Robotica (2026), **44**, pp. 612–629
doi:10.1017/S0263574726103221

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

A gray-box identification method for cable-driven parallel robots via wavelet-enhanced neural networks

Runze Wang[1], Yangmin Li[1] and Yi Zeng[2]

[1]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University Faculty of Engineering, Hong Kong
[2]Key Laboratory of Autonomous Intelligent Unmanned Systems, Harbin Institute of Technology, China
Corresponding author: Yangmin Li; Email: yangmin.li@polyu.edu.hk

Received: 2 September 2025; **Revised:** 11 November 2025; **Accepted:** 11 December 2025;
First published online: 24 March 2026

Keywords: cable-driven parallel robots; identification method; TCN-Transformer; wavelet

Abstract

Based on an in-depth study of the strong nonlinear coupling problem in the identification of cable-driven parallel robot systems, this paper proposes an intelligent identification framework that integrates wavelet transforms, Temporal Convolution Network (TCN), and Transformer. This approach first constructs a TCN-Transformer hybrid network architecture to achieve high-precision trajectory prediction and generate virtual data for identification methods to reduce experimental costs. It then separates the high- and low-frequency components of the system's dynamic characteristics through wavelet multi-scale decomposition. A gray-box identification model is constructed by combining a genetic algorithm with the TCN-Transformer network, enhancing the high-frequency characterization capability while preserving the physical mechanism. Finally, the gray-box identification method identifies the system parameters according to the virtual trajectory generated by the TCN transformer. Experimental verification demonstrates that the proposed method effectively overcomes the shortcomings of traditional identification methods, which suffer from insufficient nonlinear modeling and weak physical interpretability of data-driven methods.

1. Introduce

Cable-driven parallel robots (CDPRs), as the name implies, are special robots that use flexible cables to connect a platform or gripper to a base, replacing traditional rigid arms. Because cables replace rigid rods, CDPRs have less inertia and a higher load ratio than traditional robots, which means they can achieve higher accelerations and speeds [1]. Currently, there are various types of CDPR structures, including classic CDPR with a fixed base [2–7], mobile CDPR with an unmanned vehicle as the base [8], mobile CDPR with drones as the base [9, 10], and some CDPRs for special scenarios [11–14]. The most notable example is the cable-driven structure of the Five-Hundred-Meter Aperture Spherical Radio Telescope (FAST) in southwestern China, which is used to move the telescope's feed module [11].

Although CDPRs have many advantages, their modeling difficulty is much higher than that of traditional rigid robots due to the use of flexible ropes. It not only includes the flexible nonlinearity and multi-variable coupling of the parallel robot, but it also includes the uncertainty and nonlinearity caused by the sag and stretching of the ropes [15]. In order to establish a more accurate model, researchers have also conducted in-depth research on nonlinear friction models [6], error transmission models [5], error compensation [16], and measurement confusion identification [17].

Even if the theoretical model is perfect, inaccurate model parameters caused by installation errors, rope deformation, etc. will still lead to a poor control effect of CDPR. Most literature often assumes that the rope length remains unchanged, ignores the adverse effects of parameter changes, and seeks to use control algorithms with good robustness to suppress the impact of model uncertainty. Li *et al.* [7] proposed an adaptive integral sliding mode collaborative control algorithm, which effectively improved the robustness of the system. Lu *et al.* [18] designed a reinforcement learning algorithm based on the

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

Lyapunov function and Markov chain, which effectively suppressed the adverse effects of uncertainties such as cable elasticity and mechanical friction. Although these control methods have strong robustness, the inaccuracy of system parameters still affects their accuracy.

At present, the least squares method (LSM) and frequency domain analysis methods are widely used in various system identification works, including drone gimbals, multi-axis rotating platforms, and, of course, CDPRs. The previous three studies used the nonlinear least squares method to fit the system parameters of CDPR, using the system input and output information to obtain the system parameters through an iterative algorithm [1, 19, 20]. Kraus *et al.* [21] used frequency domain analysis theory to successfully establish a second-order model with dead zone characteristics and proposed a stability analysis method. Sun [14] identified the dynamic model of CDPR through vibration tests. Wang [4] established an analytical compact model of CGM, deriving the identification formula of the closed Jacobian matrix and its derivatives.

Although these research results have shown that the algorithm can accurately and effectively identify parameters, the noise contained in the information and the disturbances caused by the unmodeled components still seriously affect its accuracy. With the continuous advancement and improvement of intelligent algorithms, neural network algorithms have been widely used in many fields, especially in curve fitting and system identification, with outstanding advantages such as high-dimensional feature processing capabilities and nonlinear fitting capabilities. Zare [22] used the Multilayer Perceptron (MLP), LoLiMoT, and Radial Basis Function (RBF) to establish the black box model of CDPR and compared the results, finding that the various indicators of MLP were better than those of the other two networks. Li [23] uses an adaptive RBF network to establish a friction black box model of CDPR and effectively suppresses the jitter phenomenon of the controlled object. Piao [24] uses ANN to establish a black box model to estimate the end effector force in Cartesian space to achieve precise force control and continuous uncertainty compensation. Even though these methods are effective, they still need improvements. Traditional methods not only rely too much on the physical model information of the object but also can only model the low-frequency information of the system and cannot handle the high-frequency part well. Although newly proposed neural network methods are able to improve accuracy, the black box model loses its generalization ability due to the lack of physical information.

When dealing with noisy signals, the commonly used Fast Fourier Transform (FFT) method has clear drawbacks. Firstly, it only provides an overall frequency view, making it hard to pinpoint when short-lived events occur. Secondly, spectral leakage can occur, creating false frequency components and raising errors in reconstructed signals [25]. To reduce spectral leakage, researchers have used window functions and interpolation with FFT, but this is merely a fix that balances frequency resolution and leakage control [26]. In comparison, the wavelet transform uses multi-resolution analysis to precisely locate non-stationary signal parts in time and frequency, and it shows strong noise resistance [27]. Additionally, using local wavelet functions that decay quickly and focus energy can avoid this issue from the start [28]. Wavelet-enhanced neural networks have gained attention due to their high performance. Reference [30] used wavelet neural networks to study human lower limb motion and obtain desired paths for patients. Reference [29] introduced a robust adaptive control method for dynamic structured fuzzy wavelet neural networks (FWNN). Reference [30] applied auto-regressive wavelet neural networks to n-link robots. However, even though wavelet-enhanced RNNs have worked well in robot sensing and dynamics, an observation of current research shows that this promising method is mostly untested for cable-driven parallel robots.

According to the research presented above, this paper plans to propose an identification algorithm based on the wavelet method and TCN-Transformer. Firstly, based on the existing results, the CDPR model is improved to separate the parameters that will be identified from the strongly coupled equations. Then, in accordance with the characteristics of TCN and Transformer, a TCN-Transformer network architecture is constructed. The new system identification algorithm proposed in this paper is formed by combining the network with wavelets and a genetic algorithm (GA). Finally, the effectiveness, rapidity, and high precision of the proposed TCN-Transformer are verified using real experimental data. The main contributions of this paper include:

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

1. Separating the parameters corresponding to each axis from the highly coupled CDPR dynamic model makes the identification process simpler and more efficient;
2. For end-effector trajectory prediction, the TCN is integrated into the Transformer architecture, forming a TCN-Transformer architecture. This improves the network's prediction accuracy in small-sample learning environments and reduces training costs;
3. Introducing virtual trajectories generated by the TCN-Transformer algorithm into the identification process significantly shortens the data collection cycle while maintaining identification accuracy. A perplexity metric is also proposed to measure the quality of the predicted trajectory;
4. By combining wavelet theory with the GA and the proposed TCN-Transformer, a gray-box model for CDPR is successfully established, addressing the problem that traditional models cannot describe the high-frequency characteristics of the system.

The other sections of this paper are arranged as follows: Section 2 mainly describes the mathematical model and parameter separation method of CDPR; Section 3 proposes the TCN-Transformer network architecture; Section 4 constructs the identification method based on wavelet theory and TCN-Transformer; and Sections 5 and 6 present the experimental results and conclusions of the proposed methods, respectively.

2. CDPR dynamic modeling

In this Section, we will establish a dynamic model of a 3-DOF CDPR controlled by n ropes based on the general structure of CDPR. According to Lu *et al.* [18], the dynamic model of CDPR can be described as:

$$\begin{aligned}&\left(\boldsymbol{M}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{I}_m\boldsymbol{R}^{-1}\boldsymbol{J}\right)\ddot{\boldsymbol{x}}+\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{I}_m\boldsymbol{R}^{-1}\dot{\boldsymbol{J}}+\boldsymbol{C}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{F}_v\boldsymbol{R}^{-1}\boldsymbol{J}\right)\dot{\boldsymbol{x}}\\&\quad+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}_T^{-1}\boldsymbol{F}_c\mathrm{sign}(\dot{l})+\boldsymbol{G}+\boldsymbol{d}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{u},\end{aligned}\tag{1}$$

in which, $\boldsymbol{x}$ denotes the system state vector $\left[x,y,z,\alpha,\beta,\theta\right]^{\top}$, representing the absolute position and orientation of the mobile platform. The input vector $\boldsymbol{u}$ corresponds to the motor torques. The matrix $\boldsymbol{M}$ is the system inertia matrix. Assuming the platform possesses a uniform mass distribution and geometric symmetry, $\boldsymbol{M}$ is a positive definite symmetric matrix. The term $\boldsymbol{R}_T$ denotes the constant transmission ratio relating the motor angle to the rope length. $\boldsymbol{I}_M$ is the inertia matrix of the winch assembly, and $\boldsymbol{F}_v$ is the equivalent viscous damping matrix. $\boldsymbol{C}$ and $\boldsymbol{G}$ represent the Coriolis/centripetal matrix and the gravitational vector, respectively. Finally, the Jacobian matrix $\boldsymbol{J}$ is given by:

$$\boldsymbol{J}=\begin{bmatrix}\boldsymbol{k}_1^{\mathrm{T}}&\left(R_O^P\boldsymbol{a}_1\times\boldsymbol{k}_1^{\mathrm{T}}\right)^{\mathrm{T}}\\ \vdots&\vdots\\ \boldsymbol{k}_n^{\mathrm{T}}&\left(R_O^P\boldsymbol{a}_1\times\boldsymbol{k}_n^{\mathrm{T}}\right)^{\mathrm{T}}\end{bmatrix}_{n\times 6},\tag{2}$$

in which, R_O^P represents the rotation matrix between the absolute coordinate system and the body coordinate system. $\boldsymbol{k}_i,(i=1,2,\ldots,n)$ stands for the unit vector along the i-th cable direction, while $\boldsymbol{a}_i,(i=1,2,\ldots,n)$ represents the position vector of the connection point to the center in the body coordinate system. To simplify the identification process, this study implements the following transformation on (1). Since the collected experimental data already contain the system state variable $\boldsymbol{x}$ as well as its derivatives of various orders, the control value $\boldsymbol{u}$, the length of each rope, the mass matrix $\boldsymbol{M}$ and the proportional coefficient matrix $\boldsymbol{R}$ have been determined experimentally. $\boldsymbol{M}\ddot{\boldsymbol{x}}$ and $\boldsymbol{G}$ are considered constants, leading to the following mathematical treatment:

$$\begin{aligned}&\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{I}_m\boldsymbol{R}^{-1}\boldsymbol{J}\ddot{\boldsymbol{x}}+\left(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{I}_m\boldsymbol{R}^{-1}\dot{\boldsymbol{J}}+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{F}_v\boldsymbol{R}^{-1}\boldsymbol{J}\right)\dot{\boldsymbol{x}}\\&\quad+\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}_T^{-1}\boldsymbol{F}_c\mathrm{sign}(\dot{l})+\boldsymbol{d}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{R}^{-1}\boldsymbol{u}-\boldsymbol{M}\ddot{\boldsymbol{x}}-\boldsymbol{C}\dot{\boldsymbol{x}}-\boldsymbol{G},\end{aligned}\tag{3}$$

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

By multiplying the transformation matrix $\boldsymbol{R}\left(\boldsymbol{J}^{\mathrm{T}}\right)^{-1}$ on both sides of Eq. (3), a new equation will be created:

$$\boldsymbol{I}_m\boldsymbol{R}^{-1}\boldsymbol{J}\ddot{\boldsymbol{x}}+\left(\boldsymbol{I}_m\boldsymbol{R}^{-1}\dot{\boldsymbol{J}}+\boldsymbol{F}_v\boldsymbol{R}^{-1}\boldsymbol{J}\right)\dot{\boldsymbol{x}}+\boldsymbol{F}_c\mathrm{sign}(\dot{l})+\boldsymbol{d}=\boldsymbol{u}-\boldsymbol{R}\left(\boldsymbol{J}^{\mathrm{T}}\right)^{-1}\boldsymbol{M}\ddot{\boldsymbol{x}}-\boldsymbol{R}\left(\boldsymbol{J}^{\mathrm{T}}\right)^{-1}\boldsymbol{C}\dot{\boldsymbol{x}}-\boldsymbol{R}\left(\boldsymbol{J}^{\mathrm{T}}\right)^{-1}\boldsymbol{G}, \tag{4}$$

In view of the fact that the sign function sign(·) is not differentiable at the origin, this study uses the hyperbolic tangent function tanh to construct its smooth approximation. To this end, we define a differentiable alternative function si-tanh, whose mathematical expression is as follows:

$$\text{si-tanh}(x)=\tanh(kx),$$

in which k is a preset constant coefficient.

Now we define that:

$$\boldsymbol{U}^*=\boldsymbol{u}-\boldsymbol{R}\left(\boldsymbol{J}^{\mathrm{T}}\right)^{-1}\boldsymbol{M}\ddot{\boldsymbol{x}}-\boldsymbol{R}\left(\boldsymbol{J}^{\mathrm{T}}\right)^{-1}\boldsymbol{C}\dot{\boldsymbol{x}}-\boldsymbol{R}\left(\boldsymbol{J}^{\mathrm{T}}\right)^{-1}\boldsymbol{G},$$
$$\boldsymbol{T}_1=\boldsymbol{R}^{-1}\boldsymbol{J},\boldsymbol{T}_2=\boldsymbol{R}^{-1}\dot{\boldsymbol{J}},\boldsymbol{T}_3=\text{si-tanh}(\dot{l}),$$

(4) will be rewritten as follows:

$$\boldsymbol{U}^*=\boldsymbol{I}_m\boldsymbol{T}_1\ddot{\boldsymbol{x}}+\left(\boldsymbol{I}_m\boldsymbol{T}_2+\boldsymbol{F}_v\boldsymbol{T}_1\right)\dot{\boldsymbol{x}}+\boldsymbol{F}_c\boldsymbol{T}_3+\boldsymbol{d} \tag{5}$$

Each parameter matrix in (5) is defined as follows:

$$\boldsymbol{U}^*=\left[U_1^*\ U_2^*\ U_3^*\right]^{\mathrm{T}},\ddot{\boldsymbol{x}}=\left[\ddot{x}_1\ \ddot{x}_2\ \ddot{x}_3\right]^{\mathrm{T}},$$
$$\dot{\boldsymbol{x}}=\left[\dot{x}_1\ \dot{x}_2\ \dot{x}_3\right]^{\mathrm{T}},\boldsymbol{I}_m=\mathrm{diag}(I_{m1},I_{m2},I_{m3}),$$
$$\boldsymbol{F}_v=\mathrm{diag}(F_{v1},F_{v2},F_{v3}),\boldsymbol{F}_c=\mathrm{diag}(F_{c1},F_{c2},F_{c3}),$$
$$\boldsymbol{T}_i=\begin{bmatrix}T_{i11}&T_{i12}&T_{i13}\\T_{i21}&T_{i22}&T_{i23}\\T_{i31}&T_{i32}&T_{i33}\end{bmatrix}(i=1,2,3),\boldsymbol{d}=\left[d_1\ d_2\ d_3\right]^{\mathrm{T}}.$$

Substituting the above equation into (5), (6) will be obtained as follows:

$$U_i^*=\left(\sum_{j=1}^{3}T_{1ij}\ddot{x}_j+T_{2ij}\dot{x}_j\right)I_{mi}+\left(\sum_{j=1}^{3}T_{1ij}\dot{x}_j\right)F_{vi}+\left(\sum_{j=1}^{3}T_{3ij}\right)F_{ci}+d_1,(i=1,2,3). \tag{6}$$

Based on the above method, the parameters $\boldsymbol{I}_m$, $\boldsymbol{F}_v$, and $\boldsymbol{F}_c$, which need to be identified, are successfully separated from the strongly coupled equations and constructed into three independent equations. It should be noted that the matrix $\boldsymbol{T}_i(i=1,2,3)$ and the vectors $\boldsymbol{U}^*$, $\ddot{\boldsymbol{x}}$, and $\dot{\boldsymbol{x}}$ are all known quantities (which can be obtained by measurement or calculation). Therefore, the parameter identification problem can be transformed into a typical multivariate linear regression problem.

3. Data-driven trajectory prediction methods

Identification algorithms often require a large amount of actual test data, and using AI algorithms to build digital twin systems has become a research hotspot. Recently, advanced intelligent algorithms such as long short-term memory networks (LSTM) [31] and Transformer [32] have been widely applied in the trajectory prediction of various systems, including unmanned aerial vehicles (UAVs) as well as self-driving cars. This chapter aims to introduce an efficient and accurate prediction network architecture and to propose corresponding evaluation indicators to measure prediction performance.

The Transformer, proposed by Vaswani *et al.* [33], is a powerful algorithm using the self-attention method. By substituting the recursive and local convolution operations of traditional architectures with a self-attention mechanism, this architecture achieves global dependency modeling and parallel capture of long-range dependencies across sequence elements. Given an input sequence X, Vaswani [33] introduces

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

three significant matrices Q, R, and V named the 'Query' matrix, 'Key' matrix, and 'Value' matrix, respectively. These matrices can be calculated by following equations:

$$Q = X \cdot W_Q, \quad K = X \cdot W_K, \quad V = X \cdot W_V,$$

in which, W_Q, W_K, and W_V are parameter matrices. Therefore, the mathematical expression of the self-attention value can be calculated as follows:

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d_k}}\right) V, \tag{7}$$

in (7), $A(Q, K, V)$ stands for the normalized attention weight matrix, while d_k represents the dimension of the 'Key' vectors (usually consistent with the dimension of the input vector X). In time series prediction tasks, the key vector can effectively represent the temporal characteristics of historical time points (such as periodicity, trend, etc.). Another core component of the Transformer architecture is the Feedforward Neural Network (FFN). This module consists of nonlinear activation functions and cascaded linear transformations. By increasing the network depth and parameter space dimensions, the model's ability to represent nonlinear features is significantly enhanced. It is worth noting that FFN has structural similarities with the Temporal Convolutional Network (TCN) in terms of feature mapping mechanisms. Leveraging this capability, we propose a hybrid TCN-Transformer architecture that integrates both models.

Bai, Kolter, and others [34] built a temporal convolutional network (TCN) based on convolution operations. The network architecture is mainly composed of three core components: causal convolution, dilated convolution, and residual connection. The form of the first part is as follows (8):

$$y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-i}, \tag{8}$$

where the convolution kernel W is defined as $W = [w_1, w_2, \ldots, w_k]$, while k stands for the length of the kernel. This formula serves a crucial function in preventing information leakage; specifically, it blocks the influence of future information on the current prediction.

Another key variant is dilated convolution, characterized by its exponentially expanded receptive field. This design thus enables the effective modeling of long-range dependencies and global context. The mathematical form is as follows:

$$y_t = \sum_{i=0}^{k-1} w_i \cdot x_{t-d \cdot i}, \tag{9}$$

in which an expansion factor d is introduced and increases with the depth of the network. The residual connection mechanism includes a nonlinear activation function and linear transformation, and its mathematical expression is:

$$\boldsymbol{O} = \sigma(\boldsymbol{F}(\boldsymbol{X})) + \boldsymbol{W}_d \boldsymbol{X}, \tag{10}$$

where σ represents a nonlinear activation function, often using the ReLU function, and $\boldsymbol{W}_d$ is a learnable parameter matrix. Based on this topological relationship, this study constructed a TCN network module based on dilated causal convolution stacking, and its mathematical representation can be formalized as:

$$\begin{aligned} \boldsymbol{H}_t^{(1)} &= \sigma\left(\sum_{i=0}^{k-1} \boldsymbol{W}_i^{(1)} \cdot \boldsymbol{H}_{t-d \cdot i}^{(0)} + \boldsymbol{b}^{(1)}\right) \\ \boldsymbol{H}_t^{(2)} &= \sigma\left(\sum_{i=0}^{k-1} \boldsymbol{W}_i^{(2)} \cdot \boldsymbol{H}_{t-d \cdot i}^{(1)} + \boldsymbol{b}^{(2)}\right) \\ \boldsymbol{O}_t &= \boldsymbol{H}_t^{(2)} + \boldsymbol{W}_d \cdot \boldsymbol{H}_t^{(0)} \end{aligned}, \tag{11}$$

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

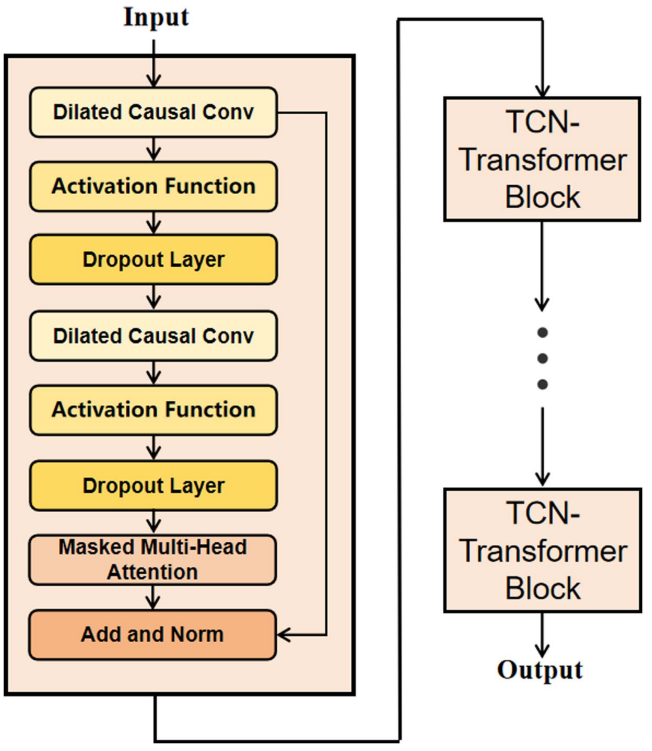


Figure 1. *TCN-Transformer network architecture schematic diagram.*

in the formula, $\boldsymbol{H}_t^{(i)}, i = 0, 1, 2$ represents the feature tensor output by the i-th layer dilated convolution operation. By deeply integrating the above TCN module with the Transformer architecture, the final TCN-Transformer architecture is revealed in Figure 1.

After the network is built, its performance needs to be evaluated. Traditional evaluation indicators such as mean square error (MSE) and mean absolute error (MAE) focus on measuring the deviation between the point prediction value and the true value, but their limitation is that they cannot quantify the model's ability to model the data distribution. To this end, this paper introduces perplexity as a supplementary evaluation indicator, aiming to evaluate the performance of the time series model from the perspective of probability generation. Assume that the output value of the data set is true and reliable, and the error of the predicted output follows a Gaussian distribution with a mean of μ as well as a variance of δ^2. The logarithmic probability density of the observation value of each prediction point $\log p(y_t|y_{1:t-1})$ can be obtained:

$$\ln p(y_t|y_{1:t-1}) = -\frac{1}{2}\ln\left(2\pi\delta_t^2\right) - \frac{(y_t - \mu_t)^2}{2\delta_t^2}. \tag{12}$$

The average negative log-likelihood value is obtained by accumulating the log probability density of each prediction point for the entire prediction output sequence and taking the average value:

$$V_{AN\,LL} = -\frac{1}{T}\sum_{t=1}^{T}\ln p(y_t|y_{1:t-1}). \tag{13}$$

Finally, we can obtain the calculation method of perplexity:

$$V_{PPL} = \exp V_{AN\,LL} = \exp\left(-\frac{1}{T}\sum_{t=1}^{T}\ln p(y_t|y_{1:t-1})\right). \tag{14}$$

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

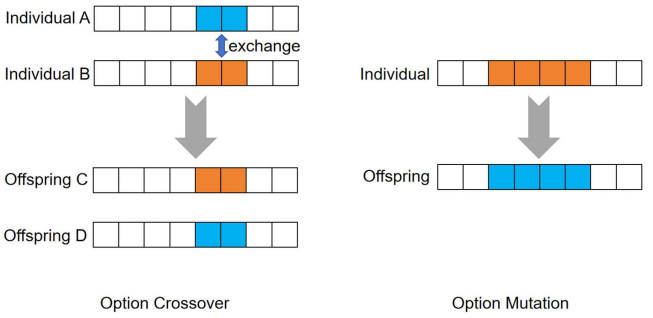


Figure 2. *Crossover and mutasion operations in GA.*

4. System identification method based on wavelet and TCN-Transformer

While the CDPR dynamic model is effective at characterizing low-frequency behavior, it does not account for high-frequency components, including high-frequency modal responses and noise. To address this limitation, this paper proposes an intelligent identification framework that integrates wavelet multiscale analysis with a deep learning network to estimate the system parameters I_m, F_v, and F_c defined in Section 2. The proposed method begins by decoupling the system output signal in the frequency domain via wavelet transform. A hybrid identification framework is then constructed, in which a GA and a TCN-Transformer network are employed to identify the parameters of the decoupled low-frequency components. Meanwhile, the TCN Transformer architecture is used for collaborative modeling and prediction of high-frequency components. The mathematical principles and implementation steps of the framework are systematically detailed in this section.

The Daubechies 4th-order wavelet (db4) was used to decompose the original data two times to construct the low-frequency approximate component and the high-frequency component. The mathematical expression is:

$$\begin{cases} S_{\text{low}} = W_{\phi}(s, 2) \\ s_{\text{high}} = W_{\psi}(s, 2) \end{cases}, \tag{15}$$

in which W is the wavelet decomposition operator, while ϕ and ψ are the scaling function and wavelet function, respectively. According to actual data testing, this algorithm can solve the problem of spectrum leakage associated with the FFT method. At this time, the low-frequency components we separated include $U^*_{i\text{low}}$, $\ddot{x}_{i\text{low}}$, $\dot{x}_{i\text{low}}$ and $\dot{l}_{j\text{low}}$ $(i = 1, 2, 3)$. Therefore, (6) can be rewritten as follows:

$$U^*_{i\text{low}} = a_{i\text{low}} I_{mi} + b_{i\text{low}} F_{vi} + c_{i\text{low}} F_{ci} + d_{i\text{low}}, (i = 1, 2, 3), \tag{16}$$

in which,

$$a_{i\text{low}} = \sum_{j=1}^{3} T_{1ij} \ddot{x}_{j\text{low}} + T_{2ij} \dot{x}_{j\text{low}}, b_{i\text{low}} = \sum_{j=1}^{3} T_{1ij} \dot{x}_{j\text{low}},$$

$$c_{i\text{low}} = \sum_{j=1}^{3} T_{3ij\text{low}} = \sum_{j=1}^{3} \text{si-tanh}, (\dot{l}_{i\text{low}}), (i = 1, 2, 3)$$

In Section 2, we transform the model identification problem into a multivariate linear regression problem. Genetic algorithms are introduced and combined with prediction networks to identify the required parameters. According to Griffiths [35], the core operating mechanism of GA mainly includes two key genetic operators: crossover recombination and gene mutation. The crossover operation refers to the exchange and recombination of chromosome segments between two parent individuals at specific gene loci to produce new individuals. The mutation operation randomly selects specific gene loci in the individual chromosome and resets their values to valid random values within the value range space. The schematic diagram of the two operations is shown in Figure 2.

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

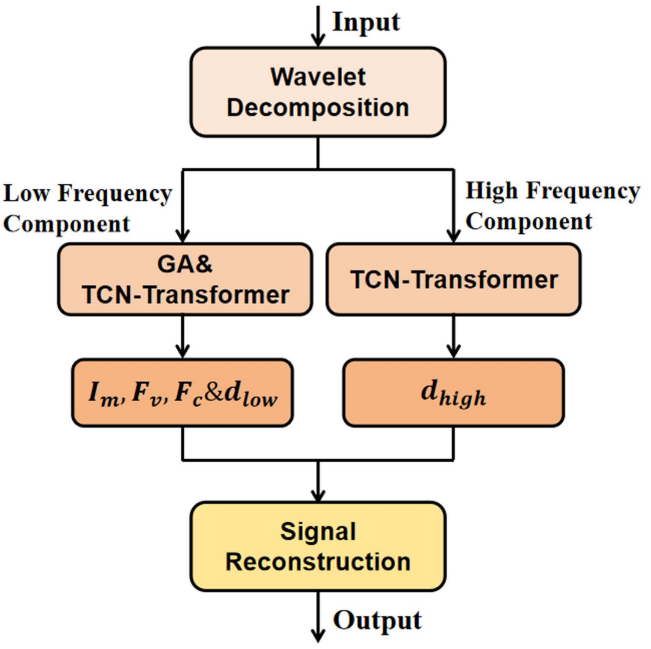


Figure 3. *Framework of system identification based on wavelet and TCN-transformer.*

The algorithmic process includes generating a population, performing crossover and genetic operations on individuals in the population to generate new individuals, calculating the fitness of each individual according to the fitness function, and selecting the best n individuals to retain. If the required target is not achieved, the above process is repeated until the required target is met. In this paper, each individual contains four genes: I'_m, F'_v, F'_c as well as d_{bios}. According to (16), the fitness function is constructed:

$$f_{\text{fitness}} = \sum_{i=1}^{n} \left(a_{\text{low}} I'_m + b_{\text{low}} F'_v + c_{\text{low}} F'_c + d_{\text{bias}} - U^*_{\text{low}} \right)^2, \tag{17}$$

where d_{bias} stands for the fixed bias, which can be fitted by the TCN-Transformer network. At this point, we can present the framework of the entire identification algorithm, as shown in Figure 3.

In most cases, obtaining the optimal system parameters requires a large amount of data. According to the trajectory prediction method proposed in Section 2, we can train a trajectory prediction system with a small amount of data and obtain a large amount of data through simulation to identify the optimal system parameters. Based on this, the overall framework of the method is constructed as follows: firstly, the TCN-Transformer model is trained using the collected small sample dataset to establish a virtual trajectory prediction system; secondly, the prediction system is used to generate high-quality virtual trajectories in batches; then, based on the generated virtual trajectory data, the identification method proposed in this study is used to construct a gray-box model of CDPRs; finally, the method proposed in this paper is verified and optimized using physical experimental data. Figure 4 is a schematic diagram of the overall framework.

In the next section, the feasibility and accuracy of this method will be demonstrated through experiments.

5. Simulation and experiments

5.1. Experimental equipment

To verify the performance advantages of the proposed method, experiments are conducted on a 3-DOF CDRS experimental platform, as shown in Figure 5. The CDPR system operates in a 3 m × 3 m × 2 m

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

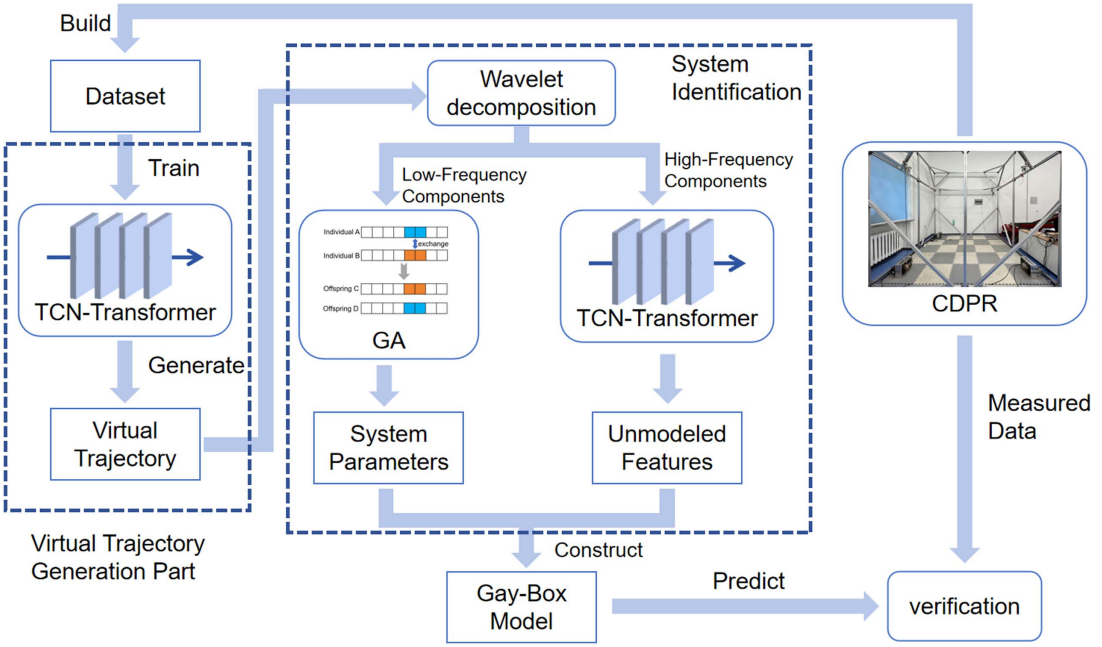


Figure 4. *Flowchart of overall framework.*

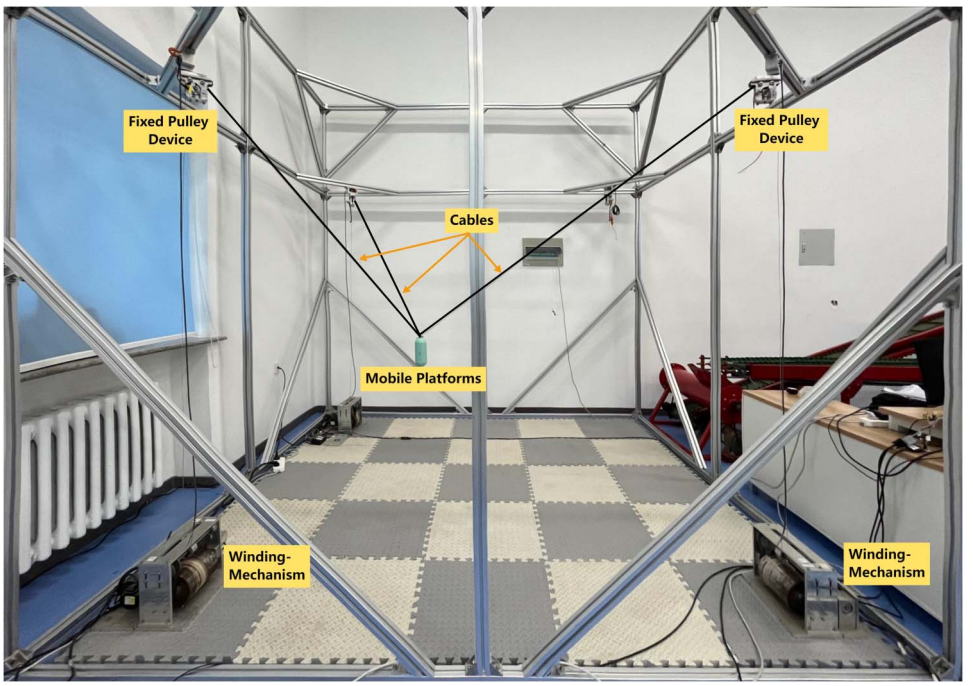


Figure 5. *Three-degree-of-freedom CDPR.*

rectangular space and consists of a mobile robot platform and three sets of cable-driven components (including winches and guiding devices).

This experiment also utilizes the Prime series high-precision motion capture system. Within the experimental space shown in Figure 6, 24 infrared optical capture cameras are evenly distributed according to the spatial geometry, forming a comprehensive three-dimensional tracking network. This enables

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

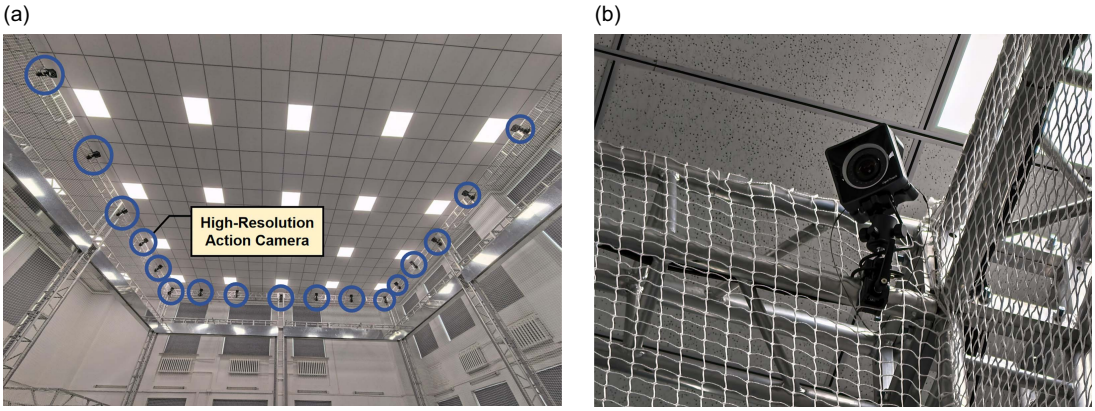


Figure 6. *Prime series high-precision motion capture system.*

comprehensive, all-around tracking and pose reconstruction of indoor moving targets. Each camera is equipped with an advanced CMOS sensor that has a resolution of 12 megapixels and a base frame rate of 300 Hz, which can be increased to 1000 Hz in high-speed mode, ensuring accurate capture of high-speed motion details. Through sub-pixel marker recognition and multi-camera joint solution technology, the system achieves a spatial positioning accuracy of 0.1 mm, an accuracy indicator verified by international standard metrology instruments.

5.2. Data-driven trajectory generation experiment

To validate the effectiveness and accuracy of the proposed algorithm, we have collected a total of 200 experimental data sets during the normal operations of the CDPR. The trajectories contained in this data set are mainly divided into four categories: flat circle trajectories, oblique circle trajectories, flat Figure 7 trajectories, and oblique Figure 7 trajectories. Each category of trajectory contains 50 complete datasets. In this part of the experiment, we will use this dataset to train the TCN-Transformer network proposed in this paper, TCN, Transformer network, and the BiLSTM, which is widely used. Meanwhile, the mean squared error (MSE), the root mean squared error (RMSE), the MAE, the mean absolute percentage error (MAPE), R-Square (R^2), and the proposed perplexity (PPL) will be calculated to measure the performance of these networks. After training, these networks will be tested on unknown trajectory generation to test their generalization performance.

The model input vector includes the time parameter t, the desired position of the mobile platform at the current moment $x_d(t)$, the measured position $x(t)$, the real-time velocity $v(t)$, the acceleration $a(t)$, the rope length $l(t)$, and the control input $u(t)$. The output is the predicted value $x(t+1)$ of the platform's position at the next moment. The experimental parameters are set as follows: TCN-Transformer, TCN, and Transformer all use a four-layer network architecture, trained for 300 cycles, with an initial learning rate set to 0.05 and an Adam optimizer; BiLSTM is configured with 256 hidden units, and the other hyperparameters remain unchanged. The number of attention heads in the Transformer framework is set to 16, the number of filters in the TCN is 5, whose width is 16, and the dropout factor is 10%. Figure 8 shows the training errors of these networks, and Table I shows the values of related indicators.

Based on the training set performance data in Table I, the TCN-Transformer network demonstrates comprehensive superiority in prediction accuracy, efficiency, and stability. Its core error metrics are significantly superior, with the x-axis MSE as low as 5.5637×10^{-6} (only 18.9% of TCN's) and the z-axis MAE as low as 1.5706×10^{-3} (30.9% lower than the Transformer). Seven of the nine metrics used for evaluation were optimal. In particular, both the MSE and MAE were the lowest, 2.1% and 28.2%

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

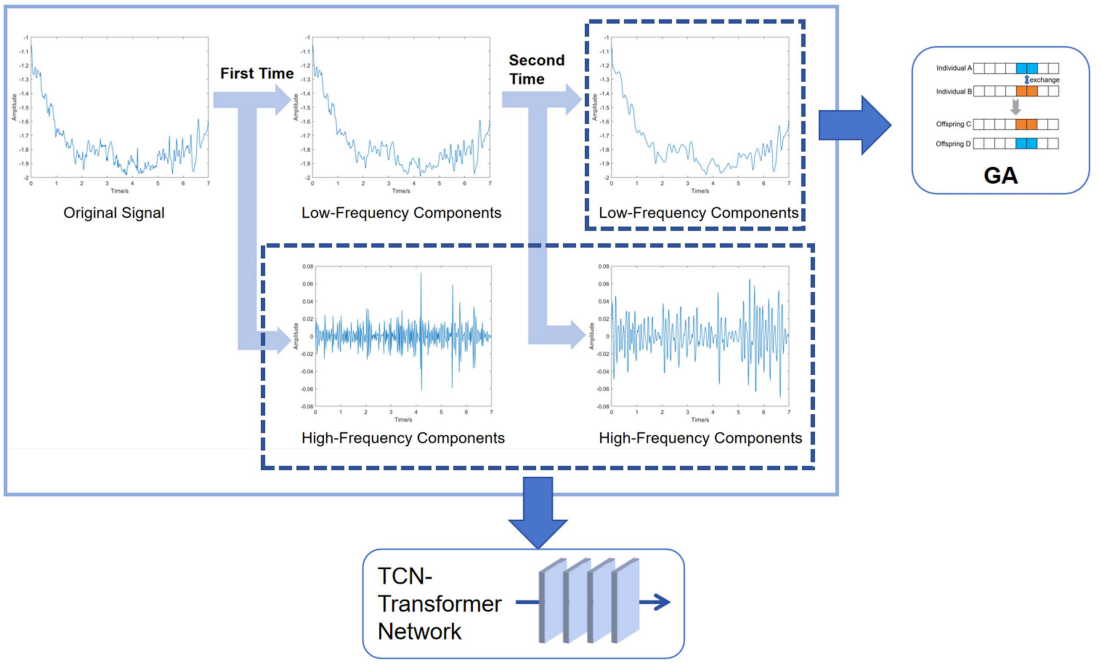


Figure 7. *The result of wavelet decomposition.*

lower than the second-place Transformer, respectively. Training efficiency also achieved a significant improvement, completing training in just 88 s, a 43.2% speedup compared to the Transformer (155 *s*) and a 32.8% reduction compared to the BiLSTM (131 *s*), breaking the barrier that high-precision models often sacrifice speed. The model's reliability was also double-checked: R^2 reaches 99.9914% (surpassing the Transformer's 99.9759%), demonstrating its strongest ability to model system dynamics. Furthermore, the perplexity PPL was lower than that of the comparison network on both the *x*/*z* axis and the average value, demonstrating the most concentrated distribution of prediction errors. In particular, compared to TCN and BiLSTM, this network, leveraging the local feature extraction capabilities of TCN temporal convolution and the global dependency modeling advantages of the Transformer, exhibits high accuracy, high efficiency, and strong stability.

Moreover, the values of MSE and RMSE can indeed reflect the overall degree of deviation between the predicted value and the actual data; however, the averaging algorithm may cause these indicators to ignore some details, such as the maximum and minimum values of the error. The data from the TCN-Transformer and the Transformer illustrate this conclusion. MAE and MAPE intuitively represent the gap between the predicted and the true value, but they do not take into account the variability of the target variable and may not accurately reflect the performance of the model. The PPL proposed in this paper can address the shortcomings of the aforementioned indicators. PPL not only reflects the overall degree of deviation but also indicates the magnitude of data fluctuation. The greater the fluctuation, the larger the PPL value.

Furthermore, these results demonstrate the high practicality of the proposed method. Firstly, to address the issues of high experimental costs, long experimental cycles, and aging losses, the method can utilize limited real data to generate large volumes of high-fidelity virtual data for parameter identification. This approach significantly shortens development cycles, reduces energy consumption, and minimizes equipment wear during testing, enabling rapid iteration and optimization. Secondly, high-quality trajectory prediction not only improves the accuracy of parameter identification but also provides a reliable model foundation for advanced control methods, enhancing both trajectory tracking precision and system stability.

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

Table I. *Comparison of prediction results of training set.*

	TCN-Transformer			TCN			Transformer			BiLSTM		
Index	***x***	***y***	***z***	***x***	***y***	***z***	***x***	***y***	***z***	***x***	***y***	***z***
MSE (m^2)	5.5637 $\times 10^{-6}$	6.7672 $\times 10^{-6}$	4.2720 $\times 10^{-6}$	2.9352 $\times 10^{-5}$	2.2196 $\times 10^{-5}$	5.8709 $\times 10^{-5}$	6.7707 $\times 10^{-6}$	1.8836 $\times 10^{-5}$	8.3078 $\times 10^{-6}$	7.3314 $\times 10^{-6}$	9.6222 $\times 10^{-6}$	9.1844 $\times 10^{-6}$
RMSE (m)	2.3588 $\times 10^{-3}$	2.6014 $\times 10^{-3}$	2.0669 $\times 10^{-3}$	5.4177 $\times 10^{-3}$	4.7113 $\times 10^{-3}$	7.6622 $\times 10^{-3}$	2.6021 $\times 10^{-3}$	4.3400 $\times 10^{-3}$	2.8823 $\times 10^{-3}$	2.7077 $\times 10^{-3}$	3.1020 $\times 10^{-3}$	3.0306 $\times 10^{-3}$
MAE (m)	1.8882 $\times 10^{-3}$	2.0677 $\times 10^{-3}$	1.5706 $\times 10^{-3}$	4.3740 $\times 10^{-3}$	3.8227 $\times 10^{-3}$	6.0105 $\times 10^{-3}$	2.0260 $\times 10^{-3}$	3.3984 $\times 10^{-3}$	2.2716 $\times 10^{-3}$	2.1217 $\times 10^{-3}$	2.5601 $\times 10^{-3}$	2.2292 $\times 10^{-3}$
MAPE (%)	0.1948	0.2235	0.1673	0.4735	0.4142	0.6476	0.2077	0.3733	0.2501	0.2264	0.2625	0.2354
PPL (%)	0.66	1.14	0.79	1.90	3.07	2.45	0.79	0.99	1.22	0.82	1.19	0.95
R^2(%)		99.9914			99.9248			99.9759			99.9737	
$T(s)$		88			71			155			131	

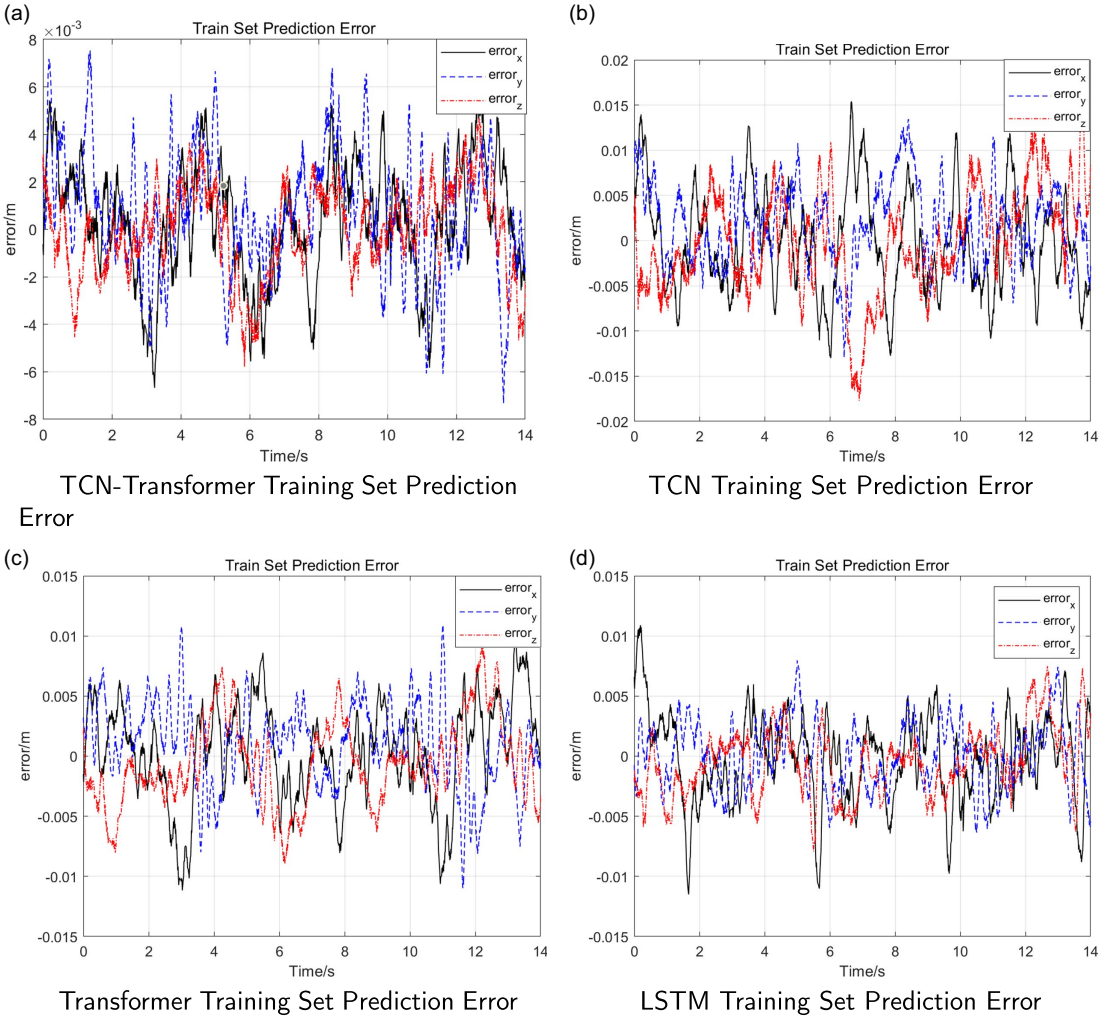


Figure 8. *The error of training set of four network structures. (a) TCN-transformer; (b) TCN; (c) Transformer; (d) LSTM.*

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

Table II. *Comparison of prediction results of test set.*

		TCN-Transformer			TCN			Transformer			BiLSTM		
Type	**Index**	***x***	***y***	***z***	***x***	***y***	***z***	***x***	***y***	***z***	***x***	***y***	***z***
Flat Circle	MSE (m^2)	6.4813 $\times10^{-5}$	9.3215 $\times10^{-5}$	1.2380 $\times10^{-5}$	9.1908 $\times10^{-5}$	4.7113 $\times10^{-3}$	6.5908 $\times10^{-5}$	1.2955 $\times10^{-4}$	4.3527 $\times10^{-4}$	9.0922 $\times10^{-5}$	5.5727 $\times10^{-5}$	1.0002 $\times10^{-5}$	5.8534 $\times10^{-5}$
	RMSE (m)	8.0506 $\times10^{-3}$	9.3215 $\times10^{-3}$	3.5185 $\times10^{-3}$	9.5869 $\times10^{-3}$	5.6617 $\times10^{-2}$	8.1184 $\times10^{-3}$	1.1382 $\times10^{-2}$	2.0863 $\times10^{-2}$	9.5353 $\times10^{-3}$	2.3607 $\times10^{-2}$	3.1627 $\times10^{-3}$	2.4194 $\times10^{-2}$
	MAE (m)	6.4268 $\times10^{-3}$	7.7022 $\times10^{-3}$	3.0207 $\times10^{-3}$	7.7520 $\times10^{-3}$	5.1232 $\times10^{-2}$	6.4427 $\times10^{-3}$	1.0111 $\times10^{-2}$	1.8894 $\times10^{-2}$	7.9806 $\times10^{-3}$	2.1335 $\times10^{-2}$	2.7300 $\times10^{-2}$	2.3329 $\times10^{-2}$
	MAPE (%)	0.7314	0.9436	0.3229	0.9358	6.9036	0.6850	1.156	2.2587	0.8463	2.3862	3.2388	2.4250
	PPL (%)	2.02	2.42	0.88	2.40	14.19	2.03	2.85	5.23	2.39	5.92	7.93	2.425
	R^2(%)	99.7911			96.0718			99.3938			98.4616		
Oblique Circle	MSE (m^2)	6.5796 $\times10^{-5}$	3.9664 $\times10^{-5}$	2.8029 $\times10^{-5}$	2.9352 $\times10^{-5}$	2.2196 $\times10^{-5}$	5.8709 $\times10^{-5}$	4.3667 $\times10^{-5}$	6.9535 $\times10^{-5}$	1.4649 $\times10^{-4}$	7.0073 $\times10^{-5}$	7.2623 $\times10^{-5}$	4.5770 $\times10^{-5}$
	RMSE (m)	8.1114 $\times10^{-3}$	6.2980 $\times10^{-3}$	5.2942 $\times10^{-3}$	5.4177 $\times10^{-3}$	4.7113 $\times10^{-3}$	7.6622 $\times10^{-3}$	6.6087 $\times10^{-3}$	8.3388 $\times10^{-3}$	1.2103 $\times10^{-2}$	8.3709 $\times10^{-3}$	8.5219 $\times10^{-3}$	6.7653 $\times10^{-3}$
	MAE (m)	7.6480 $\times10^{-3}$	5.3485 $\times10^{-3}$	4.5271 $\times10^{-3}$	4.3740 $\times10^{-3}$	3.8227 $\times10^{-3}$	6.0105 $\times10^{-3}$	5.5738 $\times10^{-3}$	7.5454 $\times10^{-3}$	1.0579 $\times10^{-2}$	6.4863 $\times10^{-3}$	6.9086 $\times10^{-3}$	5.3090 $\times10^{-3}$
	MAPE (%)	0.8834	0.5187	0.4659	0.4735	0.4142	0.6476	0.6214	0.7298	1.0517	0.6803	0.6891	0.5391
	PPL (%)	2.03	1.58	1.33	1.90	3.07	2.45	1.66	2.09	3.03	2.10	2.14	1.70
	R^2(%)	99.9231			99.9248			99.8419			99.9332		
Oblique Eight	MSE (m^2)	8.088 $\times10^{-5}$	1.0171 $\times10^{-4}$	3.4892 $\times10^{-5}$	5.3792 $\times10^{-4}$	5.9618 $\times10^{-3}$	7.0291 $\times10^{-5}$	3.4716 $\times10^{-4}$	8.4416 $\times10^{-4}$	9.5268 $\times10^{-5}$	4.6961 $\times10^{-4}$	1.4218 $\times10^{-3}$	6.6410 $\times10^{-4}$
	RMSE (m)	8.9933 $\times10^{-3}$	10.085 $\times10^{-3}$	5.9069 $\times10^{-3}$	2.3193 $\times10^{-2}$	7.7213 $\times10^{-2}$	8.3840 $\times10^{-3}$	1.8632 $\times10^{-2}$	2.9054 $\times10^{-2}$	9.7606 $\times10^{-3}$	2.1670 $\times10^{-2}$	3.7706 $\times10^{-2}$	2.5770 $\times10^{-2}$
	MAE (m)	7.3915 $\times10^{-3}$	8.4321 $\times10^{-3}$	5.6305 $\times10^{-3}$	1.9521 $\times10^{-2}$	6.7245 $\times10^{-2}$	7.1470 $\times10^{-3}$	1.5733 $\times10^{-2}$	2.6192 $\times10^{-2}$	7.9516 $\times10^{-3}$	1.9110 $\times10^{-2}$	3.2352 $\times10^{-2}$	2.4876 $\times10^{-2}$
	MAPE (%)	0.7785	0.9977	0.6012	2.4164	9.3486	0.7583	1.8910	3.2506	0.8420	1.9865	3.9657	2.5744
	PPL(%)	2.25	2.53	1.48	5.81	19.35	2.10	4.67	7.28	2.45	5.43	9.45	6.46
	R^2(%)	99.8200			93.0208			98.9562			97.8548		
Spiral	MSE (m^2)	1.5345 $\times10^{-4}$	1.1307 $\times10^{-4}$	1.3041 $\times10^{-5}$	3.5002 $\times10^{-4}$	3.5199 $\times10^{-3}$	8.0475 $\times10^{-5}$	3.4716 $\times10^{-4}$	8.4416 $\times10^{-4}$	9.5268 $\times10^{-5}$	1.4112 $\times10^{-4}$	2.9435 $\times10^{-4}$	3.2333 $\times10^{-5}$
	RMSE (m)	1.2387 $\times10^{-2}$	1.1307 $\times10^{-2}$	3.6112 $\times10^{-3}$	1.8709 $\times10^{-2}$	5.9329 $\times10^{-2}$	8.9708 $\times10^{-3}$	1.8632 $\times10^{-2}$	2.9054 $\times10^{-2}$	9.7606 $\times10^{-3}$	1.1880 $\times10^{-2}$	1.7157 $\times10^{-2}$	5.6862 $\times10^{-3}$
	MAE (m)	1.0624 $\times10^{-2}$	9.6920 $\times10^{-3}$	3.5284 $\times10^{-3}$	1.6340 $\times10^{-2}$	4.8543 $\times10^{-2}$	7.4654 $\times10^{-3}$	1.5733 $\times10^{-2}$	2.6192 $\times10^{-2}$	7.9516 $\times10^{-3}$	1.0063 $\times10^{-2}$	1.5646 $\times10^{-2}$	5.6549 $\times10^{-3}$
	MAPE (%)	1.0724	0.9630	0.3765	1.5182	6.8900	0.7914	1.8910	0.8420	0.2501	0.9490	1.5646	0.6044
	PPL (%)	3.11	2.67	0.91	4.69	14.87	2.25	4.67	7.28	2.45	2.98	4.30	1.43
	R^2(%)	99.7396			96.1333			98.9562			99.5521		

After training, we will use four trajectories to test the generalization ability of these networks: a flat circle, an oblique circle, an oblique 8 trajectory included in the training set, and a spiral trajectory that does not appear in the training set. The test results are shown in Table II.

According to the data in Table II, the TCN-Transformer network demonstrates significant advantages in performance and generalization. In known trajectory testing, its prediction accuracy surpasses that of all competing networks. For planar circular trajectories, the z-axis MSE is over 86% lower than that of the Transformer and BiLSTM, and the z-axis MAE is 87% higher than that of the BiLSTM. For oblique circular trajectories, the x-axis MSE is significantly better than that of the BiLSTM, while all three axes of the PPL are lower than those of other networks, demonstrating a more concentrated error distribution. For highly nonlinear oblique Figure 7 trajectories, the z-axis MAE is 21% to 77% lower than that of the TCN and BiLSTM, and the z-axis PPL is significantly lower than that of the BiLSTM, highlighting its

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

Table III. *Comparison of prediction results of spiral trajectory on test sets.*

Params	GA&TCN-Transformer	LSM
I_m	diag(0.189, 0.103, 0.134)	diag(0.182, 0.110, 0.146)
F_v	diag(0.285, 1.216, 0.156)	diag(0.289, 1.132, 0.020)
F_c	diag(0.622, 0.413, 0.592)	diag(0.635, 0.588, 0.859)
d_{low}	—	[−0.961, −1.388, −0.946]

robustness in modeling complex motion. Across 16 coordinate-trajectory combinations, this network achieves the lowest PPL in 75% cases and maintains a leading position in 12 of these results.

In the unknown trajectory generalization test (spiral trajectories), the TCN-Transformer's advantage further expanded: its x-axis MSE is 77% lower than the best comparison network, the Transformer, while its y-axis MAE is 73% higher than the BiLSTM. It maintained the lowest PPL on the z-axis and achieved an R^2 of 99.7396%, significantly outperforming TCN. This demonstrates the network's strong adaptability to untrained motion patterns. Its core advantages can be summarized in three key points: firstly, the TCN's local feature extraction and the Transformer's global dependency modeling capabilities achieved the lowest MSE/MAE in the test scenario; secondly, the innovative PPL metric verified its most stable prediction error distribution; and thirdly, key metrics outperformed the runner-up network by an average of over 35% when faced with unknown spiral trajectories, demonstrating the architecture's ability to capture high-frequency dynamic characteristics. These characteristics make it an ideal tool for parameter identification of complex systems, particularly in engineering scenarios requiring high-precision virtual trajectory generation.

The following will show the comparative analysis of experimental data and performance indicators between the traditional method and the intelligent identification method based on wavelet theory. Firstly, we show the results of wavelet decomposition in Figure 7. As mentioned in Section 4, the wavelet decomposition has two layers using the db4 wavelet as the base. Therefore, the original signal has been divided into three components: low-frequency components and high-frequency components from the first and second decompositions. As we can see from Figure 7, the frequency of the high-frequency components obtained from the first decomposition is significantly higher than that from the second. The obtained low-frequency components were used to identify system parameters, while the decomposed high-frequency components were used for fitting by the TCN-Transformer.

The identification values of relevant system parameters are provided by two methods based on the measured data:

$$\boldsymbol{M} = \text{diag}(2, 2, 2)(kg),$$
$$\boldsymbol{R} = \text{diag}(0.06, 0.06, 0.06).$$

Based on the constructed dataset and the virtual dataset generated by the trajectory prediction method, the system parameter comparison results obtained by this research method and the traditional method (traditional least squares method, LSM) are shown in Table III.

According to the data in Table III, it is found that the system parameters of the x-axis identified by the two methods are almost the same, F_c of the y-axis is significantly different, and F_v and F_c of the z-axis are significantly different. In order to further evaluate the advantages and disadvantages of the two methods, data reconstruction and visualization analysis are carried out based on the helical trajectory results from the aforementioned trajectory prediction and the parameter identification results. Due to space limitations, this paper only shows the reconstruction curve under the condition of $i = 1$, as shown in Figure 9.

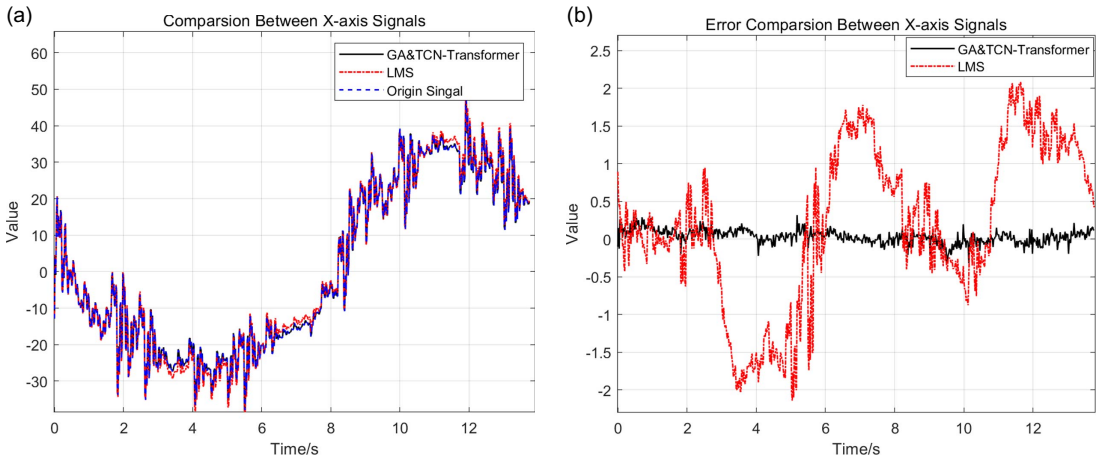
The error metrics, including MSE, MAE, and PPL, associated with the two methods are shown in Table IV.

The comparative analysis in Figure 9 shows that the traditional parameter identification method has significant deviations, while the proposed improved algorithm has significant advantages in parameter

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

Table IV. *Comparison of signal reconstruction results.*

	GA&TCN-Transformer			LSM		
Index	***x***	***y***	***z***	***x***	***y***	***z***
MSE	0.0094	0.0284	0.0294	0.7388	1.2503	1.3741
MAE	0.0762	0.1310	0.1327	0.8122	4.0231	5.1559
PPL	0.2051	0.4870	0.4877	2.2591	5.8579	8.1026



Comparison of Reconstructed Signal and Original Signal

Comparison of Two Methods' Error

Figure 9. *Comparison between TCN-transformer and LSM. (a) Signal. (b) Error.*

identification accuracy, and its residual range converges to the interval $[-0.207, 0.470]$. Comparing the data in Table IV, it can be found that the three indicators of the proposed method are superior to those of LSM. Especially in the PPL index, the value of the new method is much lower than that of LSM. This shows that the traditional methods are not able to deal with high-frequency components and noise, leading to an underestimation of the friction effect. Through wavelet enhancement and genetic algorithm optimization, the new method can more comprehensively characterize the high-frequency dynamics of the system, especially the friction nonlinearity.

It should be pointed out that although the parameter identification accuracy is improved, the signal reconstruction error is still maintained at a high level due to the inherent structure of the model, which suggests that subsequent research needs to optimize the structure of the model described in Chapter 2. Substitute the optimized parameters into the identification framework proposed in this paper, dynamically identify the system disturbance d through TCN-Transformer, and implement signal reconstruction based on Eq. (1). The reconstruction effect verification is shown in Figure 7.

The error ranges between the three-axis reconstructed signal and the original signal are $[-0.3294, 0.3206]$, $[-0.7967, 0.6641]$ and $[-0.5372, 0.9366]$ respectively. The analysis results show that the parameter identification method proposed in this study can accurately obtain the system dynamic parameters $\boldsymbol{I}_m$, $\boldsymbol{F}_v$, and $\boldsymbol{F}_c$, and realize the dynamic prediction of the system disturbance $\boldsymbol{d}$.

Applying the above results to the actual dataset, the result of one oblique circle trajectory is shown in Table V.

Figure 10 illustrates the comparison between the calculated results and the original signal, and Table V gives the index values. Thus, t we can draw a conclusion that the identification results calculated by the previously constructed virtual trajectory can still achieve a good fitting effect when applied to the actual trajectory.

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

Table V. *The results of applying simulation results to actual trajectories.*

Indexes	***x***	***y***	***z***
MSE	0.0122	0.0426	0.0505
MAE	0.1178	0.1706	0.1659
PPL	0.2428	0.5174	0.5635

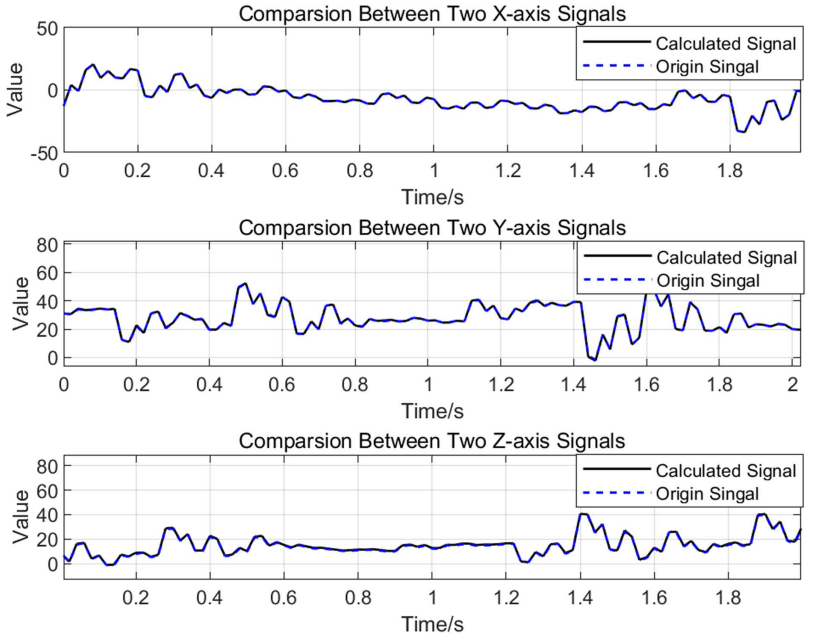


Figure 10. *Comparison of the signal reconstructed by the new method with the original signal.*

6. Conclusion

This paper addresses the challenge of identifying strongly coupled and nonlinear systems for cable-driven parallel robots by proposing an intelligent identification framework that integrates wavelet transforms and deep learning. By constructing a TCN-Transformer hybrid network architecture, high-precision trajectory prediction and virtual data generation are achieved, effectively reducing the cost of experimental data acquisition. Wavelet multiscale decomposition technology is combined to separate the high- and low-frequency components of the system's dynamic characteristics, and a gray-box identification strategy that collaborates with genetic algorithms and TCN-Transformer is designed. This enhances the ability to characterize high-frequency characteristics while preserving the physical model mechanism. Experiments show that this approach significantly improves the accuracy and efficiency of parameter identification, addressing the limitations of traditional methods in modeling nonlinearities and the lack of physical interpretability in data-driven methods. Subsequent research will focus on optimizing the model structure to further enhance system robustness and engineering applicability.

Author contributions. Conceptualization, data curation, software, formal analysis, methodology, and writing – original draft were performed by R. Wang; supervision, funding acquisition, project administration, and writing – review & editing were performed by Y. Li; supervision, funding acquisition, and writing – review & editing were performed by Y. Zeng.

Financial support. This research was funded in part by the General Research Fund of Hong Kong under Grant 15206223. And it was also supported in part by the Harbin Institute of Technology; in part by the National Natural Science Foundation of China under grants 62188101, 62303135, and 62320106001; and in part by the Heilongjiang Touyan Team Program.

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] M. Zarebidoki, J. S. Dhupia and W. Xu, "A review of cable-driven parallel robots: Typical configurations, analysis techniques, and control methods," *IEEE Robot. Autom. Mag.* **29**(3), 89–106 (2022).

[2] R. Wang and Y. Li, "Jerk-limited time-optimal model predictive path following control of cable-driven parallel robots," *IEEE Robot. Autom. Lett.* **8**(10), 6731–6738 (2023).

[3] V. Le Nguyen and R. J. Caverly, "Cable-driven parallel robot pose estimation using extended Kalman filtering with inertial payload measurements," *IEEE Robot. Autom. Lett.* **6**(2), 3615–3622 (2021).

[4] H. Wang, J. Kinugawa and K. Kosuge, "Exact kinematic modeling and identification of reconfigurable cable-driven robots with dual-pulley cable guiding mechanisms," *IEEE/ASME Trans. Mechatron.* **24**(2), 774–784 (2019).

[5] J. Gao, B. Zhou, B. Zi, S. Qian and P. Zhao, "Kinematic uncertainty analysis of a cable-driven parallel robot based on an error transfer model," *J. Mech. Robot.* **14**(5), 051008 (2022).

[6] H. Ji, W. Shang and S. Cong, "Adaptive synchronization control of cable-driven parallel robots with uncertain kinematics and dynamics," *IEEE Trans. Ind. Electron.* **68**(9), 8444–8454 (2021).

[7] X. Li, Q. Kui, W. Yao and L. Wu, "Robust parallel cooperative control of cable-driven robot system via adaptive integral sliding mode," *IEEE Robot. Autom. Lett.* **10**(6), 5433–5440 (2025).

[8] Y. Liu, Z. Cao, H. Xiong, J. Du, H. Cao and L. Zhang, "Dynamic obstacle avoidance for cable-driven parallel robots with mobile bases via sim-to-real reinforcement learning," *IEEE Robot. Autom. Lett.* **8**(3), 1683–1690 (2023).

[9] Q. Jiang and V. Kumar, "The inverse kinematics of cooperative transport with multiple aerial robots," *IEEE Trans. Robot.* **29**(1), 136–145 (2013).

[10] M. Krizmancic, B. Arbanas, T. Petrovic, F. Petric and S. Bogdan, "Cooperative aerial-ground multi-robot system for automated construction tasks," *IEEE Robot. Autom. Lett.* **5**(2), 798–805 (2020).

[11] B. Zhang, Z. Li, X. Wang, W. Shang, X. Gao, Y. Ma, R. Yao, H. Li, J. Yin, Q. Yang and Q. Li, "Trajectory planning of radio source tracking for the new feed cabin mechanism in FAST," *IEEE Trans. Control Syst. Technol.* **33**(3), 1051–1068 (2025).

[12] H. Hussein, J. C. Santos, J.-B. Izard and M. Gouttefarde, "Smallest maximum cable tension determination for cable-driven parallel robots," *IEEE Trans. Robot.* **37**(4), 1186–1205 (2021).

[13] J. Xu, B. Kim and K. Park, "A collaborative path planning method for mobile cable-driven parallel robots in a constrained environment with considering kinematic stability," *Complex Intell. Syst.* **9**(5), 4857–4868 (2023).

[14] H. Sun, X. Tang, Z. Cui and S. Hou, "Dynamic response of spatial flexible structures subjected to controllable force based on cable-driven parallel robots," *IEEE/ASME Trans. Mechatron.* **25**(6), 2801–2811 (2020).

[15] J. S. Khan, M. Mohammadi, J. Rasmussen, S. Bai and N. S. L. A. Struijk, "A Review on the Design of Assistive Cable-Driven Upper-Limb Exoskeletons and Their Experimental Evaluation," **In:** *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Prague, Czech Republic (2022) pp. 59–64.

[16] F. Zhang, W. Shang, G. Li and S. Cong, "Calibration of geometric parameters and error compensation of non-geometric parameters for cable-driven parallel robots," *Mechatronics* **77**(102595), 0957–4158 (2021).

[17] H. Wang, T. Gao, J. Kinugawa and K. Kosuge, "Finding measurement configurations for accurate robot calibration: Validation with a cable-driven robot," *IEEE Trans. Robot.* **33**(5), 1156–1169 (2017).

[18] Y. Lu, C. Wu, W. Yao, G. Sun, J. Liu and L. Wu, "Deep reinforcement learning control of fully-constrained cable-driven parallel robots," *IEEE Trans. Ind. Electron.* **70**(7), 7194–7204 (2023).

[19] P. H. Borgstrom, B. L. Jordan, B. J. Borgstrom, M. J. Stealey, G. S. Sukhatme, M. A. Batalin and W. J. Kaiser, "NIMS-PL: A cable-driven robot with self-calibration capabilities," *IEEE Trans. Robot.* **25**(5), 1005–1015 (2009).

[20] Z. Liu, Z. Qin, H. Gao, G. Sun, Z. Huang and Z. Deng, "Initial-pose self-calibration for redundant cable-driven parallel robot using force sensors under hybrid joint-space control," *IEEE Robot. Autom. Lett.* **8**(3), 1367–1374 (2023).

[21] R. Wang and Y. Li, "Analysis and multi-objective optimal design of a planar differentially driven cable parallel robot," *Robotica* **39**(12), 2193–2209 (2021).

[22] C. Sancak, F. Yamac and M. Itik, "Position control of a planar cable-driven parallel robot using reinforcement learning," *Robotica* **40**(10), 3378–3395 (2022).

[23] Z. Li, W. Shang and B. Zhang, "Hybrid impedance control of cable-driven parallel robots using adaptive friction compensation," *IEEE Trans. Ind. Electron.* **72**(1), 703–713 (2025).

[24] J. Piao, M.-C. Kim, E.-S. Kim and C.-S. Kim, "A self-adaptive inertia hybrid control of a fully constrained cable-driven parallel robot," *IEEE Trans. Autom. Sci. Eng.* **21**(2), 2063–2073 (2024).

[25] S. Mohapatra, J. Nayak, M. Mishra, G. K. Pati, B. Naik and T. Swarnkar, "Wavelet transform and deep convolutional neural network-based smart healthcare system for gastrointestinal disease detection," *Interdiscip. Sci.: Comput. Life Sci.* **13**(2), 212–228 (2021). https://doi.org/10.1007/s12539-021-00417-8

[26] Y. S. U. Vishwanath, S. Esakkirajan, B. Keerthiveena and R. B. Pachori, "A generalized classification framework for power quality disturbances based on synchrosqueezed wavelet transform and convolutional neural networks," *IEEE Trans. Instrum. Meas.* **72**, 1–13 (2023). https://doi.org/10.1109/TIM.2023.3308235

[27] R. Li, J. Xuan and T. Shi, "A frequency estimation method for two adjacent frequencies," *Measurement* **157**, 107614 (2020). https://doi.org/10.1016/j.measurement.2020.107614

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press

[28] R. Hasanzadeh Fereydooni, H. Siahkali, H. A. Shayanfar and A. H. Mazinan, "SEMG-based variable impedance control of lower-limb rehabilitation robot using wavelet neural network and model reference adaptive control," *Ind. Robot.* **47**(3), 349–358 (2020). https://doi.org/10.1108/IR-10-2019-0210
[29] V. Panwar, "Wavelet neural network-based H*infty* trajectory tracking for robot manipulators using fast terminal sliding mode control," *Robotica* **35**(7), 1488–1503 (2017). https://doi.org/10.1017/S0263574716000278
[30] M. H. Hamedani, M. Zekri and F. Sheikholeslam, "Adaptive impedance control of uncertain robot manipulators with saturation effect based on dynamic surface technique and self-recurrent wavelet neural networks," *Robotica* **37**(1), 161–188 (2019). https://doi.org/10.1017/S0263574718000930
[31] S. Alsanwy, H. Asadi, M. R. C. Qazani, S. Mohamed and S. Nahavandi, "A CNN-LSTM Based Model to Predict Trajectory of Human-Driven Vehicle," **In:** *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Honolulu, Oahu, HI, USA (2023) pp. 3097–3103.
[32] Z. Niu, T. Hassan, M. N. Boushaki, N. Werghi and I. Hussain, "Continuous wavelet network for efficient and transferable collision detection in collaborative robots," *IEEE Trans. Syst. Man Cybern. Syst.* **55**(3), 2046–2061 (2025).
[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need," *Adv. Neural Inf. Process. Syst.* **30** (2017).
[34] Y. Wang, D. Wang, B. Zi and D. Zhang, "Force-position-workspace hybrid-based stability optimization of reconfigurable cable-driven parallel robot," *Robotica* **43**(6), 1–34 (2025).
[35] C. A. Griffiths, C. Giannetti, K. T. Andrzejewski and A. Morgan, "Comparison of a bat and genetic algorithm generated sequence against lead through programming when assembling a PCB using a six-axis robot with multiple motions and speeds," *IEEE Trans. Ind. Inform.* **18**(2), 1102–1110 (2022).

Cite this article: R. Wang, Y. Li and Y. Zeng (2026). "A gray-box identification method for cable-driven parallel robots via wavelet-enhanced neural networks", Robotica **44**, 612–629. https://doi.org/10.1017/S0263574726103221

https://doi.org/10.1017/S0263574726103221 Published online by Cambridge University Press