffer D for off-policy learning. By initializing training episodes from different admissible parameter settings, the agent learns sequential adjustment strategies across the investigated parameter space rather than optimizing a single fixed configuration.

CrossQ employs twin critics, $Q_{\theta 1}$ and $Q_{\theta 2}$, to mitigate value overestimation and removes the target networks used in conventional actor-critic methods. In this implementation, batch renormalization (BRN) replaces the batch normalization used in the original CrossQ, while the current and next state-action pairs are propagated through the critics to maintain consistent normalization statistics. One critic update is performed per environment interaction (UTD=1), while the actor and entropy temperature are updated according to the policy-update delay d_π. The minimum twin Q-value provides a conservative value estimate, whereas entropy regularization promotes exploration. Through iterative surrogate interactions, the agent learns a parameter adjustment policy that improves cumulative reward within the parameter bounds while penalizing contouring error and jerk violations.

Once trained, the policy is frozen and can be reused to refine different initial interpolator settings

under the same machining condition without additional reinforcement learning updates. During deployment, the policy sequentially generates parameter adjustments within the virtual environment, and each candidate is evaluated against the prescribed contouring error and jerk targets. Among the feasible candidates encountered, the setting with the shortest predicted cycle time is retained for subsequent physical-machine validation. This separation of offline learning and reusable policy deployment enables state-dependent parameter refinement while avoiding repeated on-machine exploration. Algorithm 1 summarizes the complete BP-CrossQ training and deployment procedure.

Algorithm 1 Reusable BP-CrossQ policy training and deployment

Input: Trained BPNN surrogate f_{BP}; parameter bounds P_{low}, P_{upr}; feedrate F; reward parameters in Eq. (8); CrossQ training and deployment settings; initial parameter vector P_{ini}.

Output: Trained reusable policy π^*_ϕ; Optimized parameter vector $P^*(P_{ini}, F)$ for each deployment.

Phase I: Offline CrossQ policy training

Initialize actor π_ϕ, twin critics $Q_{\theta 1}$, $Q_{\theta 2}$ with BatchRenorm (BRN) layers, replay buffer D, temperature parameter α, critic-update counter $u \leftarrow 0$ and global step counter $k \leftarrow 0$

for *episode = 1 to* N_{ep} **do**

Initialize an admissible interpolator vector $P_0 \in [P_{low}, P_{upr}]$, evaluate the BPNN surrogate $f_{BP}(P_0, F)$ to construct the initial state $s_0 \leftarrow [e_0\ \varepsilon_{c,0}\ t_{cyc,0}\ P_0]$

for $t = 0$ to *T-1* **do**

if $k < T_{ini}$ **then** $a_t \sim \text{Uniform}(A)$ **else** $a_t \sim \pi_\phi(\cdot|s_t)$ **end if**

Update the interpolator parameters: $P_{t+1} \leftarrow \text{clip}(P_t + a_t, P_{low}, P_{upr})$

Predict motion performance: $[e_{t+1}\ \varepsilon_{c,t+1}\ t_{cyc,t+1}] \leftarrow f_{BP}(P_{t+1}, F)$

Construct the next state: $s_{t+1} \leftarrow [e_{t+1}\ \varepsilon_{c,t+1}\ t_{cyc,t+1}\ P_{t+1}]$

Calculate the reward r_t by Eq. (8)

Set the termination flag $d_t \leftarrow 0$

Store the transition $(s_t, a_t, r_t, s_{t+1}, d_t)$ in D

if $k \geq T_{ini}$ and $|D| \geq M$ **then**

Sample a mini-batch $\{(s_i, a_i, r_i, s'_i, d_i)\}_{i=1}^{M}$ from D

Sample next actions $\tilde{a}'_i \sim \pi_\phi(\cdot|s'_i)$ and record $\log\pi_\phi(\tilde{a}'_i \mid s'_i)$

Forward $[(s_i, a_i); (s'_i, \tilde{a}'_i)]$ through $Q_{\theta 1}$, $Q_{\theta 2}$ in a single pass

Compute the TD targets: $y_i = r_i + \gamma(1-d_i)sg\left[\min_{j=1,2} Q_{\theta_j}(s'_i, \tilde{a}'_i) - \alpha \log \pi_\phi(\tilde{a}'_i \mid s'_i)\right]$

Update both critics by minimizing the Bellman errors

$u \leftarrow u + 1$

if u mod $d_\pi = 0$ **then** Update the actor π_ϕ and entropy temperature α following [36] **end if**

end if

$s_t \leftarrow s_{t+1}$; $k \leftarrow k+1$

end for

end for ▷$t = T\text{-}1$ *is a time-limit truncation:* s_T *is bootstrapped, not terminal*

Store the trained policy π^*_ϕ

Phase II: Reusable sequential policy deployment

Freeze the trained policy π^*_ϕ and BPNN surrogate f_{BP}

Initialize $P_0 \leftarrow P_{ini}$, $[e_0\ \varepsilon_{c,0}\ t_{cyc,0}] \leftarrow f_{BP}(P_0, F)$, $s_0 \leftarrow [e_0\ \varepsilon_{c,0}\ t_{cyc,0}\ P_0]$

Set $P_{best} \leftarrow \emptyset$ and $t_{best} \leftarrow +\infty$.

for $t = 0$ to T_{dep} **do**

Compute $J_t = 1000A_{max,t}/T_{2,t}$

if $\varepsilon_{c,max,t} \le \varepsilon_{tar}$ and $J_t \le J_{tar}$ and $t_{cyc,t} < t_{best}$ **then**

$P_{best} \leftarrow P_t$; $t_{best} \leftarrow t_{cyc,t}$

end if

if $t < T_{dep}$ **then**

$a_t \leftarrow \text{deterministic}(\pi^*_\phi \mid s_t)$

$P_{t+1} \leftarrow \text{clip}(P_t + a_t, P_{low}, P_{upr})$

$[e_{t+1}\ \varepsilon_{c,t+1}\ t_{cyc,t+1}] \leftarrow f_{BP}(P_{t+1}, F)$; $s_{t+1} \leftarrow [e_{t+1}\ \varepsilon_{c,t+1}\ t_{cyc,t+1}\ P_{t+1}]$

end if

end for

if $P_{best} = \emptyset$ **then** Report no surrogate-feasible candidate, **Return** π^*_ϕ and $P^*(P_{ini}, F) \leftarrow \emptyset$

else Return π^*_ϕ and $P^*(P_{ini}, F) \leftarrow P_{best}$

end if

sg[·] denotes the stop-gradient operator.

5. Results and experimental validation

This section evaluates the proposed BP-CrossQ framework through analyses within the virtual environment and machining experiments on the physical CNC machine tool.

5.1. Experimental setup

All experiments and standard tests were carried out on a three-axis vertical machining center equipped with a FANUC 0i-MF Plus controller, shown in Fig. 8. The command and actual positions

of the three feed axes were acquired in real time through the FANUC FOCAS interface at a sampling rate of 1 kHz. The position-loop bandwidth of each feed axis was tuned to 40 Hz, ensuring servo dynamics matching across the three axes. The BPNN and RL models were implemented using PyTorch 2.3.0 on an industrial PC equipped with an Intel Core i9-9820X CPU and an NVIDIA GeForce RTX 4090 GPU. The validated BPNN surrogate described in Section 3.4 was used as the virtual environment for CrossQ training. The KAKINO toolpath, depicted in Fig. 2(a), was employed as the benchmark trajectory for AICC interpolator parameter optimization and performance evaluation.



Fig. 8. Experimental platform.

5.2. Sensitivity analysis of reward function

To investigate the influence of the penalty weights w_c and w_j in Eq. (8), a sensitivity analysis was conducted to quantify the trade-off among machining time, contouring accuracy, and nominal jerk constraint satisfaction. The contouring error weight was fixed at w_c=20, ensuring that the contouring accuracy term consistently imposes a conservative policy bias, while the jerk-penalty weight w_j was varied over a logarithmic grid, $w_j \in \{0.2, 0.6325, 2, 6.325, 20\}$. The lower bound represents a relatively aggressive policy that places limited emphasis on soft jerk constraints to reduce cycle time; in contrast,

the upper bound imposes a stronger penalty on jerk violations and promotes smoother motion. Ten independent random seeds were evaluated for both semi-finish and finishing conditions. Table 8 summarizes the reward function parameters and normalization bounds derived from the DOE dataset.

Table 8 The reward function parameters and normalization bounds for machining conditions.

Condition	F	ε_{tar}	J_{tar}	t_{max}	t_{min}	ε_{max}	ε_{min}	J_{max}	J_{min}
Semi-finish	1500	4.1	62500	30.766	29.259	11.061	1.902	18.75×10^4	1.25×10^4
Finishing	750	2.3	27875	57.355	56.015	9.197	1.880		

Fig. 9(a) and (b) present the training reward curves. All five weight settings exhibit rapid reward improvement during the early training stage, followed by stable convergence. This demonstrates that BP-CrossQ learns stably across a broad range of w_j values. Although the converged reward levels are similar, the resulting policies differ markedly in constraint behavior. Fig. 9(c)-(f) illustrates the distributions of maximum CE and CT, and Tables 9 and 10 summarize the corresponding violation rates and motion performance metrics. In semi-finish, increasing w_j from 0.2 to 20 reduces the mean jerk violation rate from 72.57% to 0.98%, while the mean CE violation rate decreases from 2.46% to 1.36%. Over the same range, the mean CT increases only 0.122 s (0.41%), and the maximum CE remains within 3.736-3.860 μm. The same trend holds in finishing: the jerk violation rate falls from 95.58% to 8.59% at a CT cost of 0.16%, whereas the maximum CE remains within 2.266-2.289 μm. These results indicate that w_j primarily regulates jerk constraint satisfaction, with comparatively small effects on contouring accuracy and cycle time.

A different trend is observed for the CE violation rate in finishing, which remains approximately 18-20% regardless of w_j, even though the mean maximum CE stays below the prescribed target. This behavior is associated with the tighter CE margin in finishing. When normalized by the condition-

specific CE range (ε_{max}-ε_{min}), the margin between ε_{tar} and ε_{min} is only 5.74%, compared with 24% in semi-finish. The learned policies operate closer to the CE constraint boundary, making the binary violation ratio more sensitive to stochastic variations across random seeds. The rollout results further support the role of w_j in constraint regulation. The jerk violation rate falls to 0% for $w_j \geq 2$ in both machining conditions, whereas the CE rollout violation rate remains 0% for all w_j. This contrast suggests that the nonzero training violation rates mainly reflect exploratory behavior during policy learning rather than persistent violations of the final policy.

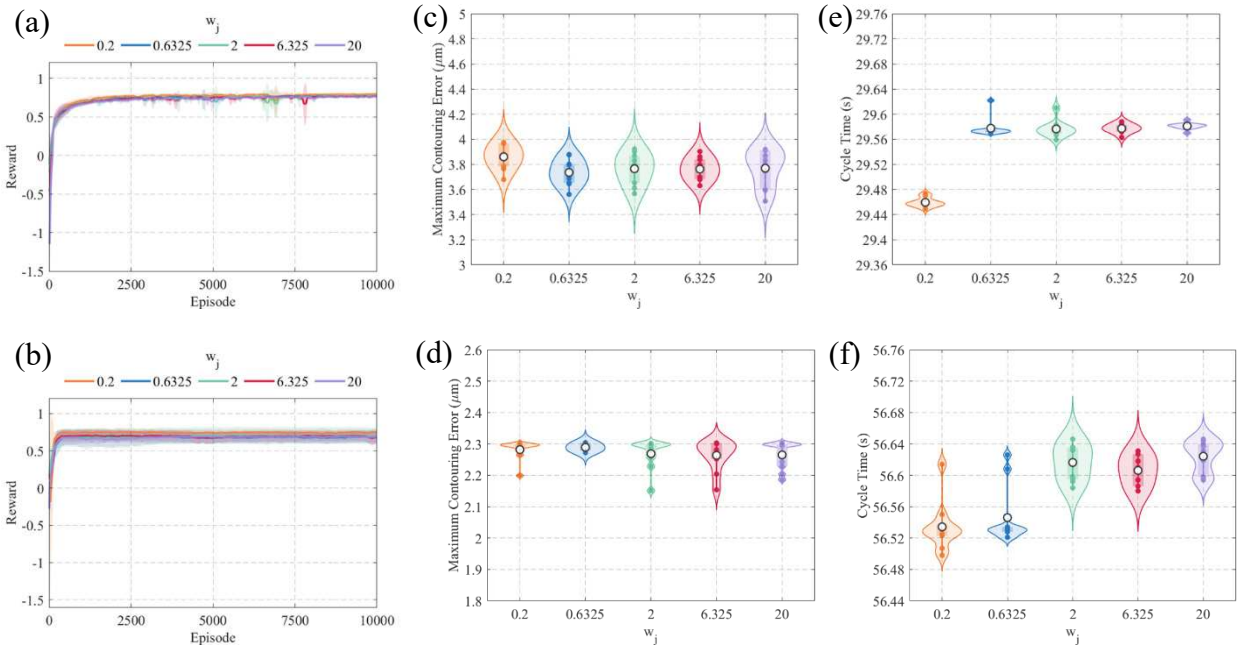


Fig. 9. Sensitivity analysis of the jerk penalty weight w_j (w_c=20, 10 random seeds) in semi-finish (top row) and finishing (bottom row): (a)(b) training reward curves, (c)(d) violin plots of maximum CE, and (e)(f) violin plots of CT.

Overall, increasing w_j suppresses aggressive motion and substantially improves jerk constraint satisfaction with only a marginal cycle time penalty. Because this penalty remains below 0.5% in both machining conditions without appreciable degradation in contouring accuracy, w_c=w_j=20 is adopted in the subsequent experiments as a balanced setting among machining efficiency, contouring accuracy, and motion smoothness.

Table 9 Contouring error and jerk violation rates during training and policy rollout evaluation (%).

Condition	Phase	Constraint	w_j				
			0.2	0.6325	2	6.325	20
Semi-finish	Training	Jerk	72.57±34.51	40.24±18.80	11.51±5.39	2.76±1.42	0.98±0.32
		CE	2.46±0.56	1.61±0.76	1.77±1.15	1.71±0.97	1.36±0.75
	Rollout	Jerk	100	60	0	0	0
		CE	0	0	0	0	0
Finishing	Training	Jerk	95.58±3.12	78.13±8.90	62.45±23.12	21.49±8.86	8.59±2.58
		CE	18.03±8.93	18.66±11.16	18.64±13.09	20.17±14.69	20.00±17.99
	Rollout	Jerk	100	100	0	0	0
		CE	0	0	0	0	0

Note: Training values are the mean ± SD of episode violation rate across 10 random seeds; rollout values are the percentage of final policies violating each constraint.

Table 10 Effect of jerk reward weight w_j on motion performance metrics

Condition	Metric	w_j				
		0.2	0.6325	2	6.325	20
Semi-finish	Max. CE (μm)	3.860±0.098	3.736±0.104	3.766±0.122	3.763±0.089	3.769±0.152
	Jerk ($\times10^4$mm/s^3)	10.572±0.621	6.108±0.434	6.065±0.193	5.750±0.486	5.706±0.412
	CT (s)	29.459±0.007	29.578±0.016	29.576±0.014	29.577±0.007	29.581±0.005
Finishing	Max. CE (μm)	2.284±0.033	2.289±0.013	2.269±0.048	2.264±0.050	2.266±0.046
	Jerk ($\times10^4$mm/s^3)	4.457±1.696	3.234±0.158	2.605±0.073	2.717±0.045	2.617±0.039
	CT (s)	56.534±0.032	56.546±0.038	56.616±0.023	56.606±0.020	56.624±0.021

Note: Values are the mean ± SD over 10 random seeds with w_c=20.

5.3. Ablation study of reward function

To quantify the contribution of the penalty terms in the RNRF of Eq. (8), an ablation study was conducted for two machining conditions with w_c=w_j=20. Four reward variants were evaluated: R_o, the complete RNRF; R_j, without the jerk penalty; R_c, without the CE penalty; and R_{cj}, without either penalty. Each variant was trained using 10 independent random seeds with identical network architecture, BPNN surrogate, and CrossQ hyperparameters. Fig. 10(a) shows that all reward variants improve rapidly during early training, and subsequently reach stable behavior. R_{cj} attains the highest reward

because removing both penalties allows the agent to prioritize cycle time reduction without penalty for CE or jerk violations. This result indicates that reward magnitude alone does not reflect policy quality and must be interpreted together with constraint satisfaction and motion performance. Fig. 10(b), (c) and Table 11 reveal the distinct roles of the two penalty terms. Removing the jerk penalty increased the jerk violation rate from 0.98% to 89.04% in semi-finish and from 8.59% to 92.08% in finishing. Similarly, removing the CE penalty increased the CE violation rate from 1.36% to 94.05% in semi-finish and from 20% to 99.69% in finishing. Removing both penalties resulted in violation rates approaching 100% in both machining conditions. These results demonstrate that the two penalties regulate different constraints and are complementary rather than redundant.

The motion performance metrics in Fig. 10(d)-(f) and Table 11 further support this interpretation. In semi-finish, removing both penalties reduced CT from 29.581 to 29.267 s (1.1%), but increased the maximum CE from 3.769 to 10.954 μm, exceeding the CE target of 4.1 μm, and the jerk from 5.706×10^4 to 18.75×10^4 mm/s^3, exceeding the jerk target of 6.25×10^4 mm/s^3. In finishing, R_{cj} similarly reduced CT from 56.624 to 56.015 s (1.1%), while increasing the maximum CE from 2.266 to 9.121 μm and the jerk from 2.617×10^4 to 14.956×10^4 mm/s^3. Thus, the modest CT improvement from removing the penalties is accompanied by severe degradation in accuracy and motion smoothness. Compared with R_{cj}, the complete reward R_o sacrifices roughly 1% of CT while reducing the maximum CE by 65% and the jerk by 69% in semi-finish. The ablation results demonstrate that the CE penalty is essential for maintaining contouring accuracy, and that the jerk penalty suppresses aggressive motion. The complete RNRF (R_o) therefore provides the most appropriate balance among machining efficiency,

contouring accuracy, and motion smoothness under both machining conditions.

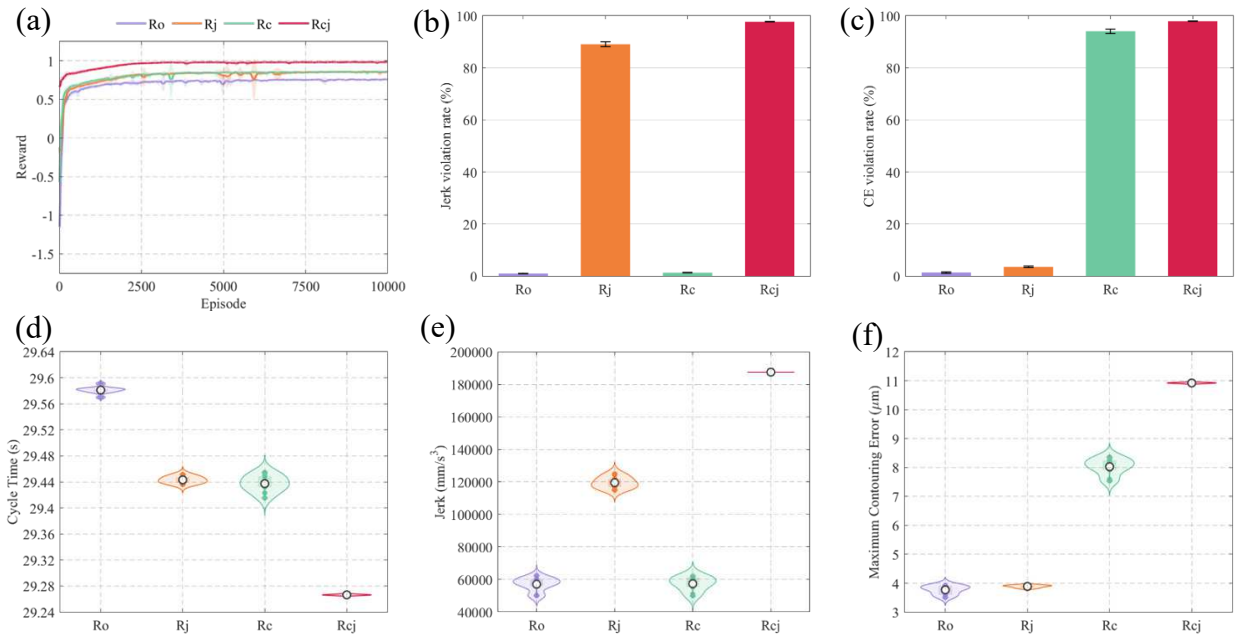


Fig. 10. Ablation study of the reward function with 10 random seeds in semi-finish (a) training reward curves, (b) jerk violation rate, (c) CE violation rate, (d) cycle time, (e) jerk, (f) maximum CE.

Table 11 Performance of the four reward variants over 10 random seeds.

Condition	Metric	R_o	R_j	R_c	R_{cj}
Semi-finish	Jerk violation (%)	0.98	89.04	1.27	97.67
	CE violation (%)	1.36	3.54	94.05	97.90
	Max. CE (μm)	3.769	3.920(+4.0%)	8.055(+114%)	10.954(+191%)
	Jerk ($\times 10^4$mm/s^3)	5.706	11.926(+109%)	5.809(+1.8%)	18.75(+229%)
	CT (s)	29.581	29.442(-0.5%)	29.440(-0.5%)	29.267(-1.1%)
Finishing	Jerk violation (%)	8.59	92.08	6.57	99.62
	CE violation (%)	20.00	21.16	99.69	99.92
	Max. CE(μm)	2.266	2.273(+0.3%)	8.826(+290%)	9.121(+303%)
	Jerk ($\times 10^4$mm/s^3)	2.617	8.40(+221%)	2.69(+2.8%)	14.96(+471%)
	CT (s)	56.624	56.498(-0.2%)	56.133(-0.9%)	56.015(-1.1%)

Note: Violation rates are training values averaged over 10 seeds; percentages in parentheses denote changes relative to R_o.

5.4. Comparative evaluation with baseline optimization methods

To evaluate the optimization performance of BP-CrossQ, both simulations and experiments were conducted using the KAKINO benchmark. For a fair comparison, two surrogate-assisted baselines,

BP-GA and BP-TD3, were developed by integrating the same BPNN surrogate with Genetic Algorithm (GA) [37] and Twin Delayed Deep Deterministic Policy Gradient (TD3) [38], respectively. The three methods shared the parameter bounds, constraints, and RNRF formulation (w_c=w_j=20), while retaining method-specific optimization procedures. GA was implemented using the pymoo library with the parameters as follows. A population of 300 individuals is evolved for 50000 generations, with 20 offspring produced per generation. Offspring are generated by simulated binary crossover (SBX) with a crossover probability of 0.9 and a distribution index η_c=10, followed by polynomial mutation applied with a probability of 0.2 per decision variable and a distribution index η_m=10. The training and deployment settings for CrossQ and TD3 are given in Table 12. The target update rate, policy noise, noise clip, and exploration noise for TD3 are 0.005, 0.2, 0.5, and 0.1, respectively. Each training run was executed for 1 million environment steps, corresponding to 10,000 episodes with a maximum of 100 steps per episode.

Table 12 Training and deployment settings for CrossQ and TD3.

Parameter	Symbol	CrossQ	TD3
Learning rate	$\eta_Q/\eta_\pi/\eta_\alpha$	0.001/0.001/0.001	0.001/0.001/–
Discount factor	γ	0.99	0.99
Policy delay	d_π	3	2
Batch size	M	256	256
Training frequency	v_{tra}	1	1
Gradient steps	N_{gra}	1	1
Warm-up steps	T_{ini}	100	100
Replay buffer capacity	N_D	1×10^6	1×10^6
Actor network structure	π	[512 512]	[400 300]
Critic network structure	Q_θ	[2048 2048]	[400 300]
Episode horizon	T	100	100
Episode number	N_{ep}	10000	10000
Rollout horizon	T_{dep}	100000	100000

Note: Each rollout step includes policy inference, parameter updating, and surrogate evaluation.

Table 13 compares manual tuning (MT), BP-GA, BP-TD3, and BP-CrossQ under two machining conditions. In semi-finish, BP-CrossQ satisfied both targets, reducing the experimental maximum CE from 4.073 to 3.888 μm and the jerk to 60313 mm/s³, while achieving the shortest experimental CT of 29.539 s. In contrast, BP-GA and BP-TD3 satisfied the jerk target but yielded experimental maximum CEs of 4.231 and 4.125μm, respectively, exceeding ε_{tar}=4.1 μm. In finishing, BP-CrossQ again satisfied both targets with a maximum CE of 1.902 μm, a jerk of 25871 mm/s^3, and the shortest CT of 56.612 s. BP-GA slightly exceeded the jerk target, and BP-TD3 marginally exceeded the CE target. BP-CrossQ was therefore the only surrogate-assisted method that satisfied contouring accuracy and jerk targets under both machining conditions.

The agreement between surrogate predictions and physical experiments also supports the validity of the virtual optimization results. For BP-CrossQ, the CE and CT discrepancies were approximately 0.143 μm (3.82%) and 39 ms (0.13%) in semi-finish, and 0.301 μm (13.66%) and 7 ms (0.01%) in finishing, respectively. Relative to the CE targets, the predicted and experimental constraint margins were 0.355 and 0.212 μm in semi-finishing, and 0.097 and 0.398 μm in finishing, respectively, confirming positive margins for the BP-CrossQ settings. In contrast, BP-GA and BP-TD3 were predicted to satisfy the CE constraint in semi-finishing but exceeded the target experimentally, highlighting the importance of physical verification. The improvement in contouring accuracy was further assessed by Welch's t-tests using ten repeated runs, which yielded $t(9.0)=19.843$ and $p=9.602\times10^{-9}$ with a 95% confidence interval of [0.150, 0.189] μm in semi-finish, and $t(16.3)=7.237$ and $p=1.767\times10^{-6}$ with a 95% confidence interval of [0.042, 0.077] μm in finishing. Both results

confirm a statistically significant reduction in contouring error relative to manual tuning.

Table 13 Performance comparison of the KAKINO toolpath among the evaluated methods.

Condition	Algorithm	A_{max}	A_c	T_a	V_c	T_2	Jerk	$\varepsilon_{c,max}$	t_{cyc}
								Sim/Exp	Sim/Exp
Semi-finish	MT	1500	1000	24	320	24	62500	4.349/4.073	29.589/29.589
	BP-GA	1335	1565	57	630	22	60682	3.997/4.231	29.576/29.569
	BP-TD3	1052	1543	60	640	17	61882	3.932/4.125	29.570/29.579
	BP-CrossQ	965	1078	64	640	16	60313	3.745/3.888	29.578/29.539
Finishing	MT	892	240	24	120	32	27875	2.304/1.988	56.632/56.616
	BP-GA	1508	1896	63	362	54	27926	2.289/2.286	56.654/56.680
	BP-TD3	1530	2000	64	393	55	27818	2.306/2.307	56.636/56.656
	BP-CrossQ	802	574	16	89	31	25871	2.203/1.902	56.619/56.612

Note: Sim/Exp values represent the selected benchmark runs (seed 2); the statistical tests are based on ten repeated physical runs per condition.

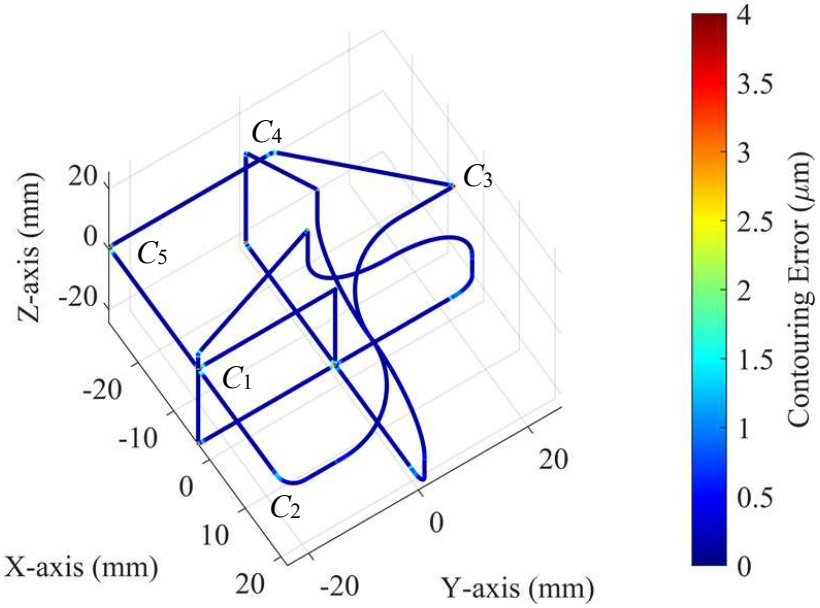


Fig. 11. Contouring error distribution along KAKINO toolpath using BP-CrossQ under semi-finish.

Fig. 11 illustrates the spatial distribution of contouring error for BP-CrossQ in semi-finish. The largest local deviations occur near the geometric transition regions C_1-C_5, where abrupt changes in curvature and feed direction increase interpolation difficulty. Nevertheless, the interpolator parameters optimized by BP-CrossQ maintain the contouring error within the prescribed accuracy target along the entire KAKINO trajectory.

5.5. Experimental validation on the physical machine tool

To validate cross-trajectory transferability of the optimized interpolator parameters, machining experiments were conducted on the same three-axis machining center using the FANUC test mold, shown in Fig. 2(b). A two-flute ball-end mill (4 mm in diameter and 90 mm in overall length) was used to machine a 6061-T6 aluminum alloy workpiece at a spindle speed of 8000 rpm. The interpolator settings obtained from MT, BP-GA, BP-TD3, and BP-CrossQ on the KAKINO benchmark were directly transferred to the test mold under identical cutting conditions within each machining mode, with the MT default settings retained as the reference.

Table 14 Performance comparison of the optimization methods on the FANUC test mold.

Condition	Method	$\varepsilon_{c,max}$(μm)	CE reduction	t_{cyc}(min:ss.sss)	CT reduction
Semi-finish	MT	7.908	–	28:23.085	–
	BP-GA	5.631	28.79%	28:04.420	1.10%
	BP-TD3	5.643	28.64%	28:04.900	1.07%
	BP-CrossQ	5.459	30.97%	28:02.600	1.20%
Finishing	MT	4.281	–	56:20.920	–
	BP-GA	4.039	5.65%	55:33.890	1.39%
	BP-TD3	4.022	6.05%	55:33.210	1.41%
	BP-CrossQ	3.741	12.61%	55:32.530	1.43%

Note: Percentages are relative to MT

The results are summarized in Table 14. BP-CrossQ achieved the largest CE reduction and the shortest CT in both machining conditions, decreasing the maximum CE by 30.97% in semi-finish and 12.61% in finishing relative to MT, with corresponding CT reductions of 1.20% and 1.43%. BP-GA and BP-TD3 also improved both metrics, but by smaller margins. The cross-trajectory experiments demonstrate that the interpolator settings optimized on the KAKINO benchmark retain their relative advantages in contouring accuracy and cycle time when applied to a more complex FANUC test mold

on the same machine tool. However, the results establish performance transfer rather than guaranteed constraint satisfaction across different toolpath geometries.

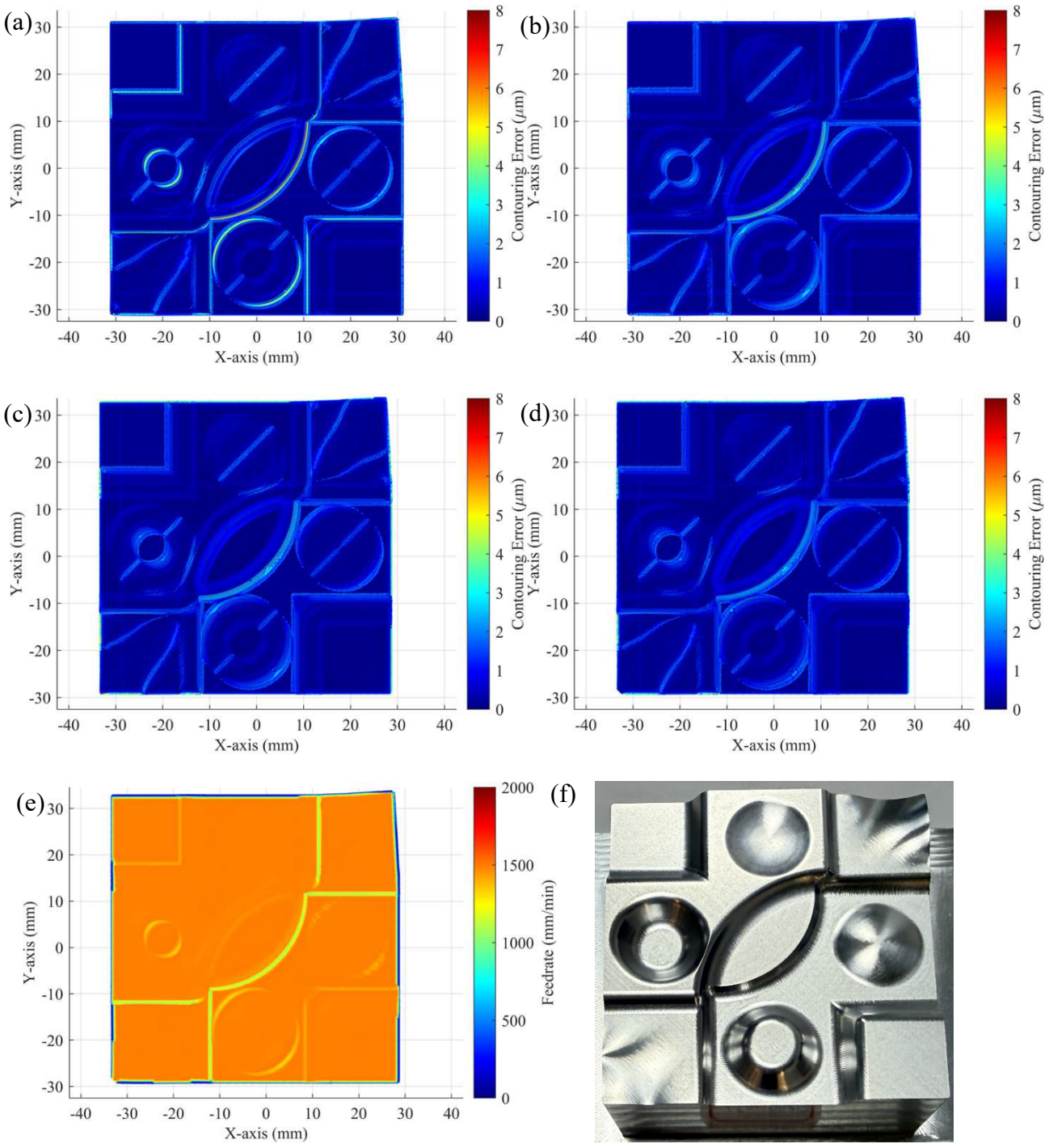


Fig. 12. Experimental validation on the FANUC test mold under semi-finish. (a)-(d) contouring error maps for MT, BP-GA, BP-TD3, and BP-CrossQ; (e)feedrate map for BP-CrossQ; (f)machined workpiece using BP-CrossQ.

Fig. 12 compares the contouring error distributions for the semi-finish experiments. BP-CrossQ exhibits a more uniform error distribution with fewer localized high-error regions than the other

methods. The maximum CE occurs at different geometric features for the four parameter settings, indicating that contouring error remains strongly dependent on local path geometry. For BP-CrossQ, residual errors concentrate mainly near path transition boundaries, where the feedrate map in Fig. 12(e) presents a correspondingly reduced feedrate. The cutting result is shown in Fig. 12(f).

Beyond machining performance, Table 15 compares the computational and commissioning costs of the evaluated methods. BP-CrossQ requires 31 min 2 s for offline agent training and only 255 ms for deployment. Including the shared data acquisition and surrogate development stages, the total commissioning time is 11 h 5 min 38 s, compared with 24 h for conventional manual tuning. Both commissioning times are calculated by summing the durations of their respective workflow stages. The data acquisition, surrogate development, and agent training are one-time setup costs, whereas deployment incurs minimal recurring computational cost when reusing the trained policy. The 600-sample off-grid testing set consists of historical field engineering data and is therefore excluded from the present commissioning cost. However, deployment on a different machine may require additional data acquisition and surrogate retraining, introducing new setup costs.

Table 15 Comparison of computational cost and commissioning time among the evaluated methods.

Commissioning stage	MT	BP-GA	BP-TD3	BP-CrossQ
Physical commissioning/ data acquisition	KAKINO test (8h) Dynamic analysis (9h) Parameter tuning (7h)	8h 13min 12s		
Surrogate training /hyperparameter tuning	N/A	1min 32s/2h 19 min 52s		
Optimizer/agent training		5min 58s	17min 4s	31min 2s
Solution generation /deployment			324ms	255ms
Total time	24h	10h40min34s	10h51min40s	11h5min38s

Note: Repeated synchronized wall-clock measurements verified complete 100,000-step RL

deployment times; data acquisition and surrogate development are shared one-time costs.

To assess the initial-state robustness and reusability of the learned policy, 50 randomly generated admissible parameter vectors were evaluated under each machining condition. The trained TD3 and CrossQ policies were kept fixed during evaluation. As shown in Table 16, BP-CrossQ and BP-TD3 achieved 100% surrogate-predicted feasibility in semi-finish, compared with 54% for BP-GA. Under the tighter finishing condition, BP-CrossQ maintained a 100% success rate, outperforming BP-GA (90%) and BP-TD3 (54%). Notably, the same condition-specific CrossQ policy successfully refined all 50 initial parameter settings into surrogate-feasible solutions without retraining. Although conventional numerical optimization remains a viable alternative for this low-dimensional parameter tuning problem, these results highlight BP-CrossQ advantages in solution consistency and reusable policy-based parameter refinement within the surrogate environment.

Table 16 Initial-state robustness evaluation of the optimization methods.

Condition	Algorithm	$\varepsilon_{c,max}$ (μm)	t_{cyc} (s)	Jerk (mm/s³)	Success rate(%)	First solution (steps)
Semi-finish	BP-GA	3.924 ± 0.076	29.575 ± 0.008	59403 ± 3419	54	3486 ± 686
	BP-TD3	3.781 ± 0.199	29.730 ± 0.037	57057 ± 2950	100	7.5 ± 6.7
	BP-CrossQ	3.632 ± 0.074	29.573 ± 0.010	53644 ± 3700	100	25.4 ± 24.0
Finishing	BP-GA	2.266 ± 0.062	56.630 ± 0.022	27357 ± 558	90	3155 ± 892
	BP-TD3	2.197 ± 0.120	56.679 ± 0.043	24708 ± 3168	54	16.2 ± 25.6
	BP-CrossQ	2.194 ± 0.054	56.612 ± 0.020	25806 ± 802	100	79 ± 1.1

Note: Mean ± SD over 50 initial states; success denotes surrogate-predicted feasibility. First-solution counts represent GA evaluations or RL steps, using successful runs only.

6. Conclusions and future work

This study developed an AI-enabled virtual commissioning framework that integrates a BPNN surrogate with CrossQ reinforcement learning for CNC interpolator parameter optimization. Trained on full-factorial DOE data and evaluated using ten-fold out-of-fold validation and 600 independent

off-grid samples, the surrogate supports continuous parameter search within the investigated domain. Correlation and SHAP analyses provided insight into the learned parameter-performance relationships. Sensitivity and ablation studies revealed the complementary roles of contouring error and jerk penalties, while increasing the jerk-penalty weight improved constraint satisfaction with less than a 0.5% increase in cycle time.

On the KAKINO benchmark, BP-CrossQ was the only surrogate-assisted method that satisfied both contouring error and jerk targets under semi-finish and finishing conditions. Its fixed, condition-specific policy attained 100% surrogate-predicted feasibility across 50 initial settings per condition without retraining, supporting consistent, reusable sequential refinement within the virtual environment. Transferring the optimized settings to a FANUC test mold reduced the maximum contouring error by 30.97% and 12.61% and the cycle time by 1.20% and 1.43%, respectively, relative to manual tuning. The total commissioning time was also reduced from 24 h to approximately 11 h, with data acquisition, surrogate development, and policy training representing one-time setup costs.

Cross-trajectory performance transfer was validated only on the same machine and controller, not guaranteeing constraint satisfaction on new geometries or generalization across machines. Future work will investigate more complex trajectories, such as random NURBS curves, incorporate uncertainty-aware feasibility assessment into surrogate-based RL, and extend the framework to five-axis systems and alternative CNC controllers.

7. Statements and declarations

Funding. This research was supported by the National Science and Technology Council (NSTC) of

Taiwan, R.O.C., under Grant Nos. NSTC 112-2221-E-150-017, 114-2221-E-150-051 and 114-2218-E-002-007.

Competing interests. The authors have no relevant financial or non-financial interests to disclose.

Author contributions. All authors contributed to the study conception and design. You-Cheng Chen and Yong-Zhong Li performed material preparation, data collection, and analysis. Ming-Tsung Lin prepared the original manuscript draft and conducted validation and supervision. All authors reviewed and commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Acknowledgements. The authors would like to thank Victor Taichung Machinery Works Co., LTD. for access to equipment required as part of this work.

Data availability. The datasets generated and analyzed during the current study are available from the corresponding author, Ming-Tsung Lin, upon reasonable request.

References

[1] R. X. Gao, J. Kruger, M. Merklein, H.-C. Mohring, J. Vancza, "Artificial Intelligence in manufacturing: State of the art, perspectives, and future directions," *CIRP Annals-Manufacturing Technology*, vol. 73, no. 2, pp. 723-749, 2024. https://doi.org/10.1016/j.cirp.2024.04.101.

[2] R. Rakholia, A. L. Suárez-Cetrulo, M. Singh, and R. S. Carbajo, "Advancing manufacturing through artificial intelligence: current landscape, perspectives, best practices, challenges and future direction," *IEEE Access*, vol. 12, pp. 131621-131637, 2024. https://doi.org/10.1109/access.2024.3458830.

[3] S. Chen, E. Turanoglu Bekar, J. Bokrantz, and A. Skoogh, "AI-enhanced digital twins in maintenance: Systematic review, industrial challenges, and bridging research-practice gaps," *Journal of Manufacturing Systems*, vol. 82, pp. 678-699, 2025. https://doi.org/10.1016/j.jmsy.2025.07.006.

[4] M. Soori, B. Arezoo, and R. Dastres, "Machine learning and artificial intelligence in CNC machine tools, A review," *Sustainable Manufacturing and Service Economics*, p. 100009, 2023. https://doi.org/10.1016/j.smse.2023.100009.

[5] V. Nasir and F. Sassani, “A review on deep learning in machining and tool monitoring: methods, opportunities, and challenges,” *The International Journal of Advanced Manufacturing Technology*, vol. 115, no. 9-10, pp. 2683-2709, 2021. https://doi.org/10.1007/s00170-021-07325-7.

[6] Y. Sun et al., “An executable machine-tool-level digital twin for closed-loop CNC milling,” *Robotics and Computer-Integrated Manufacturing*, vol. 103, p. 103363, 2027. https://doi.org/10.1016/j.rcim.2026.103363.

[7] H. Ni, T. Hu, J. Deng, B. Chen, S. Luo, and S. Ji, “Digital twin-driven virtual commissioning for robotic machining enhanced by machine learning,” *Robotics and Computer-Integrated Manufacturing*, vol. 93, p. 102908, 2025. https://doi.org/10.1016/j.rcim.2024.102908.

[8] J. Wang, X. Niu, R. X. Gao, Z. Huang, and R. Xue, “Digital twin-driven virtual commissioning of machine tool,” *Robotics and Computer-Integrated Manufacturing*, vol. 81, p. 102499, 2023. https://doi.org/10.1016/j.rcim.2022.102499.

[9] P. Petráček, P. Fojtů, T. Kozlok and M. Sulitka, “Effect of CNC interpolator parameter settings on toolpath precision and quality in corner neighborhoods,” *Applied Sciences*, vol. 12, no. 19, 9496, 2022. https://doi.org/10.3390/app12199496.

[10]R. Ward, B. Sencer, B. Jones and E. Ozturk, “Accurate prediction of machining feedrate and cycle times considering interpolator dynamics,” *International Journal of Advanced Manufacturing Technology*, vol. 116, no. 1-2, pp. 417-438, 2021. https://doi.org/10.1007/s00170-021-07211-2.

[11]H. Kim, R. Al Kontar, and C. E. Okwudire, “Intelligent feedrate optimization using an uncertainty-aware digital twin within a model predictive control framework,” *IEEE Access*, vol. 12, pp. 49947-49961, 2024. https://doi.org/10.1109/access.2024.3384471.

[12]C.-Y. Lee, S.-H. Hwang, and B.-K. Min, “Simulation-based optimization of CNC interpolator parameters for machining accuracy improvement,” *The International Journal of Advanced Manufacturing Technology*, vol. 119, no. 5-6, pp. 3757-3770, 2022. https://doi.org/10.1007/s00170-021-08434-z.

[13]H. Zhang and T. Yu. Taxonomy of reinforcement learning algorithms. In: Dong H, Ding Z, Zhang S, editors. Deep Reinforcement Learning. Singapore: Springer Singapore; 2020.

[14]S. A. A. Kolagar, M. H. Shahna and J. Mattila, “Combining deep reinforcement learning with a jerk-bounded trajectory generator for kinematically constrained motion planning,” *2025 European Control Conference*, Thessaloniki, Greece, Jun. 24-27, 2025. https://doi.org/10.23919/ECC65951.2025.11187262.

[15]B. Li, H. Zhang, P. Ye, and J. Wang, “Trajectory smoothing method using reinforcement learning for computer numerical control machine tools,” *Robotics and Computer-Integrated Manufacturing*, vol. 61, p. 101847, 2020. https://doi.org/10.1016/j.rcim.2019.101847.

[16]Y. Jiang, J. Chen, H. Zhou, J. Yang, P. Hu and J. Wang, “Contour error modeling and compensation of CNC machining based on deep learning and reinforcement learning,” *International Journal of Advanced Manufacturing Technology*, vol. 118, pp. 551-570, 2022. https://doi.org/10.1007/s00170-021-07895-6.

[17]C. Hu, T. Ou, Y. Zhu, and L. Zhu, “Gru-type larc strategy for precision motion control with accurate tracking error prediction,” *IEEE Transactions on Industrial Electronics*, vol. 68, no. 1, pp. 812-820, 2021. https://doi.org/10.1109/tie.2020.2991997.

[18]C. Hu, T. Ou, H. Chang, Z. Yu, and L. Zhu, “Deep gru neural network prediction and feedforward compensation for precision multiaxis motion control systems,” *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 3, pp. 1377 - 1388, 2020. https://doi.org/10.1109/tmech.2020.2975343.

[19]J. Li, C. Qi, Y. Li, and Z. Wu, “Prediction and compensation of contour error of cnc systems based on lstm neural-network,” *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 1, pp. 572-581, 2022. https://doi.org/10.1109/tmech.2021.3068354.

[20]D. Zhang and H. Chen, “A data-driven iterative pre-compensation method of contouring error for five-axis machine tools,” *The International Journal of Advanced Manufacturing Technology*, vol. 135, no. 3-4, pp. 1669-1684, 2024. https://doi.org/10.1007/s00170-024-14550-3.

[21]S. Ji, H. Ni, T. Hu, J. Sun, H. Yu, and H. Jin, “DT-CEPA: A digital twin-driven contour error prediction approach for machine tools based on hybrid modeling and sparse time series,” *Robotics and Computer-Integrated Manufacturing*, vol. 88, p. 102738, 2024. https://doi.org/10.1016/j.rcim.2024.102738.

[22]H.-W. Chiu and C.-H. Lee, “Intelligent machining system based on CNC controller parameter selection and optimization,” *IEEE Access*, vol. 8, pp. 51062–51070, 2020. https://doi.org/10.1109/access.2020.2980286.

[23]H.-C. Tseng, M.-S. Tsai, C.-C. Cheng and C.-J. Li, “Optimization of computer numerical control interpolation parameters using a backpropagation neural network and genetic algorithm with consideration of corner vibrations,” *Applied Sciences*, vol. 11, no. 4, 1665, 2021. https://doi.org/10.3390/app11041665.

[24]J.-A. Lin, M.-T. Lin, Y.-Z. Li and Y.-H. Wang, “CNC interpolator parameter optimization using deep learning,” *in the 4th IEEE Eurasia Conference on IOT, Communication and Engineering,* Yunlin, Taiwan, Oct. 28-30, 2022. https://doi.org/10.1109/ecice55674.2022.10042898.

[25]J.-P. Jhuang, M.-S. Tsai, Y.-S. Lin, T.-H. Zhang, Z.-W. Jian, and J.-H. Jian, “A novel approach to CNC controller parameter optimization by integrating deep learning model with particle swarm optimization,” in *Proc. 10th World Congress on Mechanical, Chemical, and Material Engineering*, Barcelona, Spain, Aug. 22-24, 2024. https://doi.org/10.11159/icmie24.116.

[26]Y.-C. Chen, M.-T. Lin, Y.-Z. Li, Y.-H. Wang, and G.-Y. Lin, “Enhancing motion performance for CNC machine tools based on AI-driven hybrid model,” *in the 19th IEEE International Conference on Control and Automation*, Tallinn, Estonia, Jun. 30 - Jul. 3, 2025. https://doi.org/10.1109/icca65672.2025.11129860.

[27]Y.-C. Chen, Y.-Z. Li, X.-K. Wang, and M.-T. Lin, “Surrogate-assisted reinforcement learning for adaptive interpolation control under accuracy constraints,” *in the 2026 IEEE/ASME International Conference on Advanced Intelligent Mechatronics,* Genova, Italy, Jul. 7-10, 2026. https://doi.org/10.1109/AIM65483.2026.11658108.

[28]FANUC Corp., AI servo tuning specification manual, 2023.

[29]FANUC Corp., FANUC AC servo motor αi-B/αi series βi-B/βi series linear motor LiS-B/LiS

series DD motor D*i*S-B/D*i*S series, Parameter manual, 2017.

[30]K. Lee, S. Ibaraki, A. Matsubara, Y. Kakino, Y. Suzuki and Y. Yamaoka, "A servo parameter tuning method for high-speed NC machine tools based on contouring error measurement," *WIT Transactions on Engineering Sciences*, vol. 44, pp. 1-12, 2003. https://doi.org/10.2495/lamdamap030171.

[31]S. H. Suh, S. K. Kang, D. H. Chung, and I. Stroud, Theory and Design of CNC Systems. London, UK: Springer, 2008.

[32]Z. Jia, J. Ma, D. Song, F. Wang and Wei Liu, "A review of contouring-error reduction method in multi-axis CNC machining," *International Journal of Machine Tools and Manufacture*, vol. 125, pp. 34-54, 2018. https://doi.org/10.1016/j.ijmachtools.2017.10.008.

[33]A. G. Asuero, A. Sayago, and A. G. González, "The correlation coefficient: an overview," *Critical Reviews in Analytical Chemistry*. vol. 36, no. 1, pp. 41-59, 2006. https://doi.org/10.1080/10408340500526766.

[34]L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: A novel bandit-based approach to hyperparameter optimization," *The Journal of Machine Learning Research*. vol. 18, pp. 1-52, 2018. https://doi.org/10.48550/arXiv.1603.06560.

[35]S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017. https://doi.org/10.48550/arXiv.1705.07874.

[36]A. Bhatt et al., "CrossQ: Batch normalization in deep reinforcement learning for greater sample efficiency and simplicity," *in the 12th International Conference on Learning Representations*, Vienna, Austria, May 7-11, 2024. https://openreview.net/forum?id=PczQtTsTIX.

[37]J. Blank and K. Deb, "pymoo: Multi-objective optimization in Python," *IEEE Access*, vol. 8, pp. 89497-89509, 2020. https://doi.org/10.1109/ACCESS.2020.2990567.

[38]S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *in the 35th International Conference on Machine Learning*, Stockholm, Sweden, Jul. 10-15, 2018. https://doi.org/10.48550/arXiv.1802.09477.