



# A novel neural network model for rolling linear guide pair optimization design

Chenghao Song<sup>1</sup>, Weiqi Du<sup>1,2</sup>, Shuxin Li<sup>1</sup>, and Junjun Han<sup>3</sup>

<sup>1</sup>School of Mechanical Engineering and Mechanics, Ningbo University, Ningbo 315211, China

<sup>2</sup>Key Laboratory of Impact and Safety Engineering of MOE, Ningbo 315211, China

<sup>3</sup>Ningbo Hilectro Precision Machinery Co., Ltd, Ningbo 315211, China

**Correspondence:** Weiqi Du (duweiqi@nbu.edu.cn)

Received: 6 November 2025 – Revised: 29 December 2025 – Accepted: 22 January 2026 – Published: 26 February 2026

**Abstract.** This paper addresses the inefficiency of conventional design methods for rolling linear guide pairs, which rely on finite-element analysis and large data demands. This study pioneers a few-shot learning framework based on meta-learning, employing a model-agnostic meta-learning strategy to train an inverted-bottleneck residual fully connected network. The network achieves high prediction accuracy ( $R^2 > 0.91$ ) with only 126 samples. An integrated parametric platform reduces modelling time from 50 to 2–3 min, significantly improving efficiency. Optimization via sequential least squares quadratic programming demonstrates a 57 % reduction in vertical deformation. However, the current work focuses on static performance optimization, leaving dynamic aspects for future research. This approach offers an efficient and data-effective paradigm for precision mechanical component design.

## 1 Introduction

The rolling linear guide is a core transmission component in precision mechanical equipment. Its structural performance directly determines the positioning accuracy and service life of high-end equipment, such as computer numerical control (CNC) machine tools and industrial robots (Yang et al., 2020; Wang et al., 2018; Zha et al., 2024; Quan and Zhao, 2024). The traditional guide design method relies much on empirical trial and error (Phuyal et al., 2020; Sahoo and Lo, 2022; Liu et al., 2023; Staroszyk et al., 2024). Designers repeatedly adjust geometric parameters and conduct the computer-aided engineering (CAE) analysis to improve mechanical performance, which is quite time-consuming (Zou and Wang, 2015; Tong et al., 2020). Therefore, a rapid and effective design method should be developed for the structural optimization of a rolling linear guide.

Several studies have been conducted for the investigation of rolling linear guide pairs. For instance, Shimizu (1998) and Ohta and Hayashi (2000) were dedicated to developing rigid models for rolling linear guides based on Hertz contact theory. Cheng (2021) introduced a novel stiffness matrix

for roller linear guides, investigating the relationship between non-uniform load distribution and contact states. Xu et al. (2023) developed an analytical dynamic model for flexible linear guides, investigating the vibration behaviour of carriages under external periodic excitations in multiple directions. Compared to numerical analysis methods, the finite-element method (FEM) provides higher computational efficiency and superior adaptability. For example, Sun et al. (2015) established a high-fidelity finite-element model accounting for ball-raceway contact. Wei et al. (2022) used FEM to characterize the impact of structural parameters on guide pair mechanical performance. Tong et al. (2019) employed FEM to develop a 5-degree-of-freedom model for calculating the fully occupied stiffness matrix of linear ball guides. Up until now, few studies have paid attention to the structural optimization of rolling linear guides. In addition, structural optimization design with the results of FEM analysis is quite time-consuming and labour-intensive because FEM analysis cannot directly give the explicit relationships between structure and performance.

Traditional optimization algorithms like PSO and GA are highly effective in handling complex problems. However,

they require thousands of computations on high-precision models, and so the cycle is very time-consuming. For example, Tong et al. (2020) employed a particle swarm optimization algorithm, using 2000 particles and running for 150 iterations to optimize the design of the linear guide. Wang et al. (2016) utilized finite-element analysis and a genetic algorithm model, conducting 1580 calculations to optimize the structure of a wind turbine blade. In recent years, machine learning techniques (Ramu et al., 2022) have offered another option for designers. For instance, Kazem et al. (2025) proposed a RAGN-R (surrogate multi-subject ensemble machine learning method) for estimating the mechanical properties of advanced materials. This method can directly establish the connection between structure and performance, enabling highly accurate predictions. Up to now, there have been many types of surrogate models. For instance, Fishwick (1989) used a multi-layer neural network model as a simulation model for a basic ballistic model. Jiang et al. (2025) proposed a multi-fidelity fully connected neural network (FCNN) model to optimize the fin structure of rocket projectiles. Roy et al. (2019) developed a support vector regression (SVR) model for the reliability estimation of engineering structure. Nagendra et al. (2005) developed a response surface model based on the optimal noise, vibration, and harshness (NVH) response sensitivity data for optimizing tailored rolled blank (TRB) components. Machine learning, particularly surrogate models, provides a novel solution for the optimization of rolling linear guide pairs. It directly maps the complex nonlinear relationships between geometric parameters (contact angle and curvature ratio) and mechanical properties (including stiffness and stress). It reduces FEM time from hours to minutes, saving significant iterative computation costs for optimization design.

However, surrogate models typically demand a large number of samples to construct datasets, resulting in substantial computational and workload burdens. To enhance efficiency, it is necessary to reduce sample size requirements without losing the accuracy. Therefore, few-shot learning methods (Huisman et al., 2021) have been developed, possessing strong generalization capabilities, minimal data requirements, and high learning efficiency. Su et al. (2022) developed a novel DRHRML (data reconstruction hierarchical recurrent meta-learning) method for few-shot learning in intelligent bearing fault detection. Liu et al. (2016) proposed a support vector machine (SVM) gait controller trained with limited samples for stable bipedal robotic walking. In the field of few-shot learning, model-agnostic meta-learning (MAML) learns a highly task-sensitive model initialization, enabling rapid adaptation to new tasks with minimal examples (Fallah et al., 2020). To validate the algorithm selection rationale in this study, a comparative analysis was conducted between MAML and other algorithms. Compared to the Reptile algorithm (Nichol and Schulman, 2018), which employs first-order approximations for improved computational efficiency, its predictive accuracy may be inadequate for com-

plex nonlinear regression problems. In contrast, meta-SGD (Li et al., 2017) introduces learnable learning rates, which increases the risk of overfitting when handling extremely small datasets. Transfer learning requires a large and relevant source task, which is often difficult to find in mechanical engineering. Therefore, this study selects MAML for application in the optimization design of rolling linear guide pairs. With only 126 sample sets, it achieves a prediction accuracy with  $R^2$  exceeding 0.91.

Rolling linear guide pairs serve as core transmission components in precision mechanical equipment. The design and performance metrics must satisfy the stringent requirements of high-end application scenarios. The details are outlined below.

1. *High positioning accuracy and motion smoothness.* High stiffness and low deformation are core requirements for ensuring precision.
2. *High load capacity and long service life.* Controlling the maximum contact stress is the key to ensuring service life.
3. *High reliability and stability.* The design must ensure consistent and predictable performance under various working conditions, preventing performance degradation caused by improper parameter matching.

This paper proposes a meta-learning-based inverted-bottleneck residual fully connected network model. The surrogate models for guide pair stiffness and maximum contact stress were established, and the SLSQP optimization algorithm was employed to solve structural parameter optimization. In addition, this study develops a parametric design platform for guide pairs, which enhances the efficiency of dataset construction through parametric modelling and automated FEM.

The structure of this paper is organized as follows: Sect. 2 establishes the response surface model for predicting the mechanical performance of guide pairs, detailing the few-shot learning process. Section 3 presents the developed parametric modelling platform and couples it with automated FEM, establishing an integrated design–simulation workflow. In Sect. 4, comprehensive analysis of the results is conducted. Finally, Sect. 5 summarizes the study.

## 2 Meta-learning-based inverted-bottleneck residual fully connected network model

Few-shot learning is a machine learning paradigm that achieves efficient learning with limited training samples. This section integrates the inverted-bottleneck structure with a fully connected network and employs the model-agnostic meta-learning (MAML) algorithm to train this model, achieving rapid cross-task adaptation capability.

### 2.1 Enhanced Latin hypercube sampling for input sample set construction

To ensure the coverage of the parameter space, a sampling strategy is introduced in this section. The Latin hypercube sampling (LHS) method is employed to ensure a uniform and representative distribution of input samples within the parameter space. In addition, a Euclidean distance constraint is introduced to prevent samples from being excessively close to each other.

Before conducting LHS, the value ranges of all parameters must be normalized to the interval [0,1]. For a raw parameter  $x_j \in [a_j, b_j]$ , the normalized value  $\bar{x}_j$  is calculated as follows:

$$x_j = \frac{x_j - a_j}{b_j - a_j}. \tag{1}$$

Specifically, the value range of each dimension is uniformly divided into  $N$  equal subintervals. For each dimension  $j$ , a random permutation  $\pi_j$  is generated, where  $\pi_j(i)$  denotes the subinterval number assigned to the  $i$ -th sample in the  $j$ -th dimension. Based on this, the specific value  $X_{ij}$  for each sample point can be obtained through the following calculation:

$$X_{ij} = \frac{\pi_j(i) - u_{ij}}{N}, \tag{2}$$

where  $u_{ij}$  is a random number sampled from the uniform distribution  $U(0,1)$ . Following LHS sampling, the parameters must undergo de-normalization processing, linearly mapping  $X_{ij}$  from [0,1] to the actual parameter range  $[a_j, b_j]$  for each variable:

$$S_{ij} = a_j + (b_j - a_j) \cdot X_{ij}, \tag{3}$$

where  $S_{ij}$  denotes the actual parameter values output by the final Latin hypercube sampling. When adding new samples, a new sample point must not lie too close to any existing samples to avoid spatial clustering and data redundancy. Therefore, when adding new samples, a Euclidean distance (Parhizkar, 2015) is introduced to ensure sample dispersion:

$$D_g(\mathbf{p}, \mathbf{X}) = \min_{k=1, \dots, N} \|\mathbf{p} - \mathbf{x}_k\|_2 > \delta_g, \tag{4}$$

where  $\mathbf{p}$  denotes the new candidate sample,  $\mathbf{X}$  represents the set of existing samples,  $N$  indicates the sample count,  $\|\cdot\|_2$  is the Euclidean norm, and  $\delta_g = 0.5$  denotes the global distance threshold. The global threshold  $\delta_g = 0.5$  was set based on empirical rules commonly used in experimental design. This setting aims to ensure a sufficient minimum Euclidean distance between sample points in the normalized parameter space. It effectively prevents sample clustering while guaranteeing spatial-filling properties. However, it should be noted that, for parameters with smaller value ranges (such as the ball-positioning angle  $\theta$  and the interference fit  $\delta$  in Sect. 4.2, which have small value ranges), a threshold of 0.5 is evidently excessive and inappropriate. Thus, these parameters should be separately assigned a threshold of  $\delta = 0.05$ .

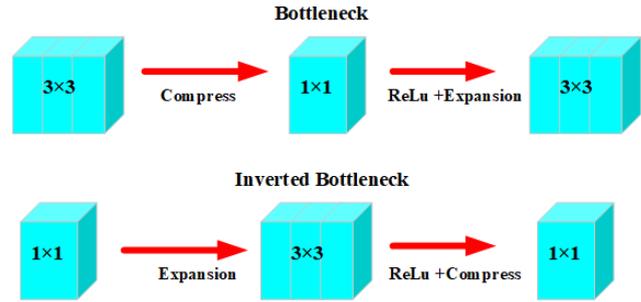


Figure 1. Comparison of dimension transformation with and without inverted-bottleneck structure.

### 2.2 Inverted-bottleneck residual fully connected network model

Deep neural networks (DNNs) are a significant branch of artificial neural networks (Sze et al., 2017). This section utilizes a fully connected deep neural network (FCNN). By integrating an inverted-bottleneck structure and residual connections, a novel deep residual multi-layer perceptron (MLP) architecture is developed. The cascade architecture – comprising an input layer, multiple residual blocks, and an output layer – effectively captures nonlinear mappings between rolling linear guide design parameters and mechanical performance.

The structural design of neural networks is a pivotal factor in determining their performance. In traditional bottleneck architectures (e.g. ResNet’s Bottleneck), a “wide–narrow–wide” pattern is typically adopted. High-dimensional inputs are compressed into a low-dimensional space for feature extraction and then are expanded back to the original high-dimensional space. In contrast, inverted-bottleneck structures (e.g. MobileNetV2) employ the opposite “narrow–wide–narrow” paradigm: low-dimensional inputs are expanded into a high-dimensional space for feature extraction and then return to low dimensionality (Sandler et al., 2018).

Figure 1 illustrates the dimensional transformation schematic of the bottleneck structure and the inverted-bottleneck structure. The inverted-bottleneck structure overcomes the limitations of feature extraction in low-dimensional spaces. This enables the model to possess stronger representational capabilities and capture higher-level features more readily.

In inverted-bottleneck structures, the ReLU activation function is typically positioned after the dimensionality expansion layer and preceding the compression layer. Its primary role is to introduce nonlinear transformations, enabling the model to learn more complex functions. Without the ReLU function, multiple linear transformations would be equivalent to a single linear transformation, severely limiting the model’s representational capacity. Figure 2 depicts the inverted-bottleneck residual fully connected network model (IB-ResFCN) proposed in this study. Table 1 specifies the de-

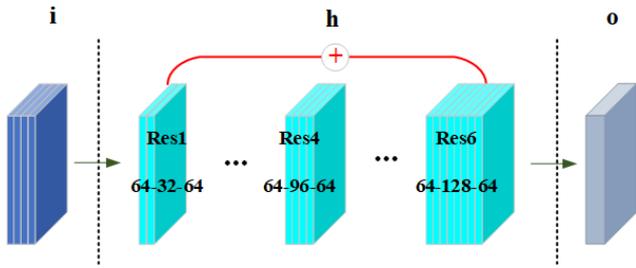


Figure 2. Schematic diagram of the inverted-bottleneck residual fully connected network architecture.

tailed parameters of the model, which consists of one input layer, six residual blocks, and two regression output layers.

Residual connections directly link each residual block and effectively prevent gradient vanishing in MLP. Their mathematical expression is as follows:

$$y = x + \sum_{i=1}^6 F(x), \tag{5}$$

where  $y$  denotes the final output,  $x$  represents the input, and  $F(x)$  signifies the transformation applied to the current input by the network layers.

Residual blocks create a shortcut path, enabling gradients to rapidly back propagate. The first three residual blocks gradually extract higher-dimensional features through progressively increasing dimensions, thereby enhancing the model’s expressive capability. The last three residual blocks employ the inverted-bottleneck structure to achieve dimensional expansion, thereby refining the features.

### 2.3 MAML for supervised learning

Based on the IB-ResFCN model, meta-learning is applied to enhance prediction accuracy. Meta-learning includes three stages: meta-training, meta-validation, and meta-testing. According to the stages, the dataset is divided into the ratio of training set ( $D_{train}$ ): validation set ( $D_{val}$ ): test set ( $D_{test}$ ) = 6 : 2 : 2.

Based on distinct task types, meta-learning can be categorized into supervised meta-learning (SML) and meta-reinforcement learning (Meta-RL). Few-shot learning has been extensively researched in the supervised-task domain. Its objective (Finn et al., 2017) is to learn a new function from a limited number of input–output pairs using prior data from similar tasks through meta-learning. This paper employs the model-agnostic meta-learning (MAML) algorithm applied to SML.

In SML, the training set is divided into  $T$  meta-tasks, each including  $K$  samples. These samples are divided into support and query sets at a 7 : 3 ratio. Figure 3 illustrates the classification method for the training set. The support set is used for the inner loop, and the query set is used for the outer loop.

As shown in Fig. 4, at the beginning of the meta-training, three critical parameters are defined: (1) initial

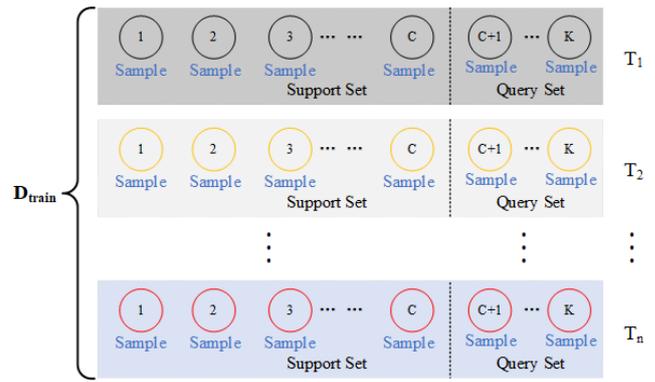


Figure 3. Schematic diagram of training-set partitioning.

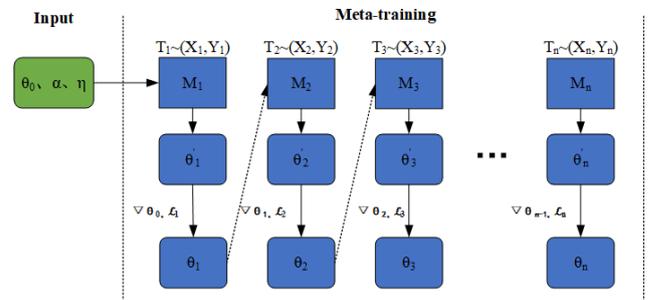


Figure 4. Meta-parameter optimization framework for meta-training process.

meta-parameters  $\theta_0$ , (2) inner-loop learning rate  $\alpha$  and (3) outer-loop learning rate  $\eta$ . Then multiple epochs of meta-training will be conducted. In each epoch, the training set is divided into  $N$  meta-tasks:  $T_i (i = 1, 2, \dots, n) \in D_{train}$ . Each task is assigned a unique meta-learner  $M_i$  and its corresponding initial parameters  $\theta_i$ . By using the corresponding task  $T_i$ , task-specific fast weights  $\theta'_i$  are computed. Subsequently, the gradient  $\nabla$  and loss  $\mathcal{L}$  for the current round are computed. Then, the meta-learner for the next round  $M_{i+1}$  and the parameters  $\theta_{i+1}$  are updated. We repeat the above process until all rounds of meta-training are completed. Finally, the meta-initialization parameters  $\theta_n$  are acquired. Table 2 shows the parameter settings for MAML training.

Considering only a single gradient update step, this section explains how  $\theta_i$  is updated to  $\theta_{i+1}$  in task  $T_{i+1}$ . Based on  $\theta_i$  and the support set data,  $\theta_{i+1}'$  is computed via gradient descent.

$$\theta'_i = \theta_i - \alpha \nabla_{\theta} \mathcal{L}_S(\theta_i) \tag{6}$$

In the above,  $\mathcal{L}_S$  represents the loss function of the support set. The purpose of the inner loop is to rapidly adapt the model parameters of the support set. In the query set, the loss  $\mathcal{L}_Q(\theta'_{i+1})$  is computed using the fast weights  $\theta'_{i+1}$ :

$$\nabla_{\theta} \mathcal{L}_Q(\theta'_{i+1}) = \frac{\delta \mathcal{L}_Q(\theta'_{i+1})}{\delta \theta'_{i+1}} \cdot \frac{\delta \theta'_{i+1}}{\delta \theta_i} \tag{7}$$

**Table 1.** Layer parameters of the fully connected neural network.

Layer	Input dimension	Output dimension	Transformation
Input layer	$i = 4$	$h_0 = 64$	4–64
Residual block (1–3)	$h_i = 64$	$h_0 = 64$	64–32/48/64–64
Residual block (4–6)	$h_i = 64$	$h_0 = 64$	64–96/128/128–64
Output layer	$h_i = 64$	$o_2 = 1$	64–32–1

**Table 2.** MAML parameter settings.

Parameter	Symbol	Value
Inner-loop steps	–	3
Inner-loop learning rate	$\alpha$	0.01
Outer-loop learning rate	$\eta$	0.001
Batch size	$N$	20
Samples per task	$K$	15

Through the chain rule, the relation of  $\theta'_{i+1}$  in  $\theta_i$  is incorporated into the gradient computation. Based on this, we update  $\theta_i$  to  $\theta_{i+1}$ :

$$\theta_{i+1} = \theta_i - \eta \nabla_{\theta} \mathcal{L}_Q(\theta'_{i+1}). \tag{8}$$

The aim of the outer loop is to optimize  $\theta_i$ , ensuring that the generated fast weights  $\theta'_{i+1}$  achieve optimal performance in the query set. After  $N$  task iterations, the final meta-initial parameters  $\theta_n$  can be calculated.

$$\min_{\theta} \sum_{i=1}^n \mathcal{L}_Q(\theta_i) = \sum_{i=1}^n \mathcal{L}_Q(\theta_{i-1} - \alpha \nabla_{\theta} \mathcal{L}_S(\theta_{i-1})) \tag{9}$$

$$\theta_n = \theta_0 \eta \sum_{i=1}^n \nabla_{\theta} \mathcal{L}_Q(\theta'_i) \tag{10}$$

SML can address both classification and regression problems. In regression analysis, a loss function  $\mathcal{L}(\theta)$  is designed to measure the error between predicted outputs and true values. The  $\mathcal{L}(\theta)$  for supervised regression is the mean squared error (MSE), formulated as follows:

$$\mathcal{L}(\theta) = \sum_{x^{(j)}, y^{(j)} \sim T_i} \|f_{\theta}(x^{(j)}) - y^{(j)}\|_2^2, \tag{11}$$

where  $x^{(j)}$  and  $y^{(j)}$  denote the input value and the true value of a sample within  $T_i$ . Algorithm 1 is the MAML algorithm for supervised learning.

### 2.4 Fine-tuning phase

The preceding section explained the algorithm for SML and the process of obtaining  $\theta_n$ . However,  $\theta_n$  cannot serve as a new starting point for rapid task adaptation of the model. Due

**Algorithm 1** MAML for Few-Shot Supervised Learning

```

Input 1:  $T_i = (X_i, Y_i)$ : Meta-training tasks
Input 2:  $\alpha, \eta, \theta'_i, \theta_i$ : Meta-training parameters
1: Randomly initialize  $\theta_0$ 
2: while  $i < N$  do
3:   Sample batch tasks  $T_i \sim D_{train}$ 
4:   for all  $T_i$  do
5:     Draw sample points  $P = \{x^{(0)}, y^{(0)}\}$  from  $T_i$ 
6:     Compute  $\nabla_{\theta_{i-1}} \mathcal{L}_S(\theta'_{i-1})$  using points  $P = \{x^{(0)}, y^{(0)}\}$  and Equation 7
7:     Calculate  $\theta'_i$  via gradient descent:  $\theta'_i = \theta_{i-1} - \alpha \nabla_{\theta} \mathcal{L}_S(\theta_{i-1})$ 
8:     Extract sample points  $P' = \{x^{(0)}, y^{(0)}\}$  from  $T_i$  for meta-update
9:   end for
10:  Update  $\theta_n$  using  $P' = \{x^{(0)}, y^{(0)}\}$  and Equation 11:  $\theta_n \leftarrow \theta_0 - \eta \sum_{i=1}^n \nabla_{\theta} \mathcal{L}_Q(\theta'_i)$ 
11: end while
    
```

to insufficient training samples per meta-task during meta-training, the model fails to adequately learn task-specific patterns. Thus, after meta-training, fine-tuning is performed using the entirety of the training-set samples. This process ensures higher model-learning efficiency. We perform  $K$  training iterations on  $\theta_n$  to update and obtain  $\theta_k$ :

$$\theta_k = \theta_{k-1} \alpha_T \nabla_{\theta} \mathcal{L}_{train}(\theta_{k-1}), \tag{12}$$

where  $\alpha_T$  denotes the learning rate during the fine-tuning phase, and  $\mathcal{L}_{train}$  represents the loss function of the training set. The value of  $\alpha_T$  is set to 1/10 of  $\eta$  since  $\theta_n$  is already situated within a well-performing region. A smaller learning rate ensures stable and precise convergence, which prevents missing of the optimal solution or divergence due to excessively large step sizes.

The parameter  $\theta_k$  is integrated into the model, and the validation loss is computed using validation set samples and Eq. (11). The training process is configured for 100 rounds, with the  $\theta_k$  parameters obtained in each round being evaluated for their loss in the validation set. If the loss decreases, the parameters  $\theta_k$  obtained in that training round are saved as  $\theta_{test}$ . If the loss shows no improvement, the early-stopping mechanism is triggered. When the validation loss ceases to decrease for 10 consecutive training epochs, the training is halted, and the process reverts to the model parameters that achieved the minimum validation loss. The  $\theta_{test}$  is integrated into the model, serving as a new starting point for rapid task adaptation.

Figure 5 illustrates the overall workflow of meta-learning, primarily divided into the training phase and the evaluation

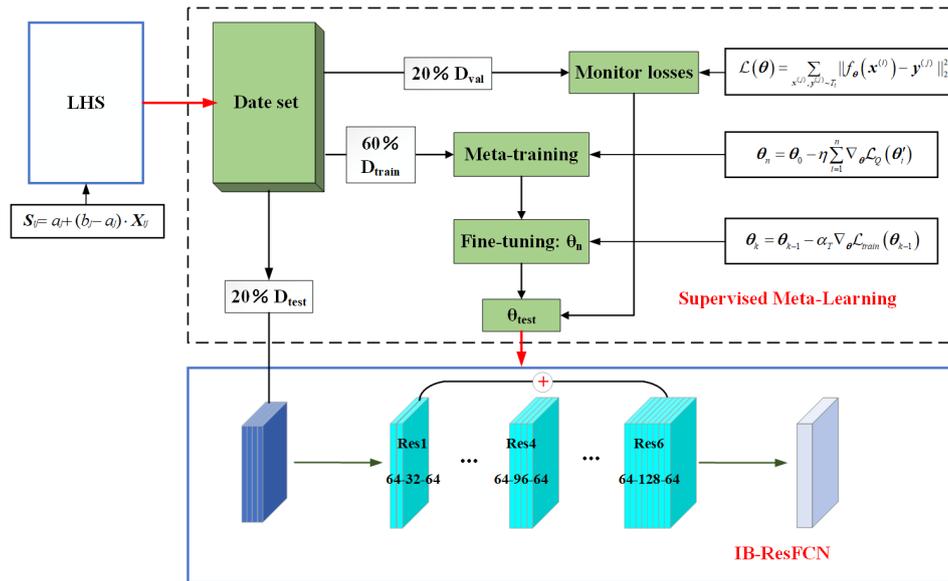


Figure 5. Overall workflow of meta-learning.

phase: (1) first, establish the dataset using LHS; (2) separate the dataset based on distinct functional purposes; (3) start training the IB-ResFCN to obtain  $\theta_{\text{test}}$ ; and (4) assign the learned  $\theta_{\text{test}}$  to the IB-ResFCN and evaluate it based on the test set.

### 3 Development of a parametric assisted design platform

Dataset construction requires modelling and finite-element analysis (FEA) for each sample. Sequential manual modelling for different samples entails a significant workload and low efficiency. Currently, parametric modelling is widely adopted, effectively reducing the time required to generate models. However, existing research rarely couples parametric modelling with FEA software for rapid modelling and analysis. This study develops an integrated platform coupling parametric modelling with FEA, significantly enhancing the efficiency of dataset construction.

The platform was developed within the Visual Studio development environment. Through API integration, the platform achieves direct invocation and collaborative control of SolidWorks and Abaqus, demonstrating robust system integration capabilities and cross-platform compatibility. The platform’s overall architecture follows a “three-tier coupling, closed-loop-driven” technical approach. It comprises three core functional modules: the interaction layer, the logic layer, and the execution layer. The platform seamlessly integrates user input, data processing, and simulation computation, forming a complete closed-loop design–simulation cycle.

The interaction layer serves as the human–machine interface of the platform, implementing a graphical interface based on C# and WinForm technology. Users can intuitively

perform parameter configuration and process control. As illustrated in Fig. 6, the overall interface consists of a parameter input zone, a structure preview zone, and a data management zone. The parameter input zone includes 25 critical fundamental parameters such as ball radius, guide rail dimensions, and mesh size. The structural schematic is embedded centrally, displaying key geometric features and dimension annotations. The data management zone enables custom file path configuration, automated saving strategies, and systematic organization of simulation results.

Building upon parameter acquisition in the interaction layer, the logic layer acts as an intermediary bridge. It maps fundamental user inputs (e.g. ball diameter  $d$ , ball-positioning angle  $\theta$ , initial contact angle  $\beta$ ) into structural feature quantities required for modelling (guide rail dimensions, ball positions, assembly relationships, etc.). This mapping relationship can be mathematically formulated as follows:

$$S = f(P), \tag{13}$$

where  $P = (d, \theta, \beta, \dots)$  denotes the input parameter set;  $S$  represents the structural feature set required for modelling; and the mapping function  $f(\cdot)$  is implemented through function fitting, lookup tables, or polynomial interpolation. To more clearly demonstrate the mapping relationship between  $S$  and  $P$ , the linear guide shown in Fig. 7 is taken as an example, with Table 3 providing specific mappings of key parameters. Through the mapping mechanism in the logic layer, automated modelling of structural-logic construction algorithms driven by dimensional chains is achieved.

The execution layer covers two core processes: 3D modelling and FEA. Upon model completion, we select the FEM option in Fig. 6 to initiate FEA. As shown in Fig. 8, the

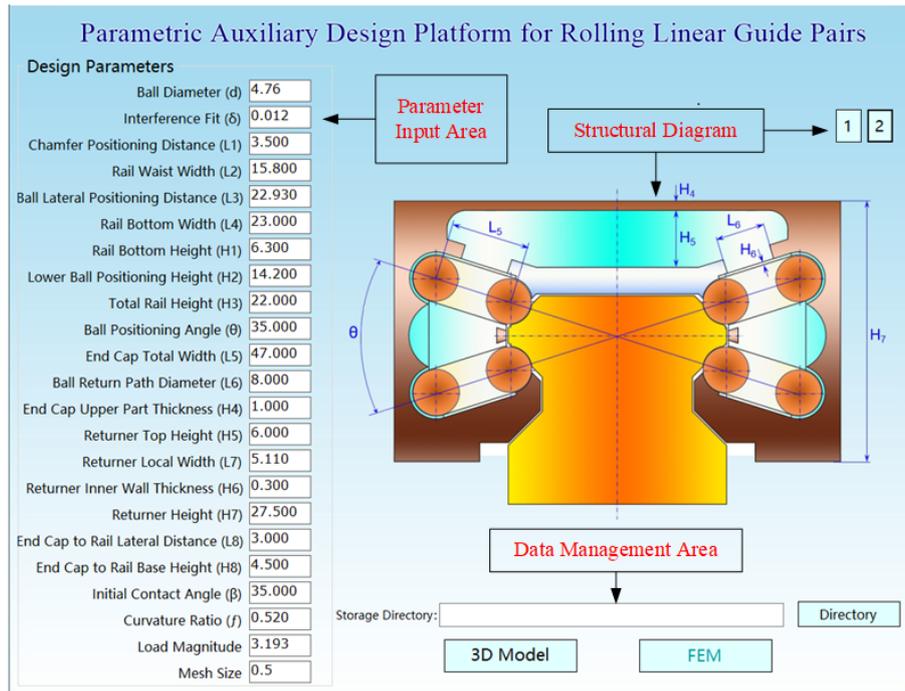


Figure 6. Schematic diagram of the parametric-platform interface. Dimensions in mm, loads in MPa, and mesh units in mm<sup>2</sup>. The structural schematic references Structural Diagram 2 as an example.

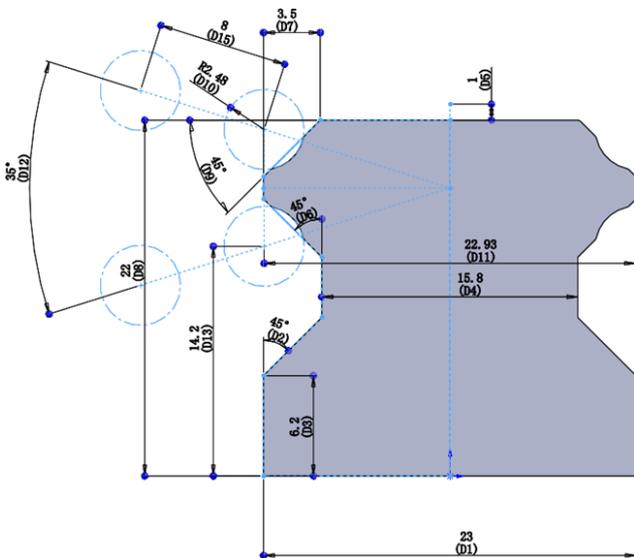


Figure 7. Parameters for parametric modelling of the linear guide.

STEP file exported from SolidWorks retains complete assembly information of the guide rail mechanism. Upon reading the file, the platform automatically imports the model into Abaqus. Subsequently, the pre-configured Python script file (py.) is automatically retrieved. This file contains command streams for preprocessing tasks and job submission modules. The platform automatically controls Abaqus to execute the

Table 3. Mapping relationship between *S* and *P* for the linear guide.

Structural feature	D1	D3	D4	D10	D12	D13
$f(\cdot)$	$8d + 148 - L6$	H1	L2	$d/2$	$\theta$	H2

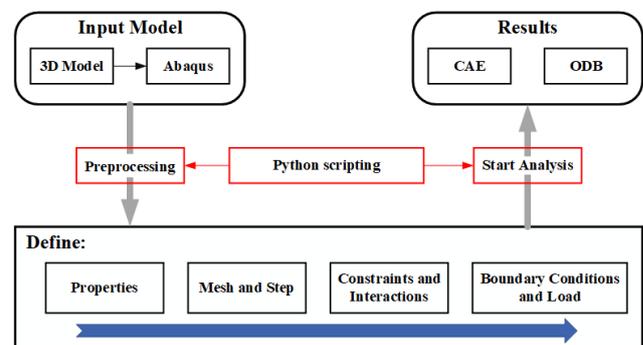


Figure 8. Automated FEA workflow diagram.

complete workflow, including defining properties, meshing, setting boundary conditions, and so on.

Upon completion of the solving process, the platform automatically archives both CAE simulation files and ODB result files. This integrated system achieves rapid modelling

and automated FEA, substantially enhancing the efficiency of dataset construction. As shown in Table 4, an experiment was conducted to evaluate the efficiency and reliability of the parametric platform. This study compared the traditional design method with the parametric-platform design method.

Initially, 100 sets of guide samples were randomly generated for modelling and simulation on the platform. Subsequently, 10 sets of models were randomly selected from the 100 sets of 3D models for manual modelling. The results indicate that, although a very small number of models failed due to parameter interference, the modelling success rate reached 98%. With high reliability, the platform boosts single-iteration efficiency dozens of times over traditional methods.

## 4 Result analysis

### 4.1 Validation of FEA results vs. experimental results

In FEA, many components of the guide rail are non-load-bearing; it is thus necessary to simplify the overall model. The rolling linear guide pair is divided into several independent structural units, with each representing a repetitive structure of the guide assembly. Each unit includes three parts: the guide rail block, the slider block, and the upper- and lower-ball sets.

In the preprocessing module, the elastic modulus and Poisson's ratio for the ball–groove contact are set to  $E = 2.06 \times 10^5$  MPa  $\nu = 0.3$ . For mesh generation, the multi-zone partitioning method is employed to obtain hexahedral meshes, with mesh refinement being applied to the contact regions. Surface-to-surface contact is adopted to ensure proper interaction at the ball–groove interface. A load is applied to the top of the slider, while full constraints are imposed on the bottom of the guide rail.

Vertical stiffness is a critical metric for assessing guide rail pair performance. It quantifies vertical deformation under vertical loads, defined by the following:

$$K_v = \frac{P}{\Delta h_v}, \quad (14)$$

where  $P$  denotes the external load,  $K_v$  represents the approximate vertical stiffness, and  $\Delta h_v$  indicates the deformation magnitude in the vertical direction. For the FEM of an independent structural unit, the schematic diagram of its simulation model is shown in Fig. 9, with the load distribution given by  $P_0 = P/n$ .  $\beta$  denotes the contact angle, defined by the oblique line connecting the curvature centres of the guide groove, the ball, and the slider. Since the groove radii of the guide rail and slider are identical, their vertical deformations are approximately equal. The total vertical deformation is given by  $\Delta h_v = \Delta h_1 + \Delta h_2$ .

Through this method, the vertical deformation of the guide rail pair can be obtained. Substituting this value into Eq. (14) yields the vertical stiffness. However, inherent errors in FEA

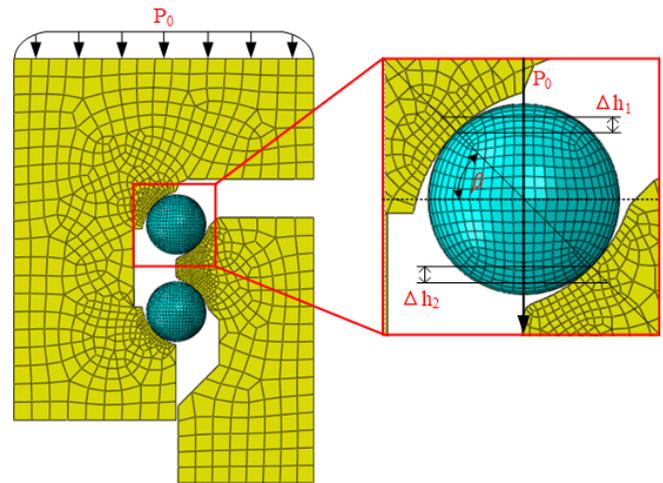


Figure 9. Schematic diagram of the simulation model.

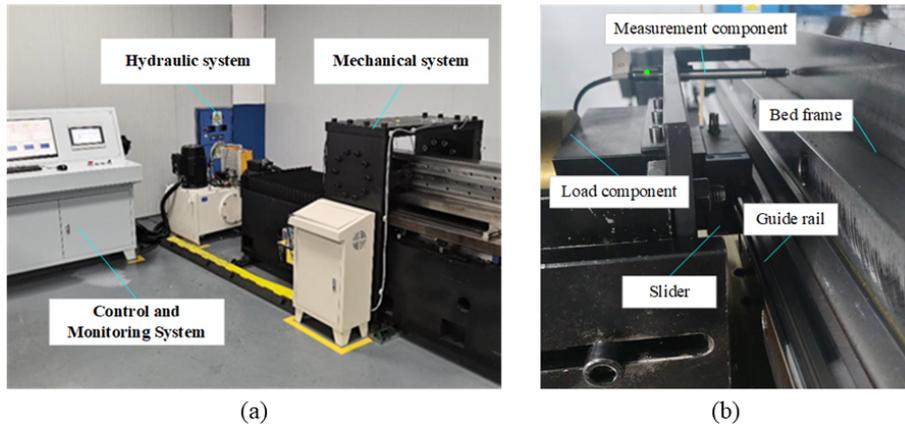
preclude achieving exact vertical-stiffness values, yielding only approximate results. To verify the accuracy of FEA results. This study takes the D45 guide pair as the subject, conducting static experiments to compare the  $\Delta h_v$  from FEA.

The full-parameter test bench for rolling linear guide pairs is illustrated in Fig. 10, primarily composed of a measurement system and a control system, a hydraulic system, and a mechanical system. Figure 10b displays the mechanical system comprising both loading and measurement components. The measurement components primarily include a contact displacement sensor (linearity error  $\leq 0.5\%$  F.S), an S-shaped tension–compression load cell (linearity error  $\leq 0.5\%$  F.S), signal amplifiers, and communication modules. The contact displacement sensor converts internal movement caused by operational accuracy variations into electrical signals. These signals are then transmitted to the control system via an amplifier, a communication module, and RS232 serial communication. The S-shaped tension–compression load cell generates resistance signals in response to external force variations, which are converted into digital communication signals by a signal amplifier and then are transmitted to the control system via RS232 serial communication. Once the signals enter the control system, the acquisition program collects, displays, and saves them.

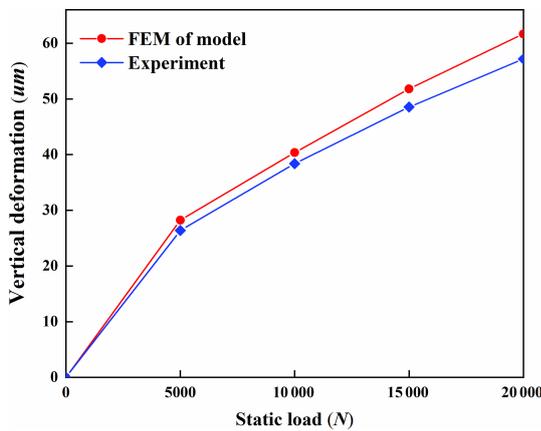
The testing procedure is as follows: install the rolling linear guide on the test bench. Apply load through the hydraulic system, increasing stepwise from 0 to 20 kN in 5 kN increments. First, obtain the force and displacement vectors at each node under the initial load condition. Next, calculate the variation in force and displacement at each measurement point. This variation is relative to the initial load condition during each force-holding period. After averaging the data, the resulting values represent the force and displacement for that time period. Finally, apply the least squares method to fit the force–displacement curves across all time periods, thereby deriving the vertical displacement.

**Table 4.** Performance comparison between traditional design and parametric platform.

Dimensions	Indicators	Traditional design	Parametric platform
Efficiency	Single modelling time	50 min	2–3 min
	CAE import time	5 min	0.5 min
	FEA setup time	25 min	1 min
Reliability	Modelling success rate	Depends on experience	98 %
	FEA setup success rate	Depends on experience	100 %



**Figure 10.** Full-parameter test bench for rolling linear guide pairs. Panel (a) shows the test bench, and panel (b) shows the mechanical system.



**Figure 11.** Comparative results of experimental data and FEM.

The obtained experimental results are compared with FEM, with the comparative outcomes being illustrated in Fig. 11. To quantitatively evaluate the prediction accuracy of the finite-element model, multiple statistical error metrics were calculated based on the aforementioned data, as shown in Table 5.

Results demonstrate a high degree of consistency between FEA and experimental outcomes, confirming the accuracy and validity of the finite-element model. This validated model can be reliably utilized for subsequent research tasks.

**Table 5.** Statistical errors of finite-element model predictions.

Load (kN)	Absolute error (μm)	Relative error (%)
5	1.83	6.94
10	1.98	5.19
15	3.26	6.72
20	4.48	7.83
Statistical values	MAE = 2.89 RMSE = 3.08	MAPE = 6.67

#### 4.2 Influence of structural parameters on mechanical performance

Using the finite-element model in Fig. 8, we analyse the influence of different parameters on the mechanical performance of the guide rail pair. The parameters and their value ranges are as follows:

1. initial contact angle  $\beta$  (30–60°) – complementary angle of the load direction relative to the raceway-transmitted resultant force on rolling elements;
2. curvature ratio  $f$  (0.5–0.6) – the ratio of the groove radius of the guide pair to the ball diameter;

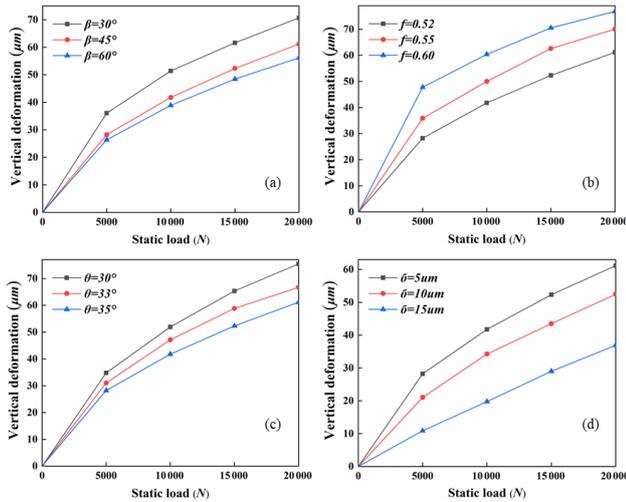


Figure 12. Influence patterns of contact parameters on guide rail pairs.

3. ball-positioning angle  $\theta$  ( $27\text{--}37^\circ$ ) – angle between the upper-left/lower-right raceway centre line and the lower-left/upper-right raceway centre line;
4. interference fit  $\delta$  ( $5\text{--}50\ \mu\text{m}$ ) – the value of interference between the balls and the grooves of the guide rail and slider.

Taking vertical deformation as the research object, FEA is conducted under loads of 5–20 kN, with  $\beta$ ,  $f$ ,  $\theta$ , and  $\delta$  as single variables. The influence of contact parameters on the vertical deformation of guide rail pairs is demonstrated, as detailed in Fig. 12.

Analysis reveals that variations in contact parameters profoundly alter the mechanical performance of rolling linear guide pairs. Furthermore, this study systematically quantified the degree of influence of various parameters on the  $\Delta h_v$  of the guide pair. As shown in Fig. 13, the samples were collected based on LHS. The multi-factor global sensitivity analysis method (Lamboni and Kucherenko, 2021) was employed, systematically analysing the main effects and total effects of each parameter.

The results indicate that the interference fit is the dominant parameter affecting vertical displacement (0.839). The ball-positioning angle and curvature ratio also exert moderate influences, while the contact angle demonstrates a minimal effect ( $< 0.1$ ). Furthermore, the total effects of all four parameters consistently exceed their respective main effects. This finding reveals significant interaction effects among the parameters, where their synergistic interplay collectively governs the vertical displacement. Therefore, the geometric parameters of the contact zone can serve as inputs, with vertical deformation and maximum contact stress as outputs. A neural network with four inputs and one output is constructed as

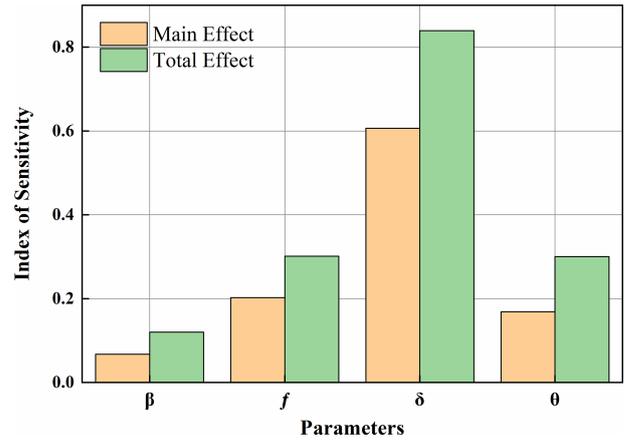


Figure 13. Multi-factor global sensitivity analysis.

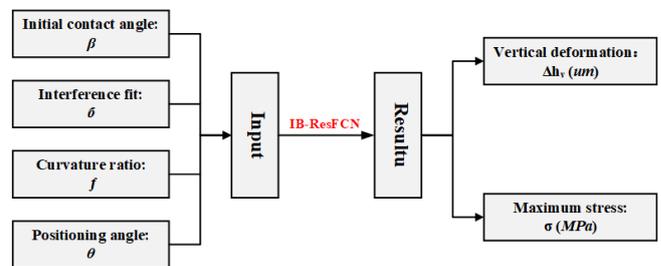


Figure 14. Neural network inputs and outputs for parameter optimization.

shown in Fig. 14. This network is trained to predict mechanical performance.

The prediction models for vertical deformation and contact stress share identical architectures, differing only in terms of parameters such as sample sets, data partitioning, and meta-learning rates.

### 4.3 Comparative analysis of network models

This section validates the structural rationality of the IB-ResFCN model and its prediction accuracy. Comparative validation experiments involving multiple models were conducted.

Taking the vertical deformation prediction model as an example, comparative experiments were designed for three key aspects: models with and without meta-learning, with and without inverted-bottleneck structures, and with and without residual connections. Experiments compared the proposed model against a conventional FCNN. Mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination ( $R^2$ ) serve as evaluation metrics.

Figure 15 presents the results of comparative experiments. To ensure statistical reliability, this study adopts the 5-fold cross-validation method. The complete dataset was randomly partitioned into five similarly sized subsets, with each subset sequentially serving as the test set for performance evaluation.

tion to avoid chance results arising from specific data partitioning. Table 6 presents the statistical results of the 5-fold cross-validation, including the mean values of three metrics for each model. Furthermore, based on the 5-fold cross-validation results, the 95 % confidence intervals for each metric were calculated using the  $t$  distribution (with  $t = 4$ ).

Overall, the improved models show significant advantages over traditional FCNN in terms of both accuracy and stability. In terms of prediction accuracy, the FCNN + IB model achieved the highest coefficient of  $R^2$  among all compared models. Regarding stability, the FCNN + MAML model demonstrates optimal performance in error control, achieving the lowest values in terms of RMSE. Statistically, all improved models show superior performance compared to the baseline model, with their 95 % CIs showing minimal overlap or complete separation. This demonstrates the effectiveness of the model improvements, their statistical significance, and the strong robustness across different data partitions.

Furthermore, this study conducted ablation experiments to investigate the impact of dimensional design in the IB-ResFCNN. Based on the network framework structure described in Sect. 2.2, a systematic comparison was conducted of the model's predictive performance across different dimensions.

As shown in Table 7, dimensional variations directly impact the model's final prediction accuracy. Based on the comparison of three accuracy metrics, the dimensionality (64–32/48/64–64) was determined to be the optimal configuration. This outcome is the prediction result for the vertical deformation model. Figure 16 presents the prediction results of both the vertical deformation and contact stress models.

Furthermore, Fig. 17 illustrates the loss evolution curve during the meta-training phase. After approximately 30 training cycles, the meta-training loss reached a plateau. This phenomenon is related to the designed distribution of meta-tasks: during each meta-iteration, the model processes a batch containing  $N = 20$  distinct tasks, where each task is constructed by randomly sampling  $K = 15$  data points from the training set. These are divided into support sets (10 samples) and query sets (5 samples) for meta-gradient computation. This random sampling strategy ensures the diversity and representativeness of training tasks. The initial rapid decline in loss stems from the model learning a robust set of initialization parameters. The observed plateau suggests successful learning of the underlying task distribution patterns. Additional meta-training iterations provide negligible improvements. With the loss value stabilizing, the model exhibits rapid adaptation ability for tasks from comparable distributions.

#### 4.4 Optimization solving results

The sensitivity analysis in Sect. 4.2 demonstrates how structural parameters affect vertical stiffness. These parameters

interact in complex ways and not through simple linear addition. To address this challenge, the SLSQP algorithm from SciPy's optimization library was implemented. This method (Gill et al., 2005) proves to be highly effective for constrained optimization problems characterized by smooth nonlinear functions in both objective and constraint formulations.

##### 4.4.1 Constraint conditions and convergence criteria

The SLSQP algorithm transforms nonlinear constrained problems into sequential quadratic programming subproblems. This iterative approach efficiently converges to optimal solutions. This approach integrates two classical optimization algorithms: sequential quadratic programming (SQP) and least squares (LS). This algorithm proves to be effective for solving the constrained optimization problem in this study:

$$\begin{aligned} & \underset{x \in R^4}{\text{minimize}} \Delta h_v(\mathbf{x}) \\ & \text{subject to } g(\mathbf{x})[\sigma] - \sigma_{\max}(\mathbf{x}) \geq 0 \\ & \mathbf{x}^l \leq \mathbf{x} \leq \mathbf{x}^u, \mathbf{x}^l = \left[ \frac{\pi}{6}, 5, \frac{1}{2} \frac{27\pi}{180} \right], \mathbf{x}^u = \left[ \frac{\pi}{3}, 50, \frac{3}{5} \frac{37\pi}{180} \right], \end{aligned} \quad (15)$$

where the optimization variable  $x \in R^4$  represents the structural parameters from Sect. 4.2;  $\Delta h_v(\mathbf{x})$  is the minimization objective function;  $g(\mathbf{x})$  denotes the constraint function; and “l” and “u” respectively define the lower and upper bounds of the optimization variables. Within the constraint conditions, the maximum allowable stress is  $[\sigma] = 800$  MPa, and  $\sigma_{\max}(\mathbf{x})$  represents the predicted maximum stress value.

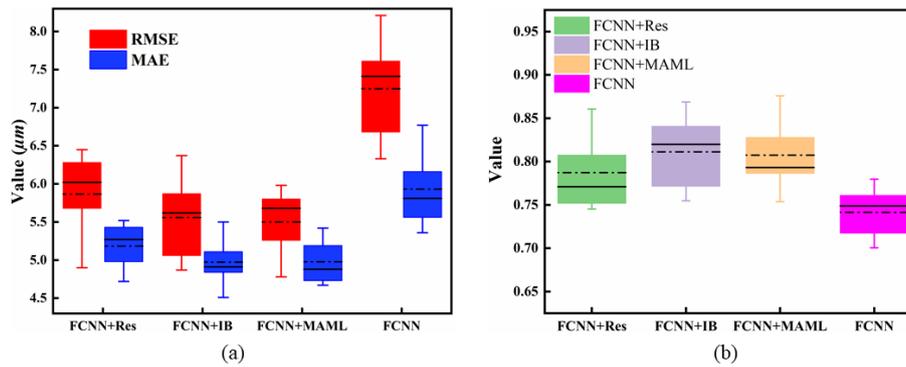
The SLSQP algorithm handles these constraints through an internal Lagrangian formulation. This formulation consistently maintains solution feasibility throughout the optimization process rather than employing an external penalty method. The convergence criteria are specified as follows:

1. function value tolerance –  $\text{ftol} = 1 \times 10^{-6}$  (relative change in objective function);
2. parameter step tolerance –  $\text{xtol} = 1 \times 10^{-6}$  (maximum change in optimization variables);
3. maximum iterations – 300 (computational budget limit).

These complementary tolerances ensure convergence stability. The optimization process terminates when any condition is met. This approach ensures both computational efficiency and high-quality solutions. Figure 18 illustrates the dual-network framework implementing this algorithm.

##### 4.4.2 Multi-start optimization results and sensitivity analysis

To ensure optimization reliability and to mitigate local optima risks, this study implements a comprehensive multi-



**Figure 15.** Comparative experiments. The black lines in both subfigures indicate median values, while the dotted lines represent mean values. Subfigure (a) displays the RMSE and MAE values for each model, and subfigure (b) shows their corresponding  $R^2$  values.

**Table 6.** The 5-fold cross-validation performance metric statistics.

Model	RMSE (95 % CI)	MAE (95 % CI)	$R^2$ (95 % CI)
FCNN	7.248 [6.317, 8.179]	5.932 [5.242, 6.622]	0.741 [0.702, 0.781]
FCNN + Res	5.866 [5.105, 6.627]	5.184 [4.773, 5.595]	0.787 [0.728, 0.846]
FCNN + IB	5.558 [4.802, 6.314]	4.974 [4.521, 5.427]	0.811 [0.752, 0.870]
FCNN + MAML	5.501 [4.902, 6.098]	4.978 [4.582, 5.374]	0.807 [0.749, 0.865]

**Table 7.** Comparison of ablation experiment data.

Res1-Res3	RMSE	MAE	$R^2$
64-32/48/64-64	4.43	3.57	0.916
64-16/32/48-64	5.93	4.50	0.861
64-32/32/32-64	6.89	5.66	0.794
64-48/48/48-64	5.93	4.62	0.846

**Table 8.** Sensitivity analysis of optimized parameters.

Parameter	Mean	S	CV (%)	Remark
$f$	0.526	0.028	5.305	Low sensitivity
$\delta$ ( $\mu\text{m}$ )	18.534	2.923	15.774	Moderately sensitive
$\beta$ (rad)	0.942	0.083	8.77	Low sensitivity
$\theta$ (rad)	0.588	0.023	3.823	Low sensitivity
$\Delta h_v$ ( $\mu\text{m}$ )	11.841	0.935	7.894	Low-sensitivity

start strategy. Starting from 100 randomly generated initial points within the feasible region, the SLSQP algorithm was executed independently for each starting configuration. Figure 19 displays the distribution histogram of vertical deformation values obtained from 100 optimization runs.

Results indicate an optimization interval of [10.54, 14.50], and the success rate is 94%. Over 70% of results exceed the mean value, clustering tightly near the optimum. The optimal solution is reachable from diverse initial conditions. These results further confirm the stability of the optimization algorithm. Additionally, this study conducted a sensitivity analysis of the initial conditions. As shown in Table 8, we calculate the coefficient of variation (CV) for all optimal parameters. The  $\Delta h_v$ ,  $\beta$ ,  $f$ , and  $\theta$  all exhibit CV values below 10%. These are classified as low-sensitivity parameters. The  $\delta$  demonstrates a CV value between 10%–20%, classifying it as moderately sensitive. This necessitates strict tolerance control during actual manufacturing.

For clearer demonstration of performance metric trade-offs across the design space, Fig. 20 shows the distribution

of all feasible solutions within the design space. In the figure, the red line indicates the stress constraint, while the magenta line represents the overall variation trend. The results show a positive correlation between vertical deformation and maximum contact stress. This trend indicates that, under the stress constraint condition, pursuing high stiffness typically requires reducing contact stress. Therefore, optimizing structural parameters improves these two critical performance metrics concurrently.

Table 9 demonstrates that the optimized guide rail pair achieves a 57% reduction in vertical deformation, a significant enhancement in vertical stiffness, and a declining trend in maximum stress values.

To ensure the accuracy of optimization results, the optimized parameters are input into a parametric platform for modelling and simulation validation, with the outcomes illustrated in Fig. 21. Validation shows a mere 2.9% deviation in maximum stress values and only 9.7% variation in extracted vertical deformation. Consequently, the optimization results demonstrate high accuracy.

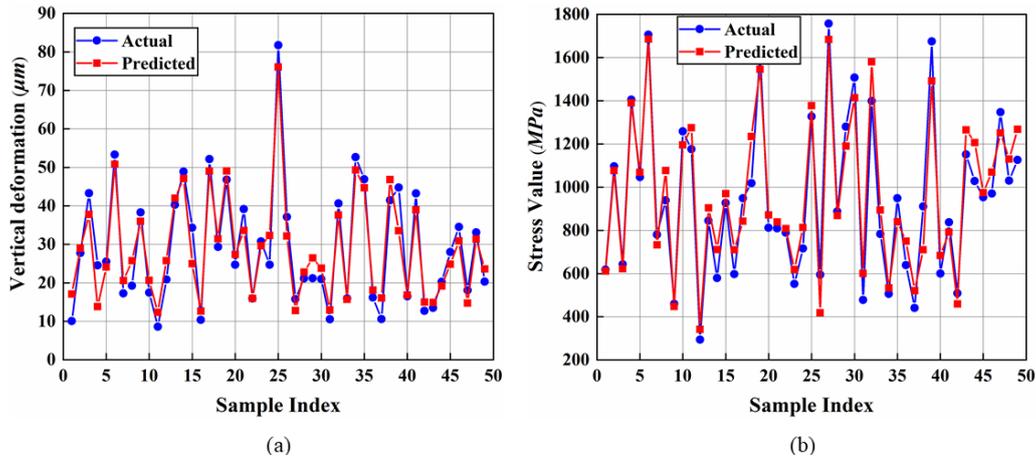


Figure 16. Prediction results. The stress prediction model accuracy metrics are as follows: MAE = 83.42, RMSE = 99.23, and  $R^2 = 0.924$ .

Table 9. Initial parameters vs. optimized parameters.

Parameters	$\beta$ ( $^\circ$ )	$\vartheta$ ( $\mu\text{m}$ )	$f$	$\theta$ ( $^\circ$ )	Stress (Mpa)	Deformation ( $\mu\text{m}$ )
Origin	45	10	0.52	35	452.2	24.67
Optimization	57.11	17.02	0.5108	35.77	446.7	10.54

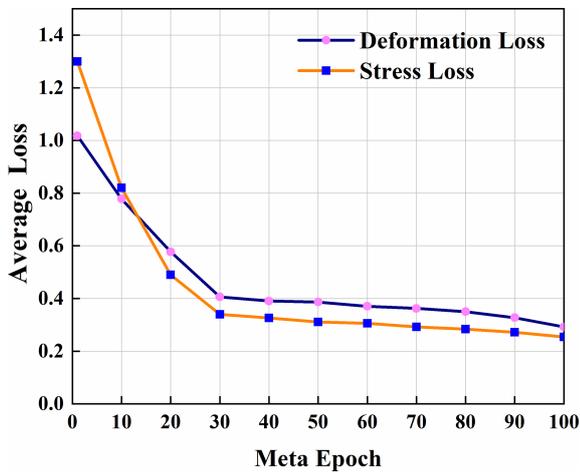


Figure 17. Loss curve during meta-training.

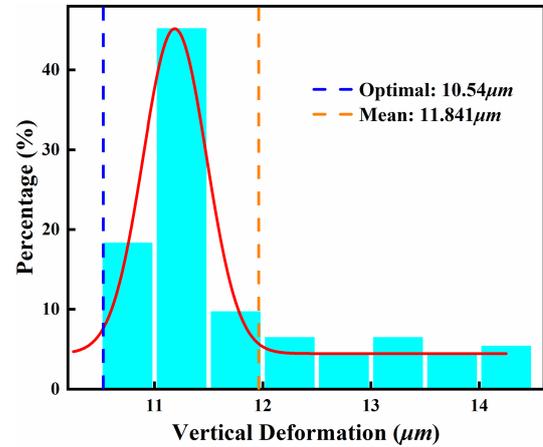


Figure 19. Distribution histogram of vertical deformation values.

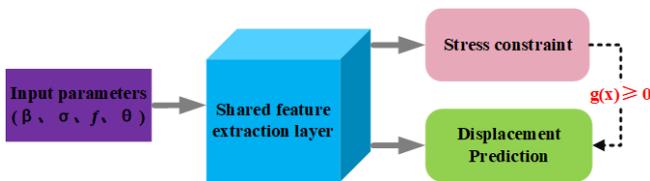
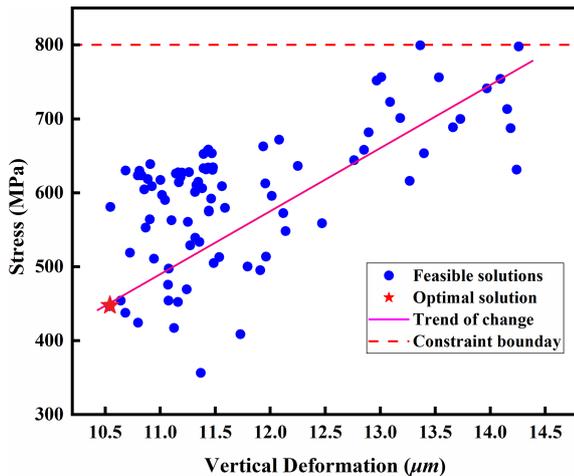


Figure 18. Schematic of dual-network joint architecture.

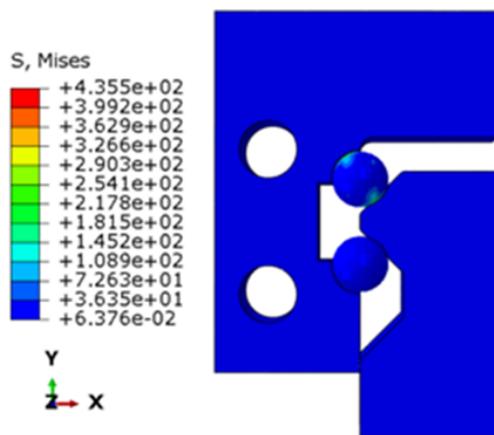
Compared with traditional FEA optimization methods, this study develops a workflow encompassing neural network modelling, dual-network collaborative frameworks, and optimization solving to derive an optimal set of structural parameters.

### 5 Conclusions

This study addresses the multi-objective optimization design problem for rolling linear guide rail pairs. An intelligent optimization framework based on a meta-learning sur-



**Figure 20.** Distribution of feasible solutions in the performance space.



**Figure 21.** Finite-element validation diagram.

rogate model is proposed in this study. This method significantly relieves the over-reliance on costly FEM in traditional design processes and effectively addresses the challenge of constructing high-precision models with limited sample data. This study conducts the following research:

1. A platform integrating parametric modelling and automated FEA was developed. This approach dramatically enhances the efficiency of sample data generation, providing a highly reliable dataset for surrogate models.
2. An IB-ResFCN model was constructed by integrating the inverted-bottleneck residual structure with FCNN. Through a MAML training strategy, the model achieves over 90% prediction accuracy for mechanical performance under few-shot conditions.
3. Based on the constructed surrogate model, the SLSQP algorithm is employed for multi-objective optimization of key structural parameters. Results demonstrate that

the optimized structure achieves significantly enhanced mechanical performance, fully validating the effectiveness and engineering value of the proposed optimization method.

**Code availability.** The code developed for this study is not publicly available due to the confidential nature of the research project.

**Data availability.** The data supporting this study are not publicly available due to the confidential nature of the research project.

**Author contributions.** CS developed the parametric design platform, performed the finite-element analysis and experimental validation, and contributed to the model development and optimization algorithms. WD participated in the platform development, provided methodological guidance, and supervised the project. SL and JH supported funding acquisition and result validation. All of the authors reviewed and approved the final paper.

**Competing interests.** The contact author has declared that none of the authors has any competing interests.

**Disclaimer.** Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. The authors bear the ultimate responsibility for providing appropriate place names. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

**Acknowledgements.** We are grateful for the financial support from the Zhejiang Jianbing & Lingyan Key Technologies R & D Programme (grant no. 2024C01245(SD2)). In addition, this study was also sponsored by the Ningbo Key Research and Development Program (grant no. 2023T013), the 2024 Cixi Key Research and Development Program (grant no. CZ2025012), and the Ningbo Key Technologies R & D Programme (grant no. 2025Z006).

**Financial support.** This research has been supported by the Zhejiang Jianbing & Lingyan Key Technologies R & D Programme (grant no. 2024C01245(SD2)), the Ningbo Key Research and Development Program (grant no. 2023T013), the 2024 Cixi Key Research and Development Program (grant no. CZ2025012) and Ningbo Key Technologies R & D Programme (grant no. 2025Z006).

**Review statement.** This paper was edited by Zhiwei Zhu and reviewed by two anonymous referees.

## References

- Cheng, S. W.: Analysis of non-uniform load distribution and stiffness for a preloaded roller linear motion guide, *Mech. Mach. Theory*, 164, 104407, <https://doi.org/10.1016/j.mechmachtheory.2021.104407>, 2021.
- Fallah, A., Mokhtari, A., and Ozdaglar, A.: On the convergence theory of gradient-based model-agnostic meta-learning algorithms, *arXiv [preprint]*, <https://doi.org/10.48550/arXiv.1908.10400>, 2020.
- Finn, C., Abbeel, P., and Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks, in: *Proceedings of the 34th International Conference on Machine Learning*, edited by: Precup, D. and Teh, Y. W., PMLR, 70, 1126–1135, <https://doi.org/10.48550/arXiv.1703.03400>, 2017.
- Fishwick, P. A.: Neural network models in simulation: a comparison with traditional modeling approaches, in: *Proceedings of the 21st conference on Winter simulation*, ACM, New York, NY, USA, 702–709, <https://doi.org/10.1145/76738.76828>, 1989.
- Gill, P. E., Murray, W., and Saunders, M. A.: SNOPT: An SQP algorithm for large-scale constrained optimization, *SIAM Rev.*, 47, 99–131, <https://doi.org/10.1137/S0036144504446096>, 2005.
- Huisman, M., van Rijn, J. N., and Plaat, A.: A survey of deep meta-learning, *Artif. Intell. Rev.*, 54, 4483–4541, <https://doi.org/10.1007/s10462-021-10004-4>, 2021.
- Jiang, Y., Cao, Q., Liu, R., Wu, W. T., He, Y., and Yan, H.: Rapid optimization of rocket projectile tailfins based on multi-fidelity neural networks with transfer learning, *Comput. Struct.*, 315, 107801, <https://doi.org/10.1016/j.compstruc.2025.107801>, 2025.
- Kazemi, F., Çiftçioğlu, A. Ö., Shafighfard, T., Asgarkhani, N., and Jankowski, R.: RAGN-R: a multi-subject ensemble machine-learning method for estimating mechanical properties of advanced structural materials, *Comput. Struct.*, 308, 107657, <https://doi.org/10.1016/j.compstruc.2024.107657>, 2025.
- Lamboni, M. and Kucherenko, S.: Multivariate sensitivity analysis and derivative-based global sensitivity measures with dependent variables, *Reliab. Eng. Syst. Safe.*, 212, 107519, <https://doi.org/10.1016/j.res.2021.107519>, 2021.
- Li, Z., Zhou, F., Chen, F., and Li, H.: Meta-SGD: learning to learn quickly for few-shot learning, *arXiv [preprint]*, *arXiv:1707.09835*, <https://doi.org/10.48550/arXiv.1707.09835>, 2017.
- Liu, W. J., Zhang, S., Lin, J., Jiang, S. N., Chen, Z., Zhu, H., and Chen, G.: Investigation of static characteristics for linear rolling guide with considering geometric errors, *Tribol. Int.*, 187, 108698, <https://doi.org/10.1016/j.triboint.2023.108698>, 2023.
- Liu, Z., Wang, L., Zhang, Y., and Chen, C. L. P.: A SVM controller for the stable walking of biped robots based on small sample sizes, *Appl. Soft Comput.*, 38, 738–753, <https://doi.org/10.1016/j.asoc.2015.10.029>, 2016.
- Nagendra, S., Staubach, J. B., Suydam, A. J., Ghunakikar, S. J., and Akula, V. R. K.: Optimal rapid multidisciplinary response networks: RAPIDDISK, *Struct. Multidiscip. O.*, 29, 213–231, <https://doi.org/10.1007/s00158-004-0472-2>, 2005.
- Nichol, A. and Schulman, J.: Reptile: a scalable meta-learning algorithm, *arXiv [preprint]*, *arXiv:1803.02999*, <https://doi.org/10.48550/arXiv.1803.02999>, 2018.
- Ohta, H. and Hayashi, E.: Sound of linear guideway type recirculating linear ball bearings, *J. Sound Vib.*, 235, 847–861, <https://doi.org/10.1006/jsvi.2000.2950>, 2000.
- Parhizkar, R.: Euclidean distance matrices: properties, algorithms and applications, *IEEE Signal Proc. Mag.*, 32, 12–30, <https://doi.org/10.1109/MSP.2015.2398954>, 2015.
- Phuyal, S., Bista, D., and Bista, R.: Challenges, opportunities and future directions of smart manufacturing: a state of art review, *Sustainable Futures*, 2, 100023, <https://doi.org/10.1016/j.sft.2020.100023>, 2020.
- Quan, L. and Zhao, W.: A review on positioning uncertainty in motion control for machine tool feed drives, *Precis. Eng.*, <https://doi.org/10.1016/j.precisioneng.2024.03.003>, 2024.
- Ramu, P., Thananjayan, P., Acar, E., Bayrak, G., Park, J. W., and Lee, I.: A survey of machine learning techniques in structural and multidisciplinary optimization, *Struct. Multidiscip. O.*, 65, 1–31, <https://doi.org/10.1007/s00158-022-03369-9>, 2022.
- Roy, A., Manna, R., and Chakraborty, S.: Support vector regression based metamodeling for structural reliability analysis, *Probabilist. Eng. Mech.*, 55, 78–89, <https://doi.org/10.1016/j.probengmech.2018.11.001>, 2019.
- Sahoo, S. and Lo, C. Y.: Smart manufacturing powered by recent technological advancements: a review, *J. Manuf. Syst.*, 64, 236–250, <https://doi.org/10.1016/j.jmsy.2022.06.008>, 2022.
- Shimizu, S.: Stiffness analysis of linear motion rolling guide, *J. Jpn. Soc. Precis. Eng.*, 64, 1573–1576, 1998.
- Staroszyk, D., Müller, J., and Ihlenfeldt, S.: Increasing the service life prediction accuracy of linear guideways considering real operating conditions, *Mech. Mach. Theory*, 201, 105734, <https://doi.org/10.2139/ssrn.4806567>, 2024.
- Su, H., Xiang, L., Hu, A., Xu, Y., and Yang, X.: A novel method based on meta-learning for bearing fault diagnosis with small sample learning under different working conditions, *Mech. Syst. Signal Pr.*, 169, 108765, <https://doi.org/10.1016/j.ymssp.2021.108765>, 2022.
- Sun, W., Kong, X., Wang, B., and Li, X.: Statics modeling and analysis of linear rolling guideway considering rolling balls contact, *P. I. Mech. Eng. C-J. Mech.*, 229, 168–179, <https://doi.org/10.1177/0954406214531943>, 2015.
- Sze, V., Chen, Y. H., Yang, T. J., and Emer, J. S.: Efficient processing of deep neural networks: a tutorial and survey, *P. IEEE*, 105, 2295–2329, <https://doi.org/10.1109/JPROC.2017.2761740>, 2017.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. C.: Mobilenetv2: inverted residuals and linear bottlenecks, in: *Proceedings of the IEEE Conference on CVPR*, 4510–4520, <https://doi.org/10.1109/CVPR.2018.00474>, 2018.
- Tong, V. C., Khim, G., Hong, S. W., and Park, C. H.: Construction and validation of a theoretical model of the stiffness matrix of a linear ball guide with consideration of carriage flexibility, *Mech. Mach. Theory*, 140, 123–143, <https://doi.org/10.1016/j.mechmachtheory.2019.05.021>, 2019.
- Tong, V. C., Khim, G., Park, C. H., and Hong, S. W.: Linear ball guide design optimization considering stiffness, friction force, and basic dynamic load rating using particle swarm optimization, *J. Mech. Sci. Technol.*, 34, 1313–1323, 2020.
- Wang, L., Kolios, A., Nishino, T., Delafin, P.-L., and Bird, T.: Structural optimisation of vertical-axis wind turbine composite blades based on finite element analysis

- and genetic algorithm, *Compos. Struct.*, 153, 123–138, <https://doi.org/10.1016/j.compstruct.2016.06.003>, 2016.
- Wang, W., Li, C., Zhou, Y., Wang, H., and Zhang, Y.: Nonlinear dynamic analysis for machine tool table system mounted on linear guides, *Nonlinear Dynam.*, 94, 2033–2045, <https://doi.org/10.1007/s11071-018-4473-x>, 2018.
- Wei, J., Li, C., and Ma, Y.: Finite element model for static characteristic analysis of rolling linear guide, *P. I. Mech. Eng. C-J. Mech.*, 236, 1721–1732, <https://doi.org/10.1177/09544062211021443>, 2022.
- Xu, M., Zhang, H., Li, C., Yao, G., Zhang, Y., and Gao, Y.: An analytical nonlinear dynamic model for linear guide with carriage flexibility, *Int. J. Nonlin. Mech.*, 148, 104251, <https://doi.org/10.1016/j.ijnonlinmec.2022.104251>, 2023.
- Yang, H., Wang, Z., Zhang, T., and Du, F.: A review on vibration analysis and control of machine tool feed drive systems, *Int. J. Adv. Manuf. Tech.*, 107, 503–525, <https://doi.org/10.1007/s00170-020-05041-2>, 2020.
- Zha, J., Cheng, K., Xue, F., and Lu, Y.: Hydrostatic guideways for precision machines: the state-of-the-art and future perspectives, *Tribol. Int.*, 110060, <https://doi.org/10.1016/j.triboint.2024.110060>, 2024.
- Zou, H. T. and Wang, B. L.: Investigation of the contact stiffness variation of linear rolling guides due to the effects of friction and wear during operation, *Tribol. Int.*, 92, 472–484, <https://doi.org/10.1016/j.triboint.2015.07.005>, 2015.