Design of A Robotic Arm Error Compensation System Based on the Transformer Model

Yuqian Li

{22074804@emails.bjut.edu.cn}

College of Computer Science, Beijing University of Technology, Beijing, China

Abstract. Industrial robots, especially robotic arms, are widely used in the industrial field today. In view of the end-effector posture error that exists in the robotic arm during the movement process, this paper takes the UR5 robotic arm as the research object and studies the error compensation method based on the Transformer model. By utilizing the temporal information in the dataset, a multi-feature temporal model compensation method based on the Transformer is proposed. Experimental results show that the proposed method can effectively predict the end-effector pose errors of the robotic arm. Compared with the baseline model, it achieves better prediction performance, demonstrating the effectiveness of the Transformer-based method in robotic arm pose error modeling.

Keywords: Robotic arms pose compensation, Transformer, Industrial robots.

1 Introduction

With the widespread application of robotic arms in intelligent manufacturing, assembly, and precision operation, the motion accuracy of robotic arms has received increasing attention. During the movement of a robotic arm, the end pose of the robotic arm usually deviates from the expected position. This error is affected by various factors such as current, load change, friction, and acceleration during the movement process [1]. Traditional error compensation methods mostly rely on geometric calibration or linear modeling [2]. Although they have certain effects in some scenarios, their modeling ability is still insufficient for error problems with obvious temporal, nonlinear and multi-factor coupling characteristics. With the development of machine learning, especially temporal models, using data-driven models to learn error patterns from robot operation data has gradually become an important direction for robotic arm error compensation research [3].

To address this issue, this paper, based on the UR5 robotic arm dataset released by NIST, first preprocesses features such as joint position, velocity, and current, and constructs end-effector Cartesian pose error labels [4]. then designs an error prediction model based on Transformer, which models the 6-dimensional error of the end-effector through time window input, and optimizes the training process by combining strategies such as standardization, early stopping,

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365357

and learning rate scheduling. Experimental results show that the proposed method can effectively reduce end-effector pose error, and the overall compensation effect is better than the MLP control model. This study can provide a certain reference for error compensation and accuracy improvement of industrial robots.

2 Dataset

2.1 Data Source

The experimental verification data in this paper comes from the standard dataset " Degradation Measurement of Robot Arm Position Accuracy " published by the Intelligent Systems Division of the Engineering Laboratory of the National Institute of Standards and Technology (NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY) [5]. This dataset takes the UR5 industrial robot arm as the research object and contains experimental data collected by the robot under different loads, speeds and cold start conditions. The sampling frequency is 125 Hz. The data content includes not only the position measurement results of the tool center point (TCP) and its deviation from the nominal position, but also the control layer information such as the position, speed, current, acceleration, torque and temperature of each joint. It can comprehensively reflect the operating status and error change characteristics of the robot arm under different working conditions and is therefore suitable for robot arm error modeling and compensation research [6].

2.2 Experimental Platform and Basic Data Information

This standard dataset data is obtained via Universal Robots 5. The data is in CSV files, one of which is a summary Excel file where the deviation between nominal and actual positions will be recorded, and 18 files that have independent samples of 125Hz will have an average of approximately 9, 000 records.

2.3 Data Collection Methods

In order to chart the interaction between the environmental stressors and the operational parameters on the positioning degradation of the robots, NIST researchers conducted a controlled sequence of experimental studies that are rigorous, multivariate, and controlled. The framework switched between separate thermal conditions of cold starts and nominal operating temperatures as well as different payloads (1.6 lbs and 4.5 lbs) and velocity profiles (half and full speed). The result of this matrix was 18 CSV files, which represented six possible operating scenarios of composites with three trials in each scenario to guarantee statistical strength in order to train the model. More importantly, to maintain the integrity of temporary dynamics that are usually obscured by fewer polling rates, every low-level controller telemetry was recorded at a high rate of 125Hz [7].

2.4 Data Processing and Analysis

The data provided by NIST uses the rotation angles of the robot's six joints to record the robot's motion and expected goals. However, the goal we want to optimize is the error of the end effector (TCP). Therefore, we can use forward kinematics to convert the joint angles into the end effector pose [8] and then calculate the deviation between it and the target pose.

According to the DH-modeling theory for robot kinematics, the transformation matrix between link coordinate systems T is as follows:

$$ {}^{i-1}_{i}T = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

Where i represents the coordinate system in which the link joint is located.

The robotic manipulator investigated in this study is the UR5, a six-degree-of-freedom robotic arm. Its transformation matrix T_{base}^{tool} from the base coordinate system $(i = 0)$ to the end-effector coordinate system $(i = 6)$ is the continuous product of the transformation matrices of each link [9]:

$$ {}^{0}_{6}T = \prod_{i=1}^{6} {}^{i-1}_{i}T(\theta_i) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \tag{2} $$

Where$P = [x, y, z]^T$represents the position vector of the end-effector in the base coordinate system, and $R = [n, o, a]$is the rotation matrix of 3×3, representing the rotational angular pose of the end-effector relative to the base. In robot controllers, the pose is usually represented by a rotation vector [10], $[rx, ry, rz]^T$, Given the rotation matrix R , the conversion formula is as follows:

First, calculate the rotation angle [10]$\theta = \arccos\left(\frac{\text{“}tr\text{”}\,(R)-1}{2}\right)$, and the rotation vecto $K = [rx, ry, rz]^T$: $\begin{bmatrix} rx \\ ry \\ rz \end{bmatrix} = \frac{\theta}{2sin\theta} \begin{bmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{bmatrix}$Based on the above calculations, the pose of the robot in the base coordinate system can be represented by a 6-dimensional vector X : $X = [x, y, z, rx, ry, rz]^T$

In this study of robotic arm errors, the pose error of the robotic arm end-effector ΔX s defined as the difference between the target pose X_{target} and X_{actual} : $\Delta X = X_{actual} - X_{target}$; where the position error ΔP and the orientation rotation error ΔO are respectively expressed as: $\Delta P = [\Delta x, \Delta y, \Delta z]^T$, $\Delta O = [\Delta rx, \Delta ry, \Delta rz]^T$ 。

The raw UR5 operation logs were first conditioned to derive a robust dataset for compensation training. Leveraging the previously defined forward kinematics, the recorded joint rotation sequences were projected into Cartesian pose coordinates to define the arm's spatial trajectory. To facilitate faster model convergence and enhance predictive fidelity, these spatial features underwent a normalization process before being partitioned for the construction and validation of the compensation framework.

2.5 Data Analysis

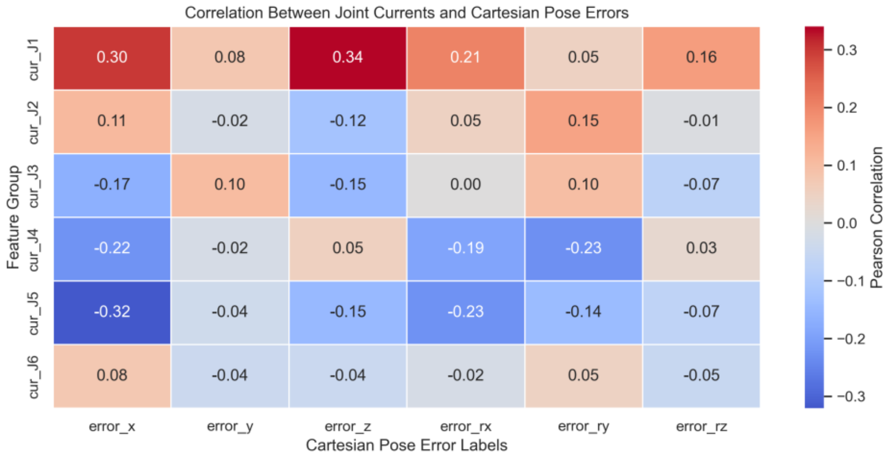

Fig. 1. Correlation Between Joint Currents and Cartesian Pose Errors (halfspeed; playload4.5bl) (Picture credit: Original)

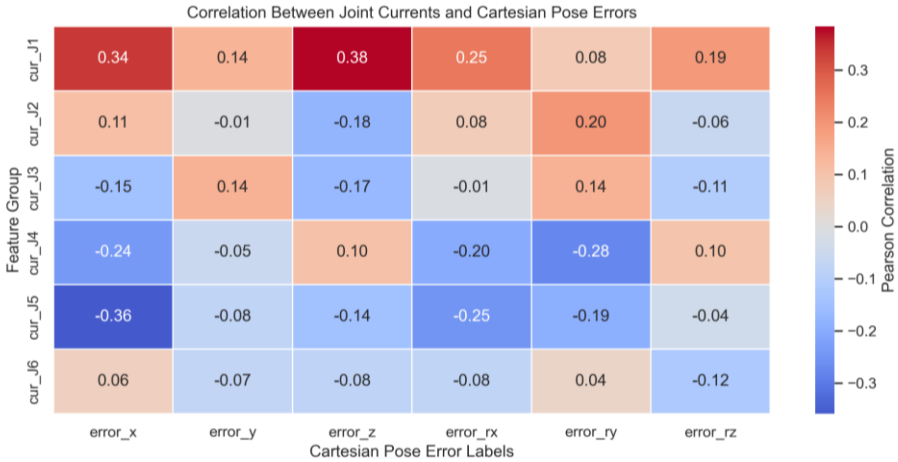

Fig. 2. Correlation Between Joint Currents and Cartesian Pose Errors (halfspeed; playload1.6bl) (Picture credit: Original)

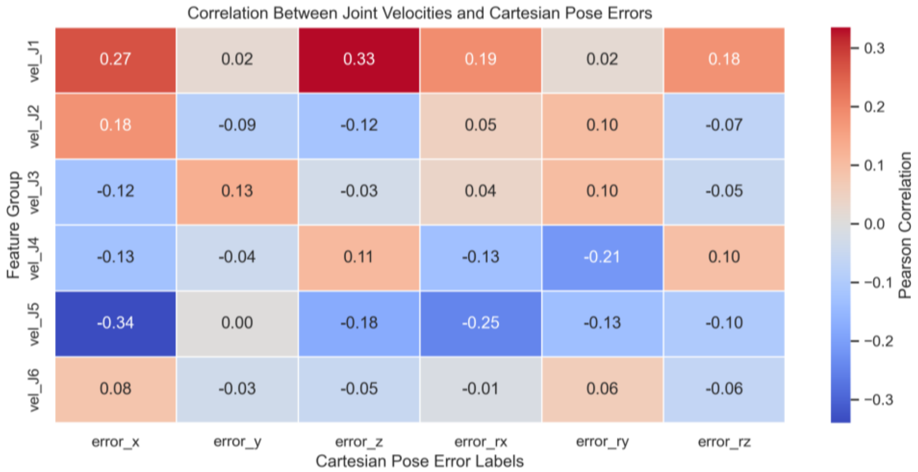

Fig. 3. Correlation Between Joint Velocities and Cartesian Pose Errors (halfspeed; playload4.5bl) (Picture credit: Original)

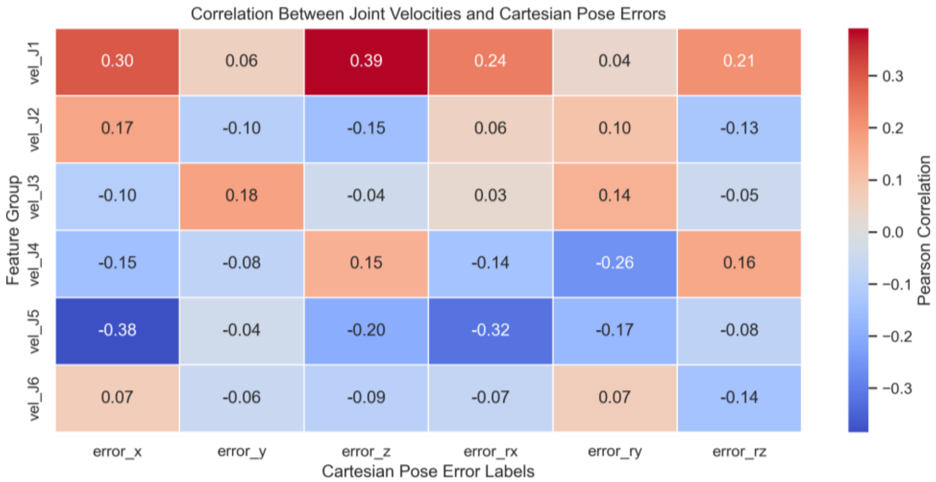

Fig. 4. Correlation Between Joint Currents and Cartesian Pose Errors (halfspeed; playload1.6bl) (Picture credit: Original)

Figure 1 to Figure 4 plot the correlation heatmap to identify the values and the magnitude of the Pearson coefficients between the end-effector error of six dimensions and internal joint telemetry (current and velocity). Instead of the stochastic distribution, the findings display an apparent statistical interconnection along the kinematic chain. Although some of the dimensions of error identified as specific to individual joint characteristics are more sensitive, the resultant cross-correlations, in which one attribute of the joint influences more than one of the error dimensions, underscore the multifaceted nature of positioning error.

It is noteworthy that the strong correlation between current / velocity profiles and pose error entitles their maintenance as input vital features of the compensation framework. Nevertheless, the data also emphasizes the fact that such dependencies are not mere simple linear mappings. This non-linearity is the empirical explanation of why deep learning architectures are applicable, as they are more adapted to capturing the high dimensionality, non-linear connections governing error propagation in robots.

3 Research Methodology

3.1 Problem Analysis

The positional errors caused by robotic end-effectors are frequently difficult to accurately compensate due to the time-varying nature and coupled multifunctional properties. In order to deal with the strong sequential dependencies and non-linearities of these error profiles, we go beyond the memorylessness of the classic Multi-Layer Perceptrons (MLPs) by implementing a regression architecture that is based on Transformers. The main strength here is the self-attention mechanism, which performs the process of dynamically comparing the multi-dimensional feature inputs with past situations. Through knowledge of the pose error in the past by contextualizing the current example error through a sliding temporal window, the model is able to capture the history-dependent nature of the accumulation of inaccuracies in complex robotic maneuvers- an effect which can be readily overlooked in simpler approaches based on compensation [11].

As a method of analysis, we formulate the compensation as a sequence-to-point regression problem, and make use of a look-back time of previous system states to estimate the current error. These input sequences are made of high-dimensional telemetry (target kinematics (position, velocity, acceleration) and dynamic profiles (torque and current). Application of the residuals between the commanded and measured poses as supervisory labels achieves deep temporal representations at various layers with the Transformer Encoder. The resultant output is obtained as a product of the latent vector of the last time step of the sequence that is transformed by a fully connected regression head into a 6-DOF (six degrees of freedom) error vector. This is in the form of a window design so that the framework can take advantage of the natural ability of the Transformer to learn long-range temporal correlations, which is directly translated into increased compensation fidelity.

3.2 Model Structure and Design

Transformer Encoder Regressor has been adopted as the error prediction model in this paper. The entire process follows the following step: 1) First, the input features are projected onto a high-dimensional space and position encoding is appended; 2) Next, 3) temporal features are extracted with the use of a multi-layer Transformer Encoder; 4) finally, the hidden state of the last time step of the sequence is taken and the 6- dimensional error prediction output of the end pose is produced with the aid of the regression.

The model has a time window of 100 in the input layer. Some of the features in each time step are target position, target velocity, target acceleration, target torque, target current, and joint control current. There are 6 joint channels included in each feature, making 36 dimensions in total. To enable later modeling of features, the dimensionality of the input is reduced to a 128-dimensional latent space using a linear layer, followed by the sine - cosine position coding which adds back temporal order information.

The data was divided not randomly but chronologically with a definite percentage of 70, 15 and 15 respectively of training, validation and testing. Holding the chronological order further to the true situation in accordance with the model generalization capability at future time points and prevents information leakage problems of random partitioning at the time-series tasks. In line with this, set shuffle=False was applied to all the data loaders of training, validation and testing.

3.3 Training Strategy and Parameter Settings

In this paper, the Mean Squared Error loss (MSELoss) is taken as a model training loss function to reduce the squared error between predicted and actual errors. The AdamW optimizer is chosen, a learning rate of is used, weight decay is used and a cosine annealing learning rate scheduling strategy is applied. The network was trained in batch size of 256 to a maximum of 100 epochs. Gradient clipping was added to the backpropagation workflow to achieve numerical stability and gradient explosion. The generalization was done through an early stopping mechanism whose patience was 20 epochs, the training was terminated when validation performance did not indicate an increase in marginal performance and the best-performing weights were reverted to the last-performing checkpoint.

4 Experiments

4.1 Experimental Environment

Any computational experiment was run in the Google Colab environment, using the NVIDIA T4 GPU backend hardware-accelerated. It was built using the PyTorch framework as its software backbone and Pandas and NumPy to coordinate data and analyze post-hoc results. The study process used a sequential pipeline: starting with pre-processing of original data and model training and then to the inference stage of prediction errors, followed by the ultimate benchmarking of the compensation effectiveness.

4.2 Evaluation Metrics

The effectiveness of the suggested compensation model is compared to the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE). These measures can be used to estimate how much left over deviation in end-effector poses by a direct statistical value of the error reduction achieved by the system by the use of these measures comparing the state of the system to the post-compensated state. Since the research is aimed at predictive fidelity, as well as practical reduction of positioning errors, the indices provide a stringent measurement of the operational feasibility of the model and its ability to alleviate the non-linear errors in the multifaceted profiles of errors found in the robotic movement.

To determine the levels of average degree of absolute deviation between predicted values and true values, the mean absolute error (MAE) is used, and is described in the following manner:

$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}^{i} - y_i\right|$, Mean Square Error (RMSE) is used to measure the degree of dispersion of the prediction errors, and is represented as follows: $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}^{i} - y_i\right)^2}$

MSE is more sensitive to big errors than MAE, hence it is able to show whether there is big deviation in prediction by the model on local samples. In the case of error compensation associated tasks of robotic arms, when there are huge prediction delays at some point in time, it could have a direct impact on the compensation effect and stability of the system hence, RMSE is effective as an auxiliary of MAE. In the present paper, RMSE is additionally applied in order to measure the accuracy of the prediction of each dimension of errors and level of residual error on compensation. The lower the RMSE, the more consistent the model results and the less the overall change of the error.

4.3 Experimental Results

Application of the compensation model produced a significant decrease in MAE in all six pose dimensions as visualized in Figures 5 -10. Quantitatively, the world mean MAE was depressed with a pre-correction baseline of 0.00006916 to 0.00004272, which is an average performance improvement of 39.93. The fact that this improvement is consistent throughout the whole 6-DOF (six degrees of freedom) space indicates that the framework was able to capture the actual error dynamics and not just overfit to particular axes. These findings verify the ability of the model to effectively and reliably counteract end-effector errors to give strong empirical support to the suggested compensation scheme.

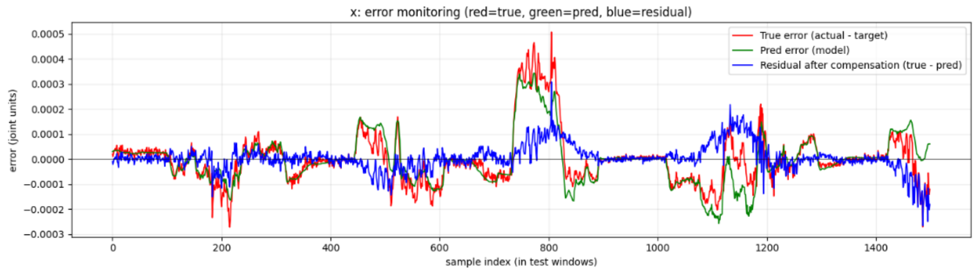

Fig. 5. Error compensation results in the X dimension (Picture credit: Original)

Fig. 6. Error compensation results in the Y dimension (Picture credit: Original)

Fig. 7. Error compensation results in the Z dimension (Picture credit: Original)

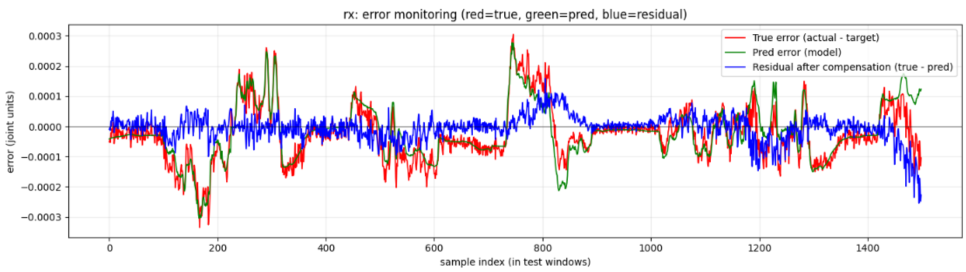

Fig. 8. Error compensation results in the RX dimension (Picture credit: Original)

Fig 9. Error compensation results in the RY dimension (Picture credit: Original)

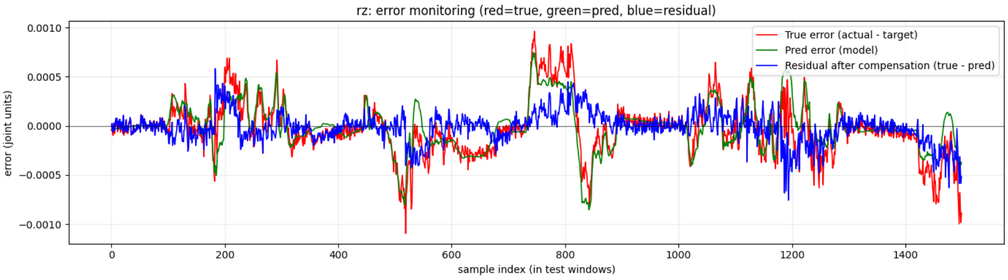

Fig. 10. Error compensation results in the RZ dimension (Picture credit: Original)

4.4 Comparative Experiment

In order to make a comparison of the proposed Transformer architecture with a Multilayer Perceptron (MLP), the MLP was run as a control to preserve the parity between the input features and data conditioning to provide a rigorous comparative analysis. Here the MLP is used to interpret the temporal window as a flattened feature map, with the help of conventional fully connected layers to project an input into the 6-dimensional error signal. Although the MLP showed some ability to reduce the errors, producing a smaller 16.71% distribution in the mean MAE (between 0.00006914 and 0.00005804), its performance was highly variable with gains concentrated mostly in the J3 dimension and less gains in others.

Compared to the results of the Transformer (Tables 1 and 2), the discrepancy in the performance is even greater. The given framework showed an improvement of 39.93% that mitigated the mean MAE to 0.00004272 and significantly improved over the MLP in the whole pose space. Such a difference highlights one of the inherent drawbacks of the static, dense networks: they cannot solve the complex temporal dependencies of robotic motion. The Transformer uses its self-attention mechanism to better decode the non-linear interaction between the system condition at the past and the instantaneous positioning error. Such comparative results support the need for sequence-conscious modelling of high-fidelity compensation in dynamical operating conditions.

Table 1. MLP Model Experimental Results

X	MAE	before=0.00004113	MAE	after=0.00003774	improve=8.23%
y	MAE	before=0.00000466	MAE	after=0.00000407	improve=12.66%
z	MAE	before=0.00003530	MAE	after=0.00002216	improve=37.23%
rx	MAE	before=0.00004532	MAE	after=0.00003996	improve=11.82%
ry	MAE	before=0.00015735	MAE	after=0.00013109	improve=16.69%
rz	MAE	before=0.00013108	MAE	after=0.00011320	improve=13.64%

Table 2. Experimental results of the Transformer model

x	MAE	before=0.00004161	MAE	after=0.00002432	improve=41.56%
y	MAE	before=0.00000463	MAE	after=0.00000277	improve=40.22%
z	MAE	before=0.00003565	MAE	after=0.00001863	improve=47.73%
rx	MAE	before=0.00004470	MAE	after=0.00002804	improve=37.26%
ry	MAE	before=0.00015770	MAE	after=0.00009571	improve=39.31%
rz	MAE	before=0.00013066	MAE	after=0.00008686	improve=33.52%

5 Conclusions

The current paper responds to the ongoing issue of pose error in UR5 manipulators by proposing a predictive framework of compensation based on transformer. Through the synthesis of the NIST public dataset and high-fidelity multi-type telemetry, we showed that sequence aware architectures apply significantly better to Mean Absolute Error (MAE) than traditional MLP baselines. The fact that this performance gap is an intrinsic feature of the Transformer underlines why the Transformer is well-suited to address the non-linear coupling and time-dependence of errors that have traditionally plagued models of the process of static errors, especially because of the dynamics of its self-attention.

Admittedly, such off-line benchmarks are encouraging, but the shift to live on-arm validation is still a decisive borderline. Existing limitations, in particular, the necessity to achieve more precise feature engineering and explore the underlying physics of the error formation more thoroughly, offer a clear direction for forthcoming work. The following actions will be taken to move these models to physical platforms to operate on real-time compensation, consider sensor fusion, and explore model-pruning at the edge to ensure engineering viability. This study, finally, provides a connected line between the characterization of data and its experimental validation, providing a blueprint of scalability in the enhancement of the precision of intelligent robotic systems in the industrial industry.

References

[1] Jiang, Y., Yu, L., Jia, H., et al.: Absolute positioning accuracy improvement in an industrial robot. Sensors. 20(16), 4354 (2020)

[2] Nguyen, T.D.V., Bonnet, V., Fernbach, P., et al.: Improving operational accuracy of a mobile manipulator by modeling geometric and non-geometric parameters. In: 2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids). IEEE. pp. 965-972 (2024)

[3] Liu, W., Liu, S., Deng, Z., et al.: Research progress on positioning error compensation technology of industrial robot. Journal of Mechanical Engineering. 59(17), 1-1 (2023)

[4] Gomez-Casas, J., Toro-Arcila, C.A., Rodríguez-Rosales, N.A., et al.: Data-driven kinematic model for the end-effector pose control of a manipulator robot. Processes. 12(12), 2831 (2024)

[5] National Institute of Standards and Technology: Degradation measurement of robot arm position accuracy. National Institute of Standards and Technology. Retrieved from https://www.nist.gov/el/intelligent-systems-division-73500/degradation-measurement-robot-arm-position-accuracy (2019)

[6] Qiao, G., Weiss, B.A.: Accuracy degradation analysis for industrial robot systems. In: International Manufacturing Science and Engineering Conference. American Society of Mechanical Engineers. 50749, V003T04A006 (2017)

[7] Qiao, G., Schlenoff, C., Weiss, B.A.: Quick positional health assessment for industrial robot prognostics and health management (PHM). In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE. pp. 1815-1820 (2017)

[8] Nguyen, H.N., Zhou, J., Kang, H.J.: A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network. Neurocomputing. 151, pp. 996-1005 (2015)

[9] Sellami, S., Klimchik, A.: A deep learning based robot positioning error compensation. In: 2021 International Conference “Nonlinearity, Information and Robotics” (NIR). IEEE. pp. 1-5 (2021)

[10] Campa, R., De La Torre, H.: Pose control of robot manipulators using different orientation representations: A comparative review. In: 2009 American Control Conference. IEEE. pp. 2855-2860 (2009)

[11] Zhang, W., Yang, D., Xu, Y., et al.: DeepHealth: A self-attention based method for instant intelligent predictive maintenance in industrial Internet of Things. IEEE Transactions on Industrial Informatics. 17(8), pp. 5461-5473 (2020).