euspen's 25th International Conference & Exhibition, Zaragoza, ES, June 2025
www.euspen.eu

AI-assisted high-precision dynamic error calibration for on-machine measurement systems

Weixin Cui, Maomao Wang, Wenbin Zhong*, Shan Lou, Paul Scott, Xiangqian Jiang, Wenhan Zeng

EPSRC Future Metrology Hub, Centre for Precision Technologies, School of Computing and Engineering, University of Huddersfield, Huddersfield, HD1 3DH, UK
**Corresponding author: W.Zhong@hud.ac.uk*

Abstract
In precision optics manufacturing, transferring components between multiple machines introduces uncertainties due to datum changes. On-machine surface measurement (OMSM) integrates a non-contact profiler directly with the manufacturing platform, overcoming this bottleneck. However, dynamic measurement errors, arising from both the host machine and the integrated subsystem, significantly degrade the measurement accuracy. Traditional offline system identification methods struggle to capture the overall system response using a generalised model under time-varying dynamics and uncertain model orders. This study proposes a novel AI-assisted calibration approach based on deep learning to enable adaptive error compensation in OMSM systems. Specifically, a regression model using Long Short-Term Memory (LSTM) networks is developed to learn the complex temporal dependencies and nonlinear dynamics inherent in the system. The experimental results demonstrate that the LSTM-based calibration achieves comparable accuracy to traditional methods while offering greater adaptability, automation, and scalability. In a practical test on an off-axis parabolic (OAP) surface, the model significantly reduced the root mean square (RMS) of measurement error from 1.9073 µm to 0.4324 µm by compensating for dynamic errors caused by vibrations and fluctuating conditions. These findings highlight the potential of LSTM networks to improve the efficiency and robustness of OMSM systems, contributing to the development of intelligent and adaptive error calibration frameworks for smart manufacturing.

Keywords: On-machine surface measurement (OMSM), Dynamic error calibration, System identification, Artificial intelligence (AI), Deep learning, Long short-term memory networks (LSTM)

1. Introduction

In precision optics manufacturing, transferring components between machines introduces uncertainties due to changes in the datum reference, which adversely affect measurement precision. On-machine surface measurement (OMSM) addresses this issue by integrating a non-contact profiler directly with the manufacturing platform [1,4]. However, OMSM systems are susceptible to dynamic measurement errors arising from the interaction between the profiler and the host machine, which can significantly degrade measurement accuracy [2,3]. These interactions are inherently coupled and complex, posing substantial challenges for conventional fixed-order modelling approaches, which often fail to capture the underlying dynamics effectively. To overcome these limitations, this study proposes a learning-based approach that leverages neural networks to model and compensate for dynamic errors with high accuracy. The architecture of the proposed method is illustrated in Fig. 1. By capturing complex temporal dependencies and nonlinear relationships within the system data, the neural network-based model demonstrates strong potential as a reliable, adaptive, and efficient solution for dynamic error calibration in OMSM systems.

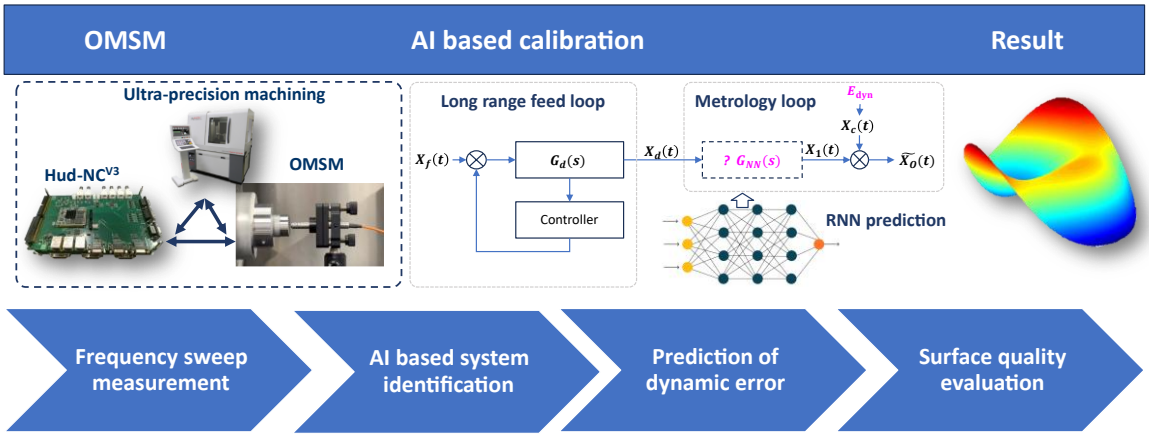

Figure 1. Diagram of the AI-assisted calibration workflow.

2. Related work

Traditional calibration methods rely on error models that incorporate the kinematic and dynamic properties of machine systems. These approaches typically require data collection under controlled conditions, followed by model selection and parameter estimation through multiple experimental trials [5]. However, developing an accurate physical model is both time-consuming and labour-intensive. Moreover, such models are typically fixed in structure and lack adaptability to real-time variations—such as self-excited vibrations or environmental fluctuations—thereby limiting their effectiveness in dynamic manufacturing environments.

Deep learning has advanced time series prediction by capturing temporal and nonlinear patterns. Common models include Recurrent Neural Networks (RNNs) [6], Long Short-Term Memory Networks (LSTMs) [7], Gated Recurrent Units (GRUs) [8], and Transformers [9]. RNNs handle sequences but struggle with long-term dependencies due to vanishing gradients. LSTMs solve this using gating mechanisms, allowing them to learn both short- and long-term patterns [10]. GRUs offer a simplified alternative to LSTMs with similar performance and lower computational cost [11]. Transformers, such as BERT and GPT, use self-attention to capture global dependencies and achieve strong results, though they require large datasets and significant computational resources [12,13]. Among these models, LSTMs provide an effective trade-off between accuracy and efficiency, making them well-suited for time series prediction in data- and resource-constrained environments.

3. Methodology

3.1. System Description and Problem Formulation

The OMSM system is set up on a Moore Nanotech 650FG v2 diamond turning machine, as shown in Fig. 2. Measurement data are collected using the in-house HUD-NCv3 controller, generating continuous surface profiles with six data channels, including probe distance, C-axis angle, and X-Y-Z-B axis positions. The probe is aligned to the surface through controlled machine motion. A Precitech CHRocodile chromatic confocal sensor, with a 100 µm range and 9 nm repeatability, measures the surface along an Archimedean spiral path at a spindle speed of 30 rpm and an X-axis feed rate of 3 mm/min.

The system's dynamic-induced measurement error is defined as $E(t) = D_2(t) - D_1(t)$. Where D1 is the reference reading from the grating encoder and D2 s the distance measured by the chromatic confocal sensor. When the linear motor moves the sensor, the measured distance follows the system response, represented as $D_2(t) = G(D_1(t))$. Where G(•) is the transferring function of the system. The traditional calibration process aims to estimate the G(•) employing system identification. However, the complexity and varying configurations of the feed system make optimal estimation impractical and inefficient.

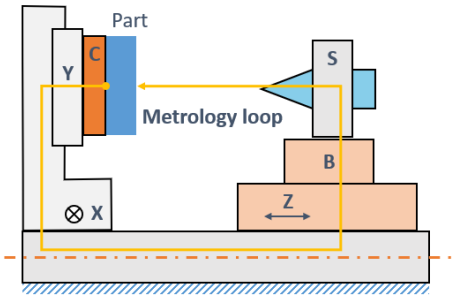

Figure 2. Measurement system set up.

3.2. Proposed AI-Assisted Calibration Method

As shown in Fig. 3, the architecture of the proposed method begins with a sequence input layer that processes time-series data, including operational parameters and environmental conditions. These inputs are fed into multiple LSTM layers to capture temporal features and nonlinear patterns. Finally, fully connected layers convert these features into regression outputs that predict dynamic measurement errors [14]. The model is trained using the MSE loss function to minimise the difference between predicted and actual errors, ensuring accurate and reliable calibration [15]. The internal mechanism of the LSTM network [16, 10], shown in Fig. 4, consists of several key gates that regulate the flow of information:

- **Forget Gate**: Determines which information from the previous memory state should be discarded. $f_t = sigmoid(W_f \cdot x_t + U_f h_{t-1} + b_f)$
- **Input Gate**: Decides how much of the current input to store in the cell state. $i_t = sigmoid(W_i \cdot x_t + U_i.h_{t-1} + b_i)$
- **Candidate Gate**: Generates new values to update the cell state. $\tilde{C}_t = \tanh(W_c \cdot x_t + U_c.h_{t-1} + b_c)$
- **Cell State Update:** Updates the cell state by combining the forget gate and candidate cell state. $C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$
- **Output Gate**: Decides how much of the cell state to pass to the hidden state. $o_t = sigmoid(W_o \cdot x_t + U_o.h_{t-1} + b_o)$

Where $W_f, W_i, W_c, W_o, U_f, U_i, U_c$ are model weights; b_f, b_i, b_c, b_o are bias; x_t, o_t are input and output; h_{t-1}, h_t denotes hidden states, C_{t-1}, C_t represents candidate cell. These gates, controlled by sigmoid and tanh functions, allow the LSTM network to update its memory and capture both short-term and long-term patterns in sequence data.

The calibration process is divided into two phases: training and prediction. During the training phase, historical data collected from the OMSM system under varying operational conditions is used to train the LSTM model offline. Once trained, the model is validated and tested to assess its ability to predict dynamic measurement errors. These predictions are then used to correct measurement inaccuracies, improving the precision of the OMSM system and confirming the effectiveness of the proposed method.

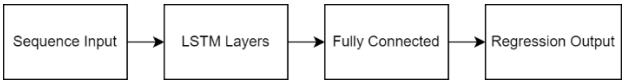

Figure 3. LSTM-based model diagram for measurement error prediction

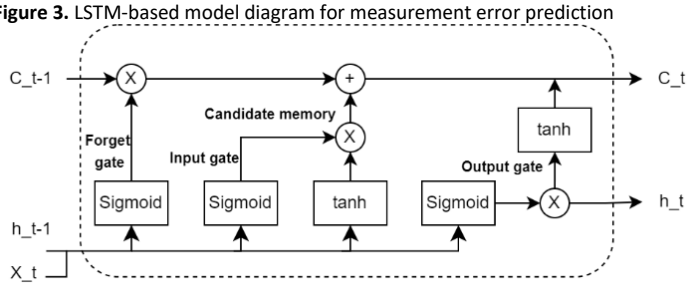

Figure 4. LSTM neural network architecture and work principle

4. Experiments

4.1. Data Acquisition and Experimental Setup

To ensure the LSTM model can generalise dynamic errors under typical conditions, a dataset was collected using the OMSM system in a setup that mimics real-world environments, including vibrations and system fluctuations. Measurements on precision optical components captured errors from both the host machine and profiler, providing data for training, validation, and testing. Under the constraint of requiring ground-truth reference measurements, the dataset was collected from a single freeform optical surface. Although this limits the ability to

evaluate the model on unseen surfaces, the calibration targets system-induced errors originating from the measurement system rather than being surface-dependent. The selected surface provides sufficient variation in measurement conditions to train and evaluate the model effectively.

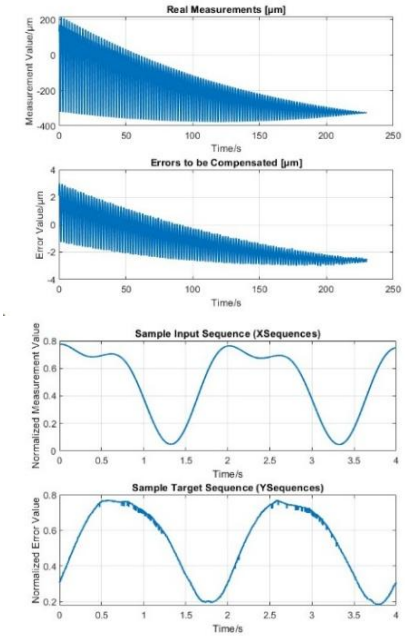

Figure 5. Measurements vs errors need to be compensated; sampled segmented sequence for training

The collected data consisted of 1D vectors representing real-time measurements and their corresponding reference values. A preprocessing pipeline was developed to prepare the data for LSTM regression training. First, both input (Xdata) and target (Ydata) were normalised to the [0, 1] range using min-max scaling to ensure consistent feature scales for stable training. The data was then segmented into overlapping sequences of a set length using a tuneable stride, allowing the model to capture temporal dependencies (see Fig. 5), as required by the LSTM input format. Next, the dataset was divided into training (70%), validation (15%), and test (15%) sets to support independent training, tuning, and evaluation. The sequences were then organised into cell arrays, with each cell holding one sequence, enabling efficient LSTM processing. This preprocessing approach ensures the data is correctly scaled, formatted, and partitioned for effective model training and evaluation.

Table 1 Measurements set-up

Conditions	**System set-up**	**Number**
Freeform surface	Elastic	230095×1
SampleLength=4000	StrideSize=50	TotalNumber=4521
Training	**Validation**	**Test**
1×4000×3165	1×4000×678	1×4000×678

4.2. Results and Analysis

LSTM training hyperparameters play a key role in model performance and convergence. *MinibatchSize* sets the number of samples per update—smaller values may improve generalisation but cause noisy gradients, while larger ones reduce noise but need more memory. *Layer Depth* controls how many LSTM layers are stacked; deeper networks capture complex patterns but may overfit and increase computation. The *Adam* optimizer is used for its adaptive learning rate and stable convergence. *InitialLearnRate* sets the update size—too high can cause instability, too low slows training. *LearnRateDropPeriod* and *LearnRateDropFactor* reduce the learning rate over time to escape local minima. *GradientThreshold* limits gradient size to prevent instability. *MaxEpoch* defines the number of training cycles, balancing accuracy and time. *ValidationFrequency* controls how often validation runs to track overfitting and support early stopping. All experiments are conducted on MATLAB(2024a), the system with specifications of core i7-11800H, 11th generation Intel processor with 16GB RAM, and GPU of NVIDIA GeForce RTX 3060 of 3584 CUDA core.

Table 3 Hyper-parameters

Parameters	**Definition**	**Value**
MinibatchSize	Number of samples each batch	32
Layer depth	Number of LSTM layers	200
Optimizer	Optimisation algorithm	Adam
InitialLearnRate	Initial learning rate for training	0.001
MaxEpoch	Total training epochs	100
ValidationFrequency	Validation in epochs	10
Loss function	$\sum(y_i - \hat{y}_i)^2$	MSE

The proposed model is tested on the whole surface dataset and compared to the traditional transfer function method, focusing on accuracy, generalisation, and efficiency. With a test dataset MSE of 0.001, accuracy is evaluated using metrics RMSE, MAD, and PV value to quantify the error reduction. Efficiency is assessed by analysing the time required for prediction and tuning, providing insights into its practical use.

Table 4 Evaluation of LSTM and transfer function

Parameter	**Value**
CPU Training	4-6h
Process	1.4s
GPU Training	1.5h
Uncompensated errors	RMSE = 1.9073
	MAD = 1.8792
	PV = 6.0042
Accuracy (Errors after LSTM)	**RMSE = 0.4324**
	MAD = 0.1412
	PV = 349.0326
Accuracy (Errors after transfer function)	RMSE = 0.6577
	MAD = 0.1804
	PV = 6.7721

Tab. 4 shows LSTM-based model outperforms the traditional transfer function approach in accuracy, efficiency, and consistency. It achieves lower prediction errors, with an RMSE of 0.4324 and MAD of 0.1412, compared to 0.6577 and 0.1804 from the traditional method. The LSTM model also offers faster GPU training (1.5 hours) and quick inference (1.4 seconds), making it suitable for practical on-machine measurement applications. However, the LSTM results show a high peak-to-valley (PV) value of 349.0326. This is not due to a unit mismatch but occurs during the model's initial characterisation phase. At the beginning of inference, the LSTM needs a few time steps to synchronise its internal memory states with the input data. During this warm-up period, temporary spikes can appear. These stabilise as the model receives more input and adjusts its internal parameters. In real-time applications, this issue can be addressed by discarding initial outputs, adding a brief initialisation step, or applying constraints to limit sudden output changes. These results show that the LSTM model provides accurate and reliable predictions with lower computational cost, making it practical and scalable for real-world use.

Fig. 6 provides visual confirmation of the LSTM model's effectiveness. Fig. 6(a) demonstrates the test data performance of error prediction, showing a close alignment between the predicted errors and the reference values, with minimal discrepancies. Fig. 6(b) and (c) demonstrate how the AI-compensated measurements align with the reference data over time, faithfully reproducing dynamic oscillations and amplitude decay. The AI-based approach accurately captures the oscillations and decreasing amplitude, showing its precision in handling dynamic signals. Fig. 6(d) compares uncompensated, transfer function– compensated, and AI-compensated errors, highlighting the AI method's better performance in reducing

errors and stabilising the measurements. Overall, the high PV is due to the LSTM's transient warm-up phase, but the model performs well once stabilised. Even with data from a single surface, the LSTM model outperformed the traditional transfer function method, showing lower RMSE, reduced MAD, and better signal stability, confirming its effectiveness for real-world error compensation.

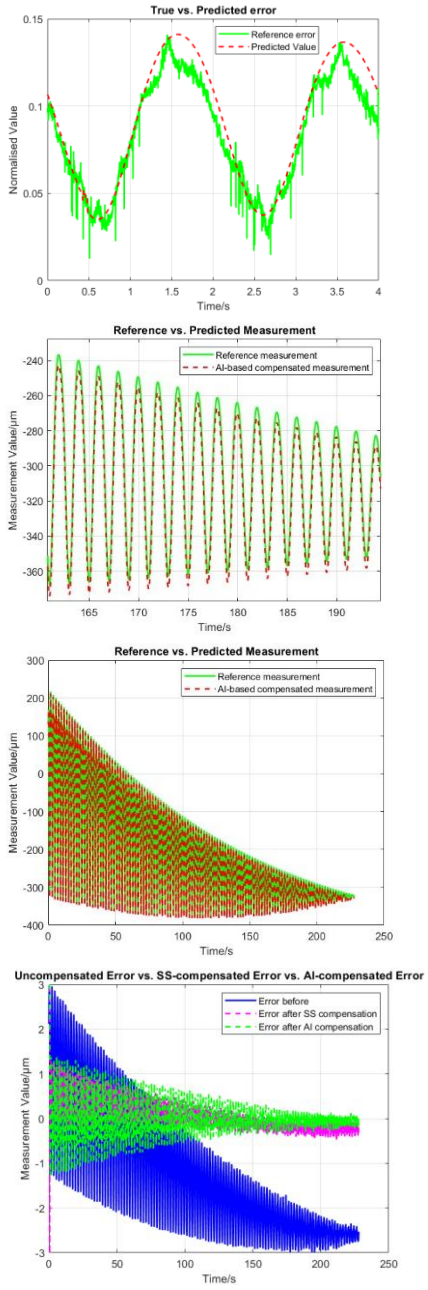

Figure 6. Evaluation performance of error predictions

5. Conclusion and Future Work

This study proposes an AI-assisted calibration framework using LSTM networks to correct dynamic measurement errors in OMSM systems. The model captures system dynamics and effectively reduces residual errors, with the median dropping from 1.8792 µm to 0.1412 µm and RMSE from 1.9073 µm to 0.4324 µm. Unlike traditional methods, it avoids complex modelling and tuning, offering a more practical solution. These results highlight the potential of AI to improve calibration, precision, and automation in smart manufacturing.

While the LSTM-based framework shows strong performance, its implementation faces some challenges. It relies on sufficient and diverse historical data with accurate reference measurements to ensure reliable learning and prevent overfitting. Since our dataset comes from a single surface, generalisation to unseen surfaces has not been fully assessed; however, the model focuses on correcting system-induced errors rather than surface-specific features. Real-time use may also require significant computing resources, especially in high-frequency systems. Future work will expand the dataset to include more operating conditions, surface types, and system dynamics to improve generalisation. The framework will be adapted for real-time use by embedding it into machine controllers for on-the-fly calibration. We also plan to explore hybrid models, such as combining LSTM with transformers and attention mechanisms, to enhance accuracy and flexibility. Interpretability methods will be added to build user trust and support wider adoption in precision manufacturing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors gratefully acknowledge the UK's EPSRC funding of Future Metrology Hub (Ref: EP/P006930/1, EP/Z53285X/1) and the UK's STFC-IPS funding (Grant Ref: ST/W005263/1).

References

[1] Gao, W., Haitjema, H., Fang, F. Z., Leach, R. K., Cheung, C. F., Savio, E., & Linares, J. M. (2019). On-machine and in-process surface metrology for precision manufacturing. *CIRP Annals*, **68**(2),843–866.

[2] Wang, M., Zhong, W., Yu, G., Scott, P., Jiang, J., & Zeng, W. (2024). Mechanical resonance induced measurement uncertainty of ultra-precision on-machine surface measurement system. In *Proceedings of EUSPEN 24th International Conference & Exhibition* (pp. 111-114). [ICE24147] euspen.

[3] Zhong, W., Tong, Z., & Jiang, J. (2021). Integration of On-machine Surface Measurement into Fast Tool Servo Machining. In E. Ozturk, D. Curtis, & H. Ghadbeigi (Eds.), *9th CIRP Conference on High Performance Cutting, HPC 2020*, *Procedia CIRP*, **101**, 238–241.

[4] Tong, Z., Zhong, W., Zeng, W., & Jiang, J. (2021). Closed-loop form error measurement and compensation for FTS freeform machining. *CIRP Annals - Manufacturing Technology*, **70**(1), 455-458.

[5] Gao, W., Ibaraki, S., Donmez, M. A., Kono, D., Mayer, J., Chen, Y.-L., Szipka, K., Archenti, A., Linares, J.-M., & Suzuki, N. (2023). Machine tool calibration: Measurement, modeling, and compensation of machine tool errors. *International Journal of Machine Tools & Manufacture*, **187**, 104017.

[6] Cardot, H. (2011). Recurrent Neural Networks for Temporal Data Processing. In R. Boné & H. Cardot (Eds.), *IntechOpen*.

[7] Graves, A. (2012). Long short-term memory. In *Supervised Sequence Labelling with Recurrent Neural Networks* (pp. 37–45).

[8] Gao, Y., & Glowacka, D. (2016, November). Deep gate recurrent neural network. In *Asian Conference on Machine Learning* (pp. 350–365). PMLR.

[9] Vaswani, A. et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

[10] Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, **31**(7), 1235–1270.

[11] Khandelwal, S., Lecouteux, B., & Besacier, L. (2016). Comparing GRU and LSTM for automatic speech recognition (Doctoral dissertation, LIG).

[12] Soydaner, D. (2022). Attention mechanism in neural networks: Where it comes and where it goes. *Neural Computing and Applications*, **34**(16), 13371–13385.

[13] Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, **3**, 111–132.

[14] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, **9**(8), 1735–1780.

[15] Naidu, G., Zuva, T., & Sibanda, E. M. (2023, April). A review of evaluation metrics in machine learning algorithms. In *Computer Science On-line Conference* (pp. 15–25). Cham: Springer International Publishing.

[16] Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, **53**(8), 5929–5955.