ENGINEERING Information Technology & Electronic Engineering
https://www.academax.com/EITEE; https://www.engineering.org.cn
https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=11344065
E-mail: eng.itee@zju.edu.cn

Engineering Journals
Available on IEEE Xplore®

Research Article
https://doi.org/10.1631/ENG.ITEE.2026.0055

A neural network algorithm for superstructure quadrilateral dynamic resistor networks on hammock surfaces with application to artificial intelligence

Yanpeng ZHENG[1], Guijie ZHANG[1✉], Xiaoyu JIANG[2✉], Zhaolin JIANG[3,4], Sung-Kwun OH[1,5]

[1]*School of Automation and Electrical Engineering, Linyi University, Linyi 276005, China*

[2]*School of Information Science and Engineering, Linyi University, Linyi 276005, China*

[3]*School of Intelligent Science and Control Engineering, Shandong Vocational and Technical University of International Studies, Rizhao 276826, China*

[4]*School of Mathematics and Statistics, Linyi University, Linyi 276005, China*

[5]*Division of Electrical and Electronic Engineering, The University of Suwon, Hwaseong 445-743, Republic of Korea*

Abstract: This study proposes a fast zeroing neural network algorithm to solve time-varying Laplacian linear systems arising from the modeling of superstructure quadrilateral dynamic resistor networks on hammock surfaces. By incorporating the intrinsic structure of the underlying special-form matrices into the core neurodynamic design, the proposed algorithm enables efficient real-time computation of electric potentials under dynamic conditions. The Lyapunov-based analysis proves global exponential convergence. Numerical simulations on resistor networks of various scales demonstrate high computational efficiency and verify convergence to solutions from arbitrary initial conditions. Furthermore, by integrating the proposed algorithm as a potential field solver with a directional potential field path planning algorithm and exploiting the natural descent property of resistor network node potentials, we propose a fast path planning algorithm for robotic navigation on hammock surfaces. Compared with conventional path planning approaches, the proposed algorithm achieves higher computational efficiency in the aforementioned hammock surface path planning task, and this advantage becomes increasingly pronounced as the scale increases. The proposed algorithm is also applied to dynamic path planning tasks, further validating its potential in robotics and control applications. Finally, we present two conjectures.

Key words: Zeroing neural network; Resistor network; Laplacian system; Equivalent resistance; Potential; Path planning

1 Introduction

Neural networks have emerged as an important computational framework for addressing complex real-time problems across scientific and engineering disciplines (Wang et al., 2019; Bapst et al., 2020; Wen et al., 2020; Gao et al., 2021; Wei et al., 2023). Among various neural networks, zeroing neural network (ZNN), which is a vector-error-based recurrent neural model proposed by Zhang YN et al. (2002), has been extensively investigated and applied owing to its dynamic problem-solving capabilities. Applications in rehabilitation robotics (Sun et al., 2023a, 2023b; Lian et al., 2024), discrete-time numerical systems (Shi and Zhang, 2020; Shi et al., 2022), and signal and image processing (Jin et al., 2024) further demonstrate the broad applicability of ZNNs.

A key strength of ZNNs is their ability to address time-varying problems, a domain where conventional solvers typically encounter limitations. This characteristic makes ZNNs suitable for dynamic physical systems with evolving parameters. Recent advancements, including fuzzy-enhanced discrete ZNNs (Qiu et al., 2024), triple-integral recurrent neural networks (Han et al., 2025), and refined discrete-time

✉ Guijie ZHANG, zhangguijie@lyu.edu.cn
✉ Xiaoyu JIANG, jxy19890422@sina.com
Yanpeng ZHENG, https://orcid.org/0000-0002-6874-9187
Guijie ZHANG, https://orcid.org/0009-0002-0247-1116
Xiaoyu JIANG, https://orcid.org/0000-0001-5857-9148
Zhaolin JIANG, https://orcid.org/0000-0002-9521-9835
Sung-Kwun OH, https://orcid.org/0000-0001-6798-8955

CLC number: TP183

Received: Feb. 26, 2026; Revision accepted: May 1, 2026;
Crosschecked: May 25, 2026; Published online: July 3, 2026

neurodynamic models (Qiu et al., 2022), have further improved stability and computational accuracy, reinforcing the effectiveness of ZNNs as a solution framework.

Traditionally, resistor networks have been used as fundamental models to investigate various physical, electronic, and material phenomena. Their controllability and repeatability make them particularly useful for simulating current flow in complex environments (Willke et al., 2017). Previous studies have investigated the effect of nonlinear topological circuits, which are based on the principles of linear topological circuits, on the behavior of optical and electronic devices (Hadad et al., 2018). In condensed matter physics, these circuits facilitate the analysis of the effect of surface defects on the resistance of topological insulators (Lüpke et al., 2017). In nanotechnology, they enable the characterization of transport processes at atomic scales (Willke et al., 2017). Notably, the foundational studies on graphene, which later led to a Nobel Prize, relied on lattice-inspired resistor network abstractions (Novoselov et al., 2004, 2005).

Beyond their roles in physics, resistor networks are extensively used in materials science and electronics. They facilitate defect identification in organic thin films (Darwish et al., 2018), enhance the performance of printed circuit board-integrated grating structures (Xu et al., 2021), and contribute to modeling techniques for microstrip reflect array designs (Hum and Du, 2017). Furthermore, numerical frameworks based on resistor networks have facilitated the quantitative performance evaluation of solid oxide fuel cells (Rhazaoui et al., 2013) and structural health monitoring in carbon fiber-reinforced polymer composites (Zhang D et al., 2021). Recently, they have also been applied to gamma-ray imaging systems (Jeon et al., 2021), illustrating their cross-domain versatility.

The theoretical analysis of resistor networks can be traced to Kirchhoff's circuit laws. These laws provide a formal basis for analyzing current and voltage distributions in electrical systems. Based on this foundation, Wu FY (2004) introduced a Laplacian operator framework for analytically computing resistances between arbitrary nodes in finite regular lattice networks. Although this method is efficient for symmetric structures, it has limitations for networks with irregular or complex topologies.

To extend the applicability of analytical models, researchers such as Izmailian et al. (2013) have introduced analytical techniques that explore the deeper algebraic properties of the Laplacian matrix, thereby extending the impedance modeling framework to complex and diverse network topologies. These advancements enable the derivation of closed-form expressions and efficient algorithms for specialized geometries, including globe, hammock, torus, and fan-shaped networks (Chair, 2014; Tan ZZ, 2015; Tan Z et al., 2018; Tan ZZ and Tan, 2020; Jiang ZL et al., 2023; Zhao WJ et al., 2023; Jiang XY et al., 2024, 2025b, 2026). Furthermore, the investigation by Chen et al. (2024) on the electrical properties of the $2 \times n$ nonregular hammock network extends the analytical boundaries of this research domain. Despite these contributions, many existing approaches assume static resistance parameters, limiting their effectiveness in practical time-dependent applications.

Recent studies report that higher-order topological dynamics can inspire physics-informed and computationally efficient artificial intelligence (AI) algorithms (Bianconi, 2021; Millán et al., 2025), and that metastructured quadrilateral lattices may emerge as important factors that influence future AI development (Bommes et al., 2013; Campen, 2017; Zhao H, 2025). In this study, the hammock resistor network is regarded as a metastructured quadrilateral lattice and the resulting neural network dynamical system is perceived as a form of topological shape dynamics on a higher-order network. From this perspective, the proposed formulation provides a potentially meaningful bridge between physics-inspired modeling and algorithm design; related advances in neural network algorithms provide methodological support for this line of research (Zhang ZJ et al., 2025).

This study develops a fast ZNN for hammock resistor networks (FZNNHRN). This algorithm is based on the structured zeroing neurodynamic framework developed for time-varying Laplacian systems in dynamic resistor networks by Jiang XY et al. (2025a). Coupling ZNN dynamics with the structural characteristics of hammock networks enables efficient real-time computation of potential distributions and equivalent resistances while adapting to resistance variations over time.

By integrating Laplacian-based resistor network modeling with fast neurodynamic computation, FZNNHRN provides a scalable real-time solution paradigm for practical systems with structural variability and temporally varying parameters. The primary contributions of this work are summarized as follows:

1. Model formulation. A fast ZNN, i.e., FZNNHRN, is proposed to solve the time-varying potential distribution and equivalent resistance in dynamic hammock surface metastructured quadrilateral resistor networks. This is the first attempt at the integration of ZNN dynamics with a hammock surface metastructured quadrilateral resistor network under time-varying resistances.

2. Computational efficiency. By exploiting the Laplacian matrix structure and sparse-matrix properties, along with fast matrix–vector multiplication, the FZNNHRN algorithm notably improves computational efficiency, thereby enabling real-time potential computation on large-scale resistor networks.

3. Simulation validation. By combining fast Laplacian matrix–vector multiplication with zeroing neurodynamics, FZNNHRN reduces the core computational complexity from $O(N^3)$ to $O(N\log_2 N)$ (N is the input size), thereby enabling real-time potential computation on large hammock resistor networks. The equivalent resistances computed by FZNNHRN closely match closed-form exact formulas, verifying the numerical accuracy of the solver. This represents a valuable contribution for solving large-scale, time-varying problems.

4. Potential applications. FZNNHRN has been applied to path planning in a hammock surface environment. By computing the node potentials, a potential field map is constructed, and the natural descending property of the electric potential is effectively embedded in the path planning algorithm design, resulting in an innovative approach for intelligent robotic navigation. The proposed framework is also extended to dynamic path planning scenarios. This achievement marks a critical step toward the interdisciplinary integration of superstructured quadrilateral resistor network theory and robotics.

5. Interdisciplinary perspective and methodological

insight. This study establishes a unifying framework at the intersection of high-order network topological shape dynamics, superstructured quadrilateral meshes, neural network algorithms, and hammock surface resistor network modeling. By bridging physical network structures with neurodynamic computation, this study provides insights into the modeling and analysis of complex systems in natural and engineered domains and a physics-inspired perspective that may facilitate the development of efficient and interpretable next-generation AI algorithms.

Fig. 1 presents the overall research framework of this study.

2 Mathematical model and equivalent resistance for hammock resistor networks

2.1 Hammock resistor networks

Hammock resistor networks, introduced by Essam et al. (2015), structurally comprise a planar rectangular grid comprising m horizontal chains (rows) and n vertical nodes (columns). Each interior node is connected to adjacent nodes via resistors: horizontal links have resistance s, and vertical connections have resistance r. The boundaries are open at the top and bottom, and each row terminates at fixed endpoints on both sides.

As shown in Fig. 2, the standard configuration comprises $m=10$ horizontal conductor lines and $n=10$ vertical conductor lines, which correspond to the rows and columns of the hammock resistor network, respectively. The top and bottom terminals are labeled as nodes O' and O, respectively. To evaluate the equivalent resistance between arbitrary nodes, the bottom node O is selected as the reference node, and its Laplacian potential is set to zero to fix the solution.

The mathematical formulation of the hammock resistor network is obtained according to Kirchhoff's laws as follows (Essam et al., 2015):

$$\boldsymbol{L}_{mn}^{\text{hammock}}\boldsymbol{V}=\boldsymbol{\mathcal{I}}, \tag{1}$$

with

$$\boldsymbol{V}=[V_1,V_2,\ldots,V_{mn}]^{\mathrm{T}},$$
$$\boldsymbol{\mathcal{I}}=[\mathcal{I}_1,\mathcal{I}_2,\ldots,\mathcal{I}_{mn}]^{\mathrm{T}},$$

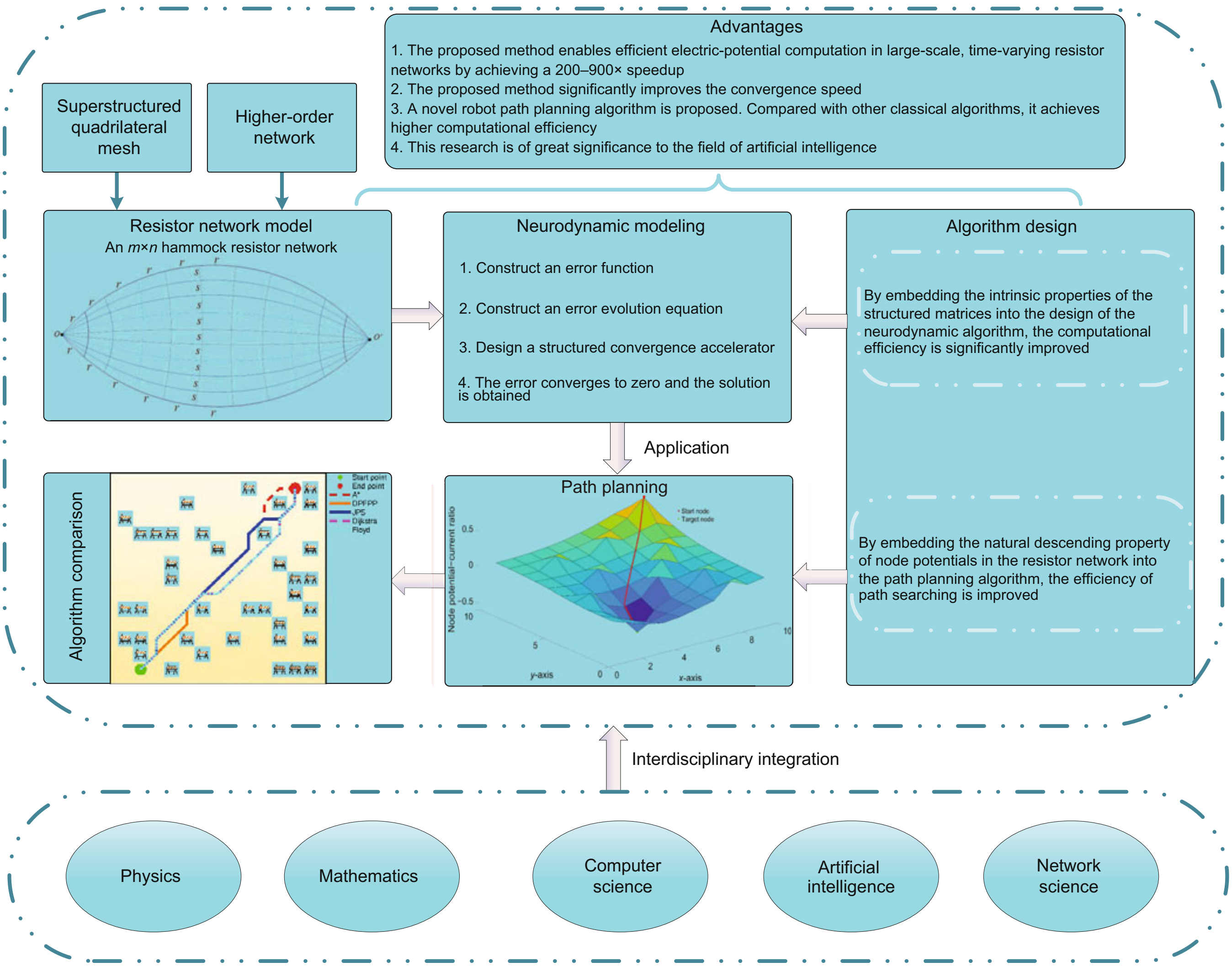

Fig. 1 Overall framework of this study: modeling, neurodynamic solver, fast algorithms, and applications. The hammock resistor network is rotated by 90° clockwise for clarity and layout

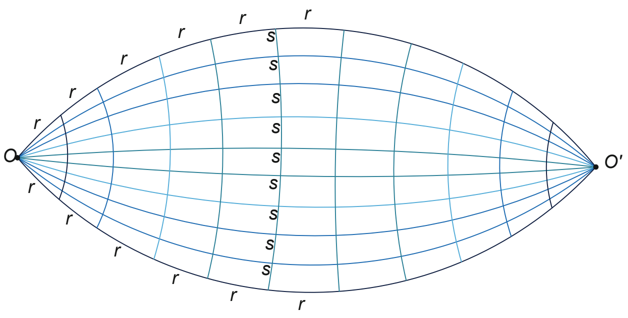

Fig. 2 Structure of the hammock resistor network with $m = n = 10$, where vertical and horizontal resistances are denoted by r and s, respectively. The hammock resistor network is rotated by 90° clockwise for clarity and layout

where V_i denotes the electric potential at the i^{th} node, $\mathcal{I}_i$ denotes the corresponding current at the i^{th} node, and $\boldsymbol{L}_{mn}^{\text{hammock}}$ denotes a matrix of dimension $mn \times mn$. The matrix is defined as

$$\boldsymbol{L}_{mn}^{\text{hammock}} = s^{-1}\boldsymbol{L}_m^{\text{DD}} \otimes \boldsymbol{E}_n + r^{-1}\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}}, \tag{2}$$

where $\boldsymbol{E}_n$ and $\boldsymbol{E}_m$ denote identity matrices with dimensions $n \times n$ and $m \times m$, respectively. $\otimes$ is the Kronecker product. The resistances along the horizontal and vertical directions are given by real values s and r, respectively ($s, r \in \mathbb{R}$), and their reciprocals are denoted by s^{-1} and r^{-1}. Matrix $\boldsymbol{L}_n^{\text{free}}$ denotes the Laplacian of a one-dimensional (1D) lattice with free boundary conditions, and $\boldsymbol{L}_m^{\text{DD}}$ denotes the 1D Laplacian with Dirichlet boundary conditions at both ends.

$$\boldsymbol{L}_m^{\text{DD}} = \boldsymbol{L}_m^{\text{per}} + \boldsymbol{K}_m, \quad \boldsymbol{K}_m = \boldsymbol{e}_1\boldsymbol{e}_m^{\text{T}} + \boldsymbol{e}_m\boldsymbol{e}_1^{\text{T}},$$

$$\boldsymbol{L}_n^{\text{free}} = \text{tridiag}(-1,\, 2,\, -1) - \left(\boldsymbol{e}_1\boldsymbol{e}_1^{\text{T}} + \boldsymbol{e}_n\boldsymbol{e}_n^{\text{T}}\right),$$

where matrix $\boldsymbol{L}_m^{\text{per}}$ denotes the Laplacian of a 1D lattice with periodic boundary conditions and $\boldsymbol{e}_i$ denotes the i^{th} standard basis column vector of appropriate dimension.

Due to the absence of current sources and sinks, the system obeys the constraint $\sum_{i=1}^{mn} \mathcal{I}_i = 0$.

2.2 Dynamic resistor networks on hammock surfaces

In practical engineering, the electrical resistance of conductive materials typically exhibits temporal variations owing to environmental factors (Greiner, 1998). A linear approach is adopted to model this behavior to describe such variations. According to Eq. (1), the dynamic behavior of the hammock resistor network can be described using the following expressions:

$$s(t) = s_0\big(1 + \rho\,|\sin t|\big), \ \ r(t) = r_0\left(1 + \rho\,|\sin t|\right), \tag{3}$$

where s_0 and r_0 denote the initial resistivity values in the horizontal and vertical directions, respectively. The parameter ρ indicates the rate at which resistivity changes over time, with t representing the temporal variable.

Dynamic resistor networks on hammock surfaces are governed by the following mathematical model:

$$\boldsymbol{L}_{mn}^{\text{hammock}}(t)\boldsymbol{V}(t) = \boldsymbol{\mathcal{I}}, \tag{4}$$

where

$$\boldsymbol{L}_{mn}^{\text{hammock}}(t) = s^{-1}(t)\big(\boldsymbol{L}_m^{\text{per}} + \boldsymbol{K}_m\big) \otimes \boldsymbol{E}_n + r^{-1}(t)\,\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}}, \tag{5}$$

and $\boldsymbol{V}(t)$ represents the electric potential as a function of time.

2.3 Exact formula for equivalent resistance

To compute the equivalent resistance $R_{\tau,\phi}$ between arbitrary nodes τ and ϕ, an external voltage source is applied across them, with all other nodes left unconnected to external sources. We denote the electric potentials at nodes τ and ϕ as V_τ and V_ϕ, respectively, and the resulting current through the source as $\mathcal{I}_{\tau,\phi}$. By Ohm's law, the equivalent resistance is expressed as

$$R_{\tau,\phi} = \frac{V_\tau - V_\phi}{\mathcal{I}_{\tau,\phi}}. \tag{6}$$

Excluding the top terminal O' and bottom terminal O, the equivalent resistance $R_{\text{hammock}}(r_1, r_2)$ between two arbitrary nodes $r_1 = (x_1, y_1)$ and $r_2 = (x_2, y_2)$ within the hammock resistor network can be described as follows:

$$\begin{aligned} R_{\text{hammock}}(r_1, r_2) &= \frac{r\,(y_2 - y_1)^2}{n(m+1)} \\ &+ \frac{2s}{m+1}\left(\sum_{i=2}^{m+1} \frac{\alpha' S_i^2(y_1) - 2\beta' S_i(y_2)S_i(y_1) + \gamma' S_i^2(y_2)}{\sinh(2l_i)\,\sinh(2nl_i)}\right), \end{aligned} \tag{7}$$

where $S_i(y) = \sin\dfrac{(i-1)\pi y}{m+1}$, the coefficients are defined as

$$\begin{cases} \alpha' = \cosh\left((2n - 2x_1 + 1)l_i\right)\cosh\left((2x_1 - 1)l_i\right), \\ \beta' = \cosh\left((2n - 2x_2 + 1)l_i\right)\cosh\left((2x_1 - 1)l_i\right), \\ \gamma' = \cosh\left((2n - 2x_2 + 1)l_i\right)\cosh\left((2x_2 - 1)l_i\right), \end{cases} \tag{8}$$

and the quantity l_i used in the above hyperbolic functions can be alternatively expressed in either of the following equivalent forms:

$$l_i = \frac{1}{2}\ln\lambda_i' \quad \text{or} \quad \cosh(2l_i) = \frac{1}{2}u_i. \tag{9}$$

The eigenvalue-related term u_i is obtained from

$$u_i = 2 + 2h\left(1 - \cos\frac{(i-1)\pi}{m+1}\right), \tag{10}$$

where $h = s/r$. λ_i' denotes the greater root of the characteristic equation of the hammock resistor network:

$$\lambda'^2 - u_i\lambda' + 1 = 0, \tag{11}$$

where λ is the root variable.

When two nodes lie on the same radial line in the hammock resistor network, denoted as $r_1 = (x, y_1)$ and $r_2 = (x, y_2)$, the resistance $R_{\text{hammock}}(r_1, r_2)$ can be defined as follows:

$$\begin{aligned} R_{\text{hammock}}(r_1, r_2) = &\frac{2s}{m+1}\sum_{i=2}^{m+1}\left(S_i(y_2) - S_i(y_1)\right)^2 \\ &\cdot \frac{\cosh((2x-1)l_i)\cosh((2n-2x+1)l_i)}{\sinh(2l_i)\sinh(2nl_i)} \\ &+ \frac{r(y_2 - y_1)^2}{n(m+1)}. \end{aligned} \tag{12}$$

When two nodes in the hammock resistor network lie on the same horizontal line, denoted as $r_1 = (x_1, y)$ and

$r_2 = (x_2, y)$, the resistance $R_{\text{hammock}}(r_1, r_2)$ can be defined as follows:

$$R_{\text{hammock}}(r_1, r_2) = \frac{4s}{m+1}\sum_{i=2}^{m+1}\frac{\sinh((x_2-x_1)l_i)}{\sinh(2l_i)} \cdot \frac{\cosh((n-x_2+x_1)l_i)\ S_i^2(y)}{\cosh(nl_i)}. \tag{13}$$

3 Solving for the electric potential of the hammock resistor network using ZNNs

This section presents the definition of the vector-valued error function:

$$\boldsymbol{E}(t) = \boldsymbol{L}_{mn}^{\text{hammock}}(t)\boldsymbol{V}(t) - \boldsymbol{\mathcal{I}}, \tag{14}$$

where $\boldsymbol{L}_{mn}^{\text{hammock}}(t)$ is defined in Eq. (5) and $\boldsymbol{V}(t) \in \mathbb{R}^{mn}$. When the error function asymptotically approaches zero, $\boldsymbol{V}(t)$ may be considered an exact solution of Eq. (4).

To achieve the convergence of $\frac{\mathrm{d}\boldsymbol{E}(t)}{\mathrm{d}t}$ to zero, we adopt the gradient descent method (Zhang YN et al., 2002):

$$\frac{\mathrm{d}\boldsymbol{E}(t)}{\mathrm{d}t} = -\boldsymbol{\kappa}(t)\boldsymbol{\mathcal{F}}(\boldsymbol{E}(t)), \tag{15}$$

where $\boldsymbol{\kappa}(t) \in \mathbb{R}^{mn}$ denotes a positive definite matrix that regulates the rate of convergence and $\boldsymbol{\mathcal{F}}(\cdot) : \mathbb{R}^{mn} \to \mathbb{R}^{mn}$ denotes a vector-valued activation function.

To enhance the convergence performance of the model, we define $\boldsymbol{\kappa}(t) = \gamma\left(\left(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\right)^{\mathrm{T}}\boldsymbol{L}_{mn}^{\text{hammock}}(t) + \lambda\boldsymbol{E}_{mn}\right)^k$, where $\boldsymbol{E}_{mn}$ denotes the mn-dimensional identity matrix. The parameters γ, λ, and k must satisfy the conditions $\gamma > 0$, $\lambda \geq 1$, and $k \geq 1$. The improved model is named FZNNHRN. Through proper tuning of the design parameters γ, λ, and k, the convergence rate and precision of the system can be effectively controlled:

$$\begin{aligned}\boldsymbol{L}_{mn}^{\text{hammock}}(t)\dot{\boldsymbol{V}}(t) = &-\boldsymbol{\kappa}(t)\,\boldsymbol{\mathcal{F}}\Big(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\boldsymbol{V}(t) - \boldsymbol{\mathcal{I}}\Big)\\ &-\dot{\boldsymbol{L}}_{mn}^{\text{hammock}}(t)\boldsymbol{V}(t).\end{aligned} \tag{16}$$

The overall performance of neural networks is considerably affected by the choice of the activation function $\boldsymbol{\mathcal{F}}(\cdot)$; several representative types are summarized in Section S1 of the supplementary materials. Furthermore, detailed scalar-form dynamics of individual neurons and schematics of hardware-oriented circuits are provided in Section S2 of the supplementary materials.

4 Development of the fast algorithm for FZNNHRN

Fast algorithms are essential in computer science. They provide significant improvements in computational efficiency and resource utilization. Additionally, they help meet stringent performance requirements, particularly in large-scale data processing tasks. In this section, we focus on leveraging the specific structural characteristics of $\boldsymbol{L}_{mn}^{\text{hammock}}(t)$ to develop an efficient algorithm specifically adapted to the FZNNHRN model.

4.1 Efficient computation of $\boldsymbol{L}_{mn}^{\text{hammock}}(t)\,\boldsymbol{V}(t)$

Given matrix $\boldsymbol{L}_{mn}^{\text{hammock}}(t)$ and time-dependent vector $\boldsymbol{V}(t)$, let $\boldsymbol{\nu} = \boldsymbol{V}_i(t^*) = [V_1(t^*), V_2(t^*), \ldots, V_{mn}(t^*)]^{\mathrm{T}}$ denote the instantaneous state at time t^*. To obtain matrix–vector product $\boldsymbol{L}_{mn}^{\text{hammock}}(t)\boldsymbol{\nu}$, individual terms $(\boldsymbol{L}_m^{\text{per}} \otimes \boldsymbol{E}_n)\boldsymbol{\nu}$, $(\boldsymbol{K}_m \otimes \boldsymbol{E}_n)\boldsymbol{\nu}$, and $(\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}})\boldsymbol{\nu}$ can be evaluated independently. $\boldsymbol{L}_m^{\text{per}} \otimes \boldsymbol{E}_n$ denotes a circulant matrix uniquely determined by its first row. $\boldsymbol{K}_m \otimes \boldsymbol{E}_n$ denotes a sparse matrix, with most of its elements being zero, which facilitates fast computation. $\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}}$ exhibits a block diagonal structure. Based on the structural properties of these matrices and by leveraging accelerated computation for special-form matrices, we propose efficient techniques for computing matrix–vector multiplications. $\boldsymbol{L}_m^{\text{per}} \otimes \boldsymbol{E}_n$ is constructed by applying a circular shift to its first column $\boldsymbol{c}$, where vector $\boldsymbol{c}$ takes the following form:

$$\boldsymbol{c} = \begin{cases} 2\,\boldsymbol{e}_1 - \boldsymbol{e}_{n+1}, & m = 2,\\ 2\,\boldsymbol{e}_1 - \boldsymbol{e}_{n+1} - \boldsymbol{e}_{mn-n+1}, & m > 2,\end{cases} \tag{17}$$

where $\boldsymbol{e}_i \in \mathbb{R}^{mn}$ denotes the i^{th} standard basis vector (i.e., the vector whose i^{th} entry equals 1 and all other entries equal 0).

The efficient method for computing $(\boldsymbol{L}_m^{\text{per}} \otimes \boldsymbol{E}_n)\boldsymbol{\nu}$ is described in Algorithm 1.

Algorithm 1 Fast algorithm for matrix–vector multiplication $(\boldsymbol{L}_m^{\text{per}} \otimes \boldsymbol{E}_n)\boldsymbol{\nu}$

1: Compute $\boldsymbol{\omega} = \mathrm{FFT}(\boldsymbol{\nu})$ // FFT: the fast Fourier transform
2: Compute $\boldsymbol{d} = \mathrm{FFT}(\boldsymbol{c})$
3: Compute the element-wise vector product: $\boldsymbol{\varpi} = \boldsymbol{\omega} \cdot \boldsymbol{d}$
4: Compute the final result: $(\boldsymbol{L}_m^{\text{per}} \otimes \boldsymbol{E}_n)\boldsymbol{\nu} = \mathrm{IFFT}(\boldsymbol{\varpi})$ /* IFFT: the inverse fast Fourier transform */

Matrix–vector product $(\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}})\boldsymbol{\nu}$ can be viewed as performing m independent matrix–vector multiplications, where each sub-block $\boldsymbol{L}_n^{\text{free}}$ is separately multiplied by the vector $\boldsymbol{\xi}_i$:

$$(\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}})\boldsymbol{\nu} = \sum_{i=1}^{m}\Big(\boldsymbol{e}_i \otimes \boldsymbol{L}_n^{\text{free}}\boldsymbol{\xi}_i\Big), \tag{18}$$

where $\boldsymbol{\nu}$ is partitioned into m blocks, represented by $\boldsymbol{\xi}_i = [\boldsymbol{\nu}_{(i-1)n+1}, \boldsymbol{\nu}_{(i-1)n+2}, \ldots, \boldsymbol{\nu}_{(i-1)n+n}]^{\mathrm{T}}$, $i = 1, 2, \ldots, m$.

Upon computing each $\boldsymbol{z}_i = \boldsymbol{L}_n^{\text{free}}\boldsymbol{\xi}_i$, $(\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}})\boldsymbol{\nu}$ can be obtained by arranging the m vectors $\boldsymbol{z}_i$ in order (Algorithm 2).

Algorithm 2 Fast algorithm for $(\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}})\boldsymbol{\nu}$

1: **for** $i = 1, 2, \ldots, m$ **do**
2: $\quad\boldsymbol{z}_i(1) = 2\boldsymbol{\xi}_i(1) - \boldsymbol{\xi}_i(2)$
3: $\quad$**for** $j = 2, 3, \ldots, n-1$ **do**
4: $\quad\quad\boldsymbol{z}_i(j) = -\boldsymbol{\xi}_i(j-1) + 2\boldsymbol{\xi}_i(j) - \boldsymbol{\xi}_i(j+1)$
5: $\quad$**end for**
6: $\quad\boldsymbol{z}_i(n) = -\boldsymbol{\xi}_i(n-1) + \boldsymbol{\xi}_i(n)$
7: **end for**
8: Concatenate m sub-vectors $\boldsymbol{z}_i$: $\left(\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}}\right)\boldsymbol{\nu} = \left[\boldsymbol{z}_1^{\mathrm{T}}, \boldsymbol{z}_2^{\mathrm{T}}, \ldots, \boldsymbol{z}_m^{\mathrm{T}}\right]^{\mathrm{T}}$

By applying Algorithms 1 and 2 and

$$\begin{aligned}\boldsymbol{L}_{mn}^{\text{hammock}}(t)\boldsymbol{V}(t) = &s^{-1}(t)\left((\boldsymbol{L}_m^{\text{per}} + \boldsymbol{K}_m) \otimes \boldsymbol{E}_n\right)\boldsymbol{V}(t)\\ &+ r^{-1}(t)\,\Big(\boldsymbol{E}_m \otimes \boldsymbol{L}_n^{\text{free}}\Big)\boldsymbol{V}(t),\end{aligned} \tag{19}$$

the fast computation of $\boldsymbol{L}_{mn}^{\text{hammock}}(t)\boldsymbol{V}(t)$ is achieved. The computation of $\dot{\boldsymbol{L}}_{mn}^{\text{hammock}}(t)\boldsymbol{V}(t)$ requires the initial differentiation of $\boldsymbol{L}_{mn}^{\text{hammock}}(t)$; all remaining computational steps are identical to those used for $\boldsymbol{L}_{mn}^{\text{hammock}}(t)\boldsymbol{V}(t)$.

4.2 Efficient computation of matrix powers

The matrix $\left(\left(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\right)^{\mathrm{T}}\boldsymbol{L}_{mn}^{\text{hammock}}(t)+\lambda\boldsymbol{E}_{mn}\right)^{k}$ and activated error vector $\boldsymbol{\mathcal{F}}(\cdot)=[f_1,f_2,\ldots,f_{mn}]^{\mathrm{T}}$ (f_i is the scalar activation function of a single neuron) aid in developing a fast algorithm as follows:

Laplacian matrix $\boldsymbol{L}_{mn}^{\text{hammock}}(t)$ can be diagonalized via a similarity transformation using the transition matrix

$$\boldsymbol{\mathcal{T}}^{-1}\boldsymbol{L}_{mn}^{\text{hammock}}\boldsymbol{\mathcal{T}}=s^{-1}\boldsymbol{\Lambda}_m\otimes\boldsymbol{E}_n+r^{-1}\boldsymbol{E}_m\otimes\boldsymbol{\Lambda}_n\triangleq\boldsymbol{\Lambda}_{mn},\tag{20}$$

where $\boldsymbol{\mathcal{T}}\triangleq\mathbb{S}_m^{\mathrm{I}}\otimes\mathbb{C}_n^{\mathrm{III}}$ (Pei and Yeh, 2001; Garcia and Yih, 2018).

$\mathbb{S}_m^{\mathrm{I}}$ is the type-I discrete sine transform (DST-I), defined as

$$\mathbb{S}_m^{\mathrm{I}}=\left(\sqrt{\frac{2}{m+1}}\sin\frac{jk\pi}{m+1}\right)_{k,j=1}^{m},\tag{21}$$

and $\mathbb{C}_n^{\mathrm{III}}$ is the type-III discrete cosine transform (DCT-III), defined as

$$\mathbb{C}_n^{\mathrm{III}}=\left(\sqrt{\frac{2}{n}}d_k\cos\frac{(k-1)(2j-1)\pi}{2n}\right)_{k,j=1}^{n},\tag{22}$$

with $d_k=\min\left\{1,\frac{\sqrt{2}}{2}-1+k\right\}$.

Matrices $\boldsymbol{L}_m^{\mathrm{DD}}$ and $\boldsymbol{L}_n^{\text{free}}$ can be transformed into diagonal matrices $\boldsymbol{\Lambda}_m$ and $\boldsymbol{\Lambda}_n$, respectively, via similarity transformations:

$$\begin{cases}\boldsymbol{\Lambda}_m=\operatorname{diag}\left(\lambda_k'\right)_{k=0}^{m-1},\ \lambda_k'=2-2\cos\dfrac{(k+1)\pi}{m+1},\\[2mm]\boldsymbol{\Lambda}_n=\operatorname{diag}\left(\lambda_j\right)_{j=0}^{n-1},\ \lambda_j=2-2\cos\dfrac{j\pi}{n}.\end{cases}\tag{23}$$

$\boldsymbol{L}_{mn}^{\text{hammock}}(t)$ is equivalent to the diagonal matrix $\boldsymbol{\Lambda}_{mn}$ under a similarity transformation:

$$\boldsymbol{\Lambda}_{mn}(t)=\operatorname{diag}\big(\lambda_{0,0}(t),\lambda_{0,1}(t),\ldots,\ \lambda_{m-1,n-1}(t)\big),\tag{24}$$

where $\lambda_{k,j}=s^{-1}(t)\lambda_k'+r^{-1}(t)\lambda_j$.

According to Eq. (20), $\boldsymbol{L}_{mn}^{\text{hammock}}(t)$ can be expressed as follows:

$$\begin{aligned}\boldsymbol{L}_{mn}^{\text{hammock}}&=\left(\mathbb{S}_m^{\mathrm{I}}\otimes\mathbb{C}_n^{\mathrm{III}}\right)\boldsymbol{\Lambda}_{mn}\left(\mathbb{S}_m^{\mathrm{I}}\otimes\mathbb{C}_n^{\mathrm{III}}\right)^{-1}\\&=\left(\mathbb{S}_m^{\mathrm{I}}\otimes\boldsymbol{E}_n\right)\left(\boldsymbol{E}_m\otimes\mathbb{C}_n^{\mathrm{III}}\right)\boldsymbol{\Lambda}_{mn}\\&\quad\cdot\left(\mathbb{S}_m^{\mathrm{I}}\otimes\boldsymbol{E}_n\right)\left(\boldsymbol{E}_m\otimes\left(\mathbb{C}_n^{\mathrm{III}}\right)^{-1}\right).\end{aligned}\tag{25}$$

Since $\boldsymbol{L}_{mn}^{\text{hammock}}$ is a symmetric matrix, we have $\boldsymbol{L}_{mn}^{\text{hammock}}\left(\boldsymbol{L}_{mn}^{\text{hammock}}\right)^{\mathrm{T}}=\left(\boldsymbol{L}_{mn}^{\text{hammock}}\right)^{2}$. Thus, we obtain the following equation:

$$\begin{aligned}\left(\left(\boldsymbol{L}_{mn}^{\text{hammock}}\right)^{2}+\lambda\boldsymbol{E}_{mn}\right)^{k}&=\left(\mathbb{S}_m^{\mathrm{I}}\otimes\boldsymbol{E}_n\right)\left(\boldsymbol{E}_m\otimes\mathbb{C}_n^{\mathrm{III}}\right)\\&\quad\cdot\boldsymbol{M}^{k}\left(\mathbb{S}_m^{\mathrm{I}}\otimes\boldsymbol{E}_n\right)\left(\boldsymbol{E}_m\otimes\left(\mathbb{C}_n^{\mathrm{III}}\right)^{-1}\right).\end{aligned}\tag{26}$$

$\boldsymbol{M}=\boldsymbol{\Lambda}_{mn}^{2}+\lambda\boldsymbol{E}_{mn}$ denotes a diagonal matrix:

$$\boldsymbol{M}(t)=\operatorname{diag}\big(\lambda_{0,0}^{2}(t),\lambda_{0,1}^{2}(t),\ldots,\lambda_{m-1,n-1}^{2}(t)\big)+\lambda\boldsymbol{E}_{mn}.\tag{27}$$

The computation of matrix–vector product $\left(\left(\mathbf{L}_{mn}^{\text{hammock}}\right)^{2}+\lambda\boldsymbol{E}_{mn}\right)^{k}\boldsymbol{\mathcal{F}}(\cdot)$ begins with the evaluation of $\boldsymbol{\varsigma}_{mn}=\left(\boldsymbol{E}_m\otimes\left(\mathbb{C}_n^{\mathrm{III}}\right)^{-1}\right)\boldsymbol{\mathcal{F}}(\cdot)$:

$$\boldsymbol{\varsigma}_{mn}=\sum_{i=1}^{m}\left(\boldsymbol{e}_i\otimes(\mathbb{C}_n^{\mathrm{III}})^{-1}\boldsymbol{\chi}_i\right),\tag{28}$$

where

$$\boldsymbol{\chi}_i=[f_{i+1},f_{i+2},\ldots,f_{i+n}],\ i=1,2,\ldots,m.\tag{29}$$

When $\boldsymbol{\varsigma}_{mn}=\left(\boldsymbol{E}_m\otimes\left(\mathbb{C}_n^{\mathrm{III}}\right)^{-1}\right)\boldsymbol{\chi}_i$ is evaluated, the expression $\left(\boldsymbol{E}_m\otimes\left(\mathbb{C}_n^{\mathrm{III}}\right)^{-1}\right)\boldsymbol{\mathcal{F}}(\cdot)$ can be obtained by sequentially assembling the corresponding m components. Note that $\left(\mathbb{C}_n^{\mathrm{III}}\right)^{-1}\boldsymbol{\chi}_i$ allows for fast computation via the inverse of DCT-III (IDCT-III).

$$\begin{aligned}\left(\mathbb{S}_m^{\mathrm{I}}\otimes\boldsymbol{E}_n\right)\boldsymbol{\varsigma}_{mn}&=\left(\mathbb{S}_m^{\mathrm{I}}\otimes\boldsymbol{E}_n\right)\sum_{j=1}^{n}\left(\boldsymbol{e}_{m,j}\otimes\boldsymbol{\varsigma}_{j,m}\right)\\&=\sum_{j=1}^{n}\left(\left(\mathbb{S}_m^{\mathrm{I}}\boldsymbol{e}_{m,j}\right)\otimes\boldsymbol{\varsigma}_{j,m}\right),\end{aligned}\tag{30}$$

where $\boldsymbol{e}_{m,j}$ represents m-dimensional linearly independent unit vectors and $\boldsymbol{\varsigma}_{j,m}$ denotes the transpose of the j^{th} column vector of $\boldsymbol{\varsigma}_{mn}$, $j=1,2,\ldots,n$. Notably, the product $\left(\mathbb{S}_m^{\mathrm{I}}\right)\boldsymbol{e}_{m,j}$ provides the j^{th} column of $\mathbb{S}_m^{\mathrm{I}}$, facilitating highly efficient computation. Let $\boldsymbol{b}_j$ represent the vector obtained after processing. Consequently, the multiplication of matrix $\boldsymbol{M}^{k}(t)$ and vector $\boldsymbol{b}_j$ corresponds to an element-wise vector operation.

Similarly, we perform DCT-III by applying the approach described in Eq. (29) and arrange the resulting vectors sequentially to obtain a new vector $\boldsymbol{\zeta}_j$. Following the approach in Eq. (30), the matrix–vector product $\left(\mathbb{S}_m^{\mathrm{I}}\otimes\boldsymbol{E}_n\right)\boldsymbol{\zeta}_j$ can be computed as $\left(\left(\boldsymbol{L}_{mn}^{\text{hammock}}\right)^{2}+\lambda\boldsymbol{E}_{mn}\right)^{k}\boldsymbol{\mathcal{F}}(\cdot)$. A fast algorithm is obtained in Algorithm 3.

Algorithm 3 Efficient computation of matrix–vector products involving $\left((\boldsymbol{L}_{mn}^{\text{hammock}})^{2}+\lambda\boldsymbol{E}_{mn}\right)^{k}\boldsymbol{\mathcal{F}}(\cdot)$

for $i=1,2,\ldots,m$ **do**
  $\boldsymbol{\chi}_i=[f_{i+1},f_{i+2},\ldots,f_{i+n}]$
  $\boldsymbol{\varsigma}_i=\left(\mathbb{C}_n^{\mathrm{III}}\right)^{-1}\boldsymbol{\chi}_i$
end for
Concatenate all $\boldsymbol{\varsigma}_i$ to form $\boldsymbol{\varsigma}=\left[\boldsymbol{\varsigma}_1^{\mathrm{T}},\boldsymbol{\varsigma}_2^{\mathrm{T}},\ldots,\boldsymbol{\varsigma}_m^{\mathrm{T}}\right]^{\mathrm{T}}$
for $j=1,2,\ldots,n$ **do**
  $\boldsymbol{\vartheta}_j=\left[\boldsymbol{\varsigma}_j,\boldsymbol{\varsigma}_{j+n},\ldots,\boldsymbol{\varsigma}_{j+(m-1)n}\right]$
  $\boldsymbol{b}_j=\mathbb{S}_m^{\mathrm{I}}(\boldsymbol{\vartheta}_j)$
end for
Concatenate the n vectors $\boldsymbol{b}_j$ to obtain $\boldsymbol{b}$
Compute the element-wise product: $\boldsymbol{\sigma}=\boldsymbol{M}^{k}(t)\boldsymbol{b}$
for $i=1,2,\ldots,m$ **do**
  $\boldsymbol{\theta}_i=[\boldsymbol{\sigma}_{i+1},\boldsymbol{\sigma}_{i+2},\ldots,\boldsymbol{\sigma}_{i+n}]$
  $\boldsymbol{\zeta}_i=\mathbb{C}_n^{\mathrm{III}}\boldsymbol{\theta}_i$
end for
Concatenate all $\boldsymbol{\zeta}_i$ to form $\boldsymbol{\zeta}=\left[\boldsymbol{\zeta}_1^{\mathrm{T}},\ \boldsymbol{\zeta}_2^{\mathrm{T}},\ \ldots,\ \boldsymbol{\zeta}_m^{\mathrm{T}}\right]^{\mathrm{T}}$
for $j=1,2,\ldots,n$ **do**
  $\boldsymbol{\iota}_j=\left[\boldsymbol{\zeta}_j,\boldsymbol{\zeta}_{j+n},\ldots,\boldsymbol{\zeta}_{j+(m-1)n}\right]$
  $\boldsymbol{\Upsilon}_j=\mathbb{S}_m^{\mathrm{I}}\boldsymbol{\iota}_j$
end for
Concatenate the n vectors $\boldsymbol{\Upsilon}_j$ to obtain the final result

4.3 Comparison of computational efficiency

Typically, traditional applications of matrix–vector multiplication, such as computing it directly in MATLAB, incur high computational costs, with complexities of $O(N^2)$ for the submodules $\boldsymbol{L}_{mn}^{\text{hammock}}\boldsymbol{v}$ and $O(N^3)$ for the full matrix expression in Eq. (25), where $N=mn$. In contrast, FZNNHRN considerably reduces this burden. Specifically, Algorithms 1 and 2 achieve complexities of $O(N\log_2 N)$ and $O(N)$, respectively. Algorithm 3, which incorporates DCT-III, IDCT-III, and DST-I transforms (Pei and Yeh, 2001), operates at $O(N\log_2 N)$ complexity. Consequently, the right-hand side of the implicit dynamical Eq. (25) can now be computed with high efficiency in $O(N\log_2 N)$ complexity instead of the original $O(N^3)$ complexity. The FZNNHRN solver not only reduces computational complexity but also lowers memory requirements, thereby enabling scalable computation of large-scale resistor networks.

To assess the performance of FZNNHRN, simulations are conducted on resistor networks of varying sizes and compared with the improved recurrent neural network (IMRNN) (Wu WQ and Zheng, 2020) (i.e., Eq. (16)). Both models are evaluated under identical settings (Table 1), with $(\gamma,\lambda,k)=(3,2,2)$, and a linear activation function (LAF) is applied. The simulation duration for these two models is set to $t=0.3$ s, by which point the models have converged. In the resistor networks, the horizontal and vertical resistivities are defined as time-varying functions $s=r=1+0.004|\sin t|$. Fig. 3 presents a comparative runtime analysis, where FZNNHRN consistently exhibits superior computational efficiency over IMRNN under the same hardware and parameter settings. Specifically, FZNNHRN achieves a speedup ranging from approximately 400 to 3000 times. Furthermore, the computational advantage of FZNNHRN becomes pronounced as the network scales up.

Table 1 Variable settings for different neural models under special cases

Method	Act.	γ	λ	k	t (s)
ZNN	LAF	3	–	–	3
SAND	LAF	3	2	–	5
IMRNN	LAF	3	2	2	0.3
FZNNHRN	LAF	3	2	2	0.3

–: the parameters are not used; Act.: activation function

Fig. 4 presents a runtime comparison between FZNNHRN and two benchmark algorithms, ZNN (Zhang YN et al., 2002) and the saturation-allowed neural dynamics (SAND) (Jin et al., 2021), which are widely applied in general linear systems. All parameter configurations are detailed in Table 1. Performance evaluation is performed using resistor networks of different sizes. Corresponding results are visualized through line plots.

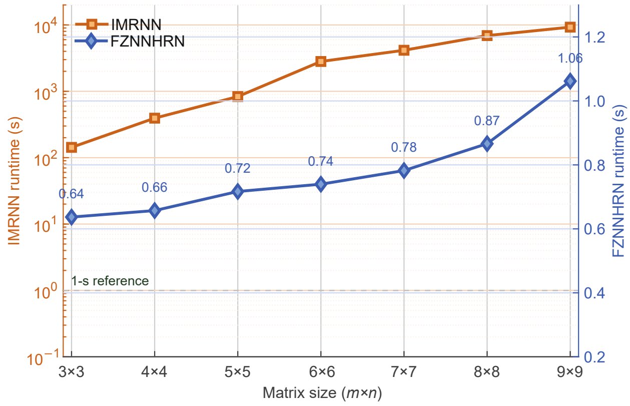

Fig. 3 CPU runtime comparison between IMRNN and FZNNHRN

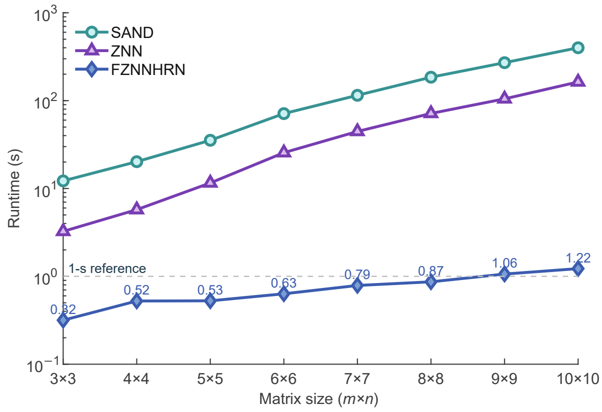

Fig. 4 CPU runtime comparison between FZNNHRN and other methods

The data reveal a consistent advantage of FZNNHRN in terms of computational speed across all tested scales. As the network size increases, the performance gap between FZNNHRN and the other two methods notably widens, highlighting the superior efficiency of FZNNHRN in handling large-scale problems.

Tables 2 and 3 summarize the central processing unit (CPU) runtime performance of various methods applied to dynamic resistor networks on hammock surfaces of increasing scale. From the results, several noteworthy findings can be derived:

Table 2 Comparison of execution time across different algorithms at various network scales

Algorithm	Execution time (s)				
	3×3	5×5	10×10	15×15	20×20
IMRNN	142.965 951	832.873 119	> 60	–	–
ZNN	3.248 444	11.561 791	163.138 395	780.358 618	> 60
SAND	6.592 436	18.627 017	211.201 810	1097.315 839	> 60
FZNNHRN	**0.636 861**	**0.716 805**	**1.224 283**	**3.457 068**	**18.154 461**

–: not evaluated. The best results are in bold

Table 3 Runtime analysis of the FZNNHRN model for increasing network scales

Algorithm	Runtime (s)				
	30 × 30	40 × 40	50 × 50	60 × 60	70 × 70
FZNNHRN	71.181 507	208.507 147	487.341 039	1103.834 166	1806.847 113

1. FZNNHRN consistently exhibits the shortest execution time among all compared approaches, reflecting its superior computational efficiency.

2. With increasing network scale, the performance advantage of FZNNHRN becomes pronounced.

3. Under identical resource and time constraints, FZNNHRN can solve notably larger networks, demonstrating robust effectiveness in handling complex dynamic problems.

5 Theoretical analysis of convergence

Theorem 1 Consider the Laplacian matrix $\boldsymbol{L}_{mn}^{\text{hammock}} \in \mathbb{R}^{mn}$ representing the resistive network, along with a current vector $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{mn}$. When a monotonically increasing odd activation function $\boldsymbol{\mathcal{F}}(\cdot)$ is employed, the state vector $\boldsymbol{V}(t)$ of FZNNHRN is guaranteed to globally converge to the theoretical solution $\boldsymbol{V}^*(t) = \left(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\right)^{-1}\boldsymbol{\mathcal{I}}$ of Eq. (16), regardless of the initial condition $\boldsymbol{V}(0) \in \mathbb{R}^{mn}$.

Proof Consider the time-varying hammock lattice $\boldsymbol{L}_{mn}^{\text{hammock}} \in \mathbb{R}^{mn}$ and the current vector $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{mn}$. Define the error term as

$$\boldsymbol{E}(t) = \boldsymbol{V}(t) - \boldsymbol{V}^*(t) = \boldsymbol{V}(t) - \left(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\right)^{-1}\boldsymbol{\mathcal{I}}, \quad (31)$$

with $\dfrac{\mathrm{d}\boldsymbol{E}(t)}{\mathrm{d}t} = \boldsymbol{L}_{mn}^{\text{hammock}}(t)\dfrac{\mathrm{d}\boldsymbol{V}(t)}{\mathrm{d}t}$.

Coupled with Eq. (16), the following equation can be obtained:

$$\begin{aligned}\dot{\boldsymbol{E}}(t) = &-\gamma\left(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\left(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\right)^{\mathrm{T}} + \lambda\boldsymbol{E}_{mn}\right)^{k} \\ &\cdot\boldsymbol{\mathcal{F}}\left(\boldsymbol{E}(t)\right).\end{aligned} \quad (32)$$

Consider the Lyapunov candidate function $L(\boldsymbol{E}(t),t)$ defined as $L(\boldsymbol{E}(t),t) = \frac{1}{2}\|\boldsymbol{E}(t)\|_{\mathrm{F}}^2$ ($\|\cdot\|_{\mathrm{F}}^2$ represents the squared Frobenius norm), which can also be expressed as $L(\boldsymbol{E}(t),t) = \frac{1}{2}\operatorname{tr}\left(\boldsymbol{E}^{\mathrm{T}}(t)\boldsymbol{E}(t)\right)$. When $\boldsymbol{E}(t) \neq \boldsymbol{0}$, the temporal derivative of $L(\boldsymbol{E}(t),t)$ is formulated as follows:

$$\begin{aligned}\frac{\mathrm{d}L(\boldsymbol{E}(t),t)}{\mathrm{d}t} &= \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\operatorname{tr}\left(\boldsymbol{E}^{\mathrm{T}}(t)\boldsymbol{E}(t)\right) = \operatorname{tr}\left(\boldsymbol{E}^{\mathrm{T}}(t)\dot{\boldsymbol{E}}(t)\right) \\ &= -\gamma\operatorname{tr}\Bigg(\left(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\left(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\right)^{\mathrm{T}} + \lambda\boldsymbol{E}_{mn}\right)^{k} \\ &\quad\cdot\boldsymbol{E}^{\mathrm{T}}(t)\boldsymbol{\mathcal{F}}(\boldsymbol{E}(t))\Bigg).\end{aligned} \quad (33)$$

By applying standard trace inequalities for symmetric positive semidefinite matrices, a symmetric positive semidefinite matrix can be obtained by

$$\frac{\mathrm{d}L(\boldsymbol{E}(t),t)}{\mathrm{d}t} \leq -\gamma\left(\boldsymbol{\alpha}^2(t) + \lambda\right)^{k}\sum_{i=1}^{m}\sum_{j=1}^{n}e_{ij}f(e_{ij}), \quad (34)$$

where $\boldsymbol{\alpha}(t) = \lambda_{\min}(\boldsymbol{L}_{mn}^{\text{hammock}}(t))$, e_{ij} is an element of matrix $\boldsymbol{E}$, and $\boldsymbol{\mathcal{F}}(\boldsymbol{E}(t)) = [f(e_{11}), f(e_{12}), \ldots, f(e_{mn})]^{\mathrm{T}}$.

Because $\boldsymbol{\mathcal{F}}(\cdot)$ is odd and monotonically increasing, $e_{ij}f(e_{ij}) \geq 0$ with equality only when $e_{ij} = 0$. Therefore, $\dfrac{\mathrm{d}L(\boldsymbol{E}(t),t)}{\mathrm{d}t} \leq 0$, which guarantees the global convergence of the error term $\boldsymbol{E}(t)$ to zero for any initial condition by the Lyapunov stability theorem.

For the LAF case, the convergence rate is expressed as $\mathrm{ECR} = \gamma\left(\boldsymbol{\alpha}^2(t) + \lambda\right)^{k}$.

Detailed derivations of Eq. (34) and the LAF case are provided in the supplementary materials.

Theorem 2 Consider the lattice matrix $\boldsymbol{L}_{mn}^{\text{hammock}} \in \mathbb{R}^{mn}$ of a resistive network and a current vector $\boldsymbol{\mathcal{I}} \in \mathbb{R}^{mn}$. When the activation function is defined as $\boldsymbol{\mathcal{F}}(x) = x \cdot \boldsymbol{\psi}(x)$, where $\boldsymbol{\psi}(x) > 0$ for all x, the state vector $\boldsymbol{V}(t)$ of the FZNNHRN model, initialized from an arbitrary state $\boldsymbol{V}(0)$, is guaranteed to converge globally to the theoretical solution $\boldsymbol{V}^*(t) = \left(\boldsymbol{L}_{mn}^{\text{hammock}}(t)\right)^{-1}\boldsymbol{\mathcal{I}}$ of Eq. (16).

The complete derivation is provided in the supplementary materials.

6 Numerical examples

Fig. 5 displays the temporal behavior of the electric potential $\boldsymbol{V}(t)$ within a 3 × 3 resistive network, while considering different activation functions and distinct resistivity variation rates ρ. Each curve corresponds to a distinct component of $\boldsymbol{V}(t)$. The network is configured with $(\gamma, \lambda, k) = (10, 3, 2)$, and the resistor values vary over time as $s = r = 1 + 0.004|\sin t|$.

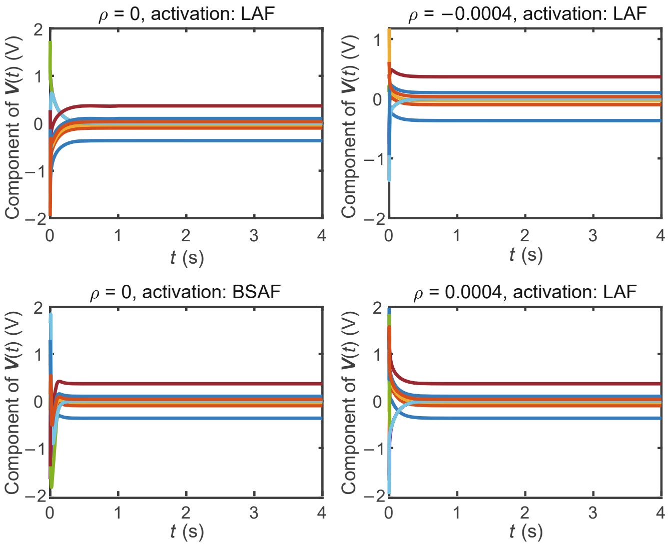

Fig. 5 Time-varying process of potential $\boldsymbol{V}(t)$ under different activation functions and resistivity variation rates ρ (each curve starts from a randomly generated initial value)

Although different activation functions yield similar dynamic profiles, variations in ρ result in prominent trajectory differences, indicating that the system is highly sensitive to resistivity changes.

Fig. 6 shows the time-dependent decay of $\|\boldsymbol{L}_{mn}^{\text{hammock}}\boldsymbol{V}(t)-\boldsymbol{\mathcal{I}}\|_2$ ($\|\cdot\|_2$ denotes the Euclidean norm) for a 3×3 hammock resistor network with $s=1+0.004|\sin t|$ and $r=100(1+0.004|\sin t|)$, starting from $\boldsymbol{V}(0)$ with $(\gamma,\lambda,k)=(10,3,2)$. All activation functions drive the error to zero; however, the smooth power sigmoid function (SPSF), power sigmoid activation function (PSAF), bipolar sigmoid activation function (BSAF), and LAF converge markedly faster.

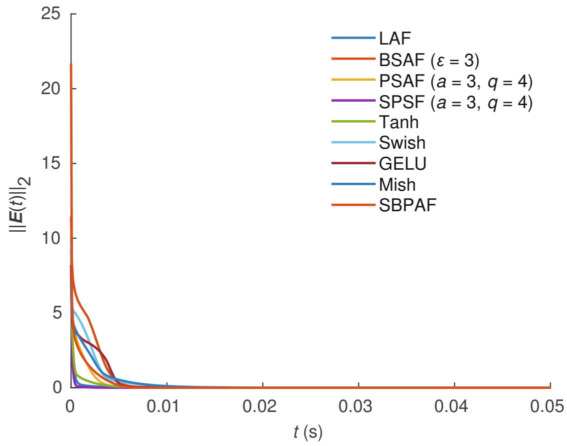

Fig. 6 Evolution of $\|\boldsymbol{L}_{mn}^{\text{hammock}}(t)\boldsymbol{V}(t)-\boldsymbol{\mathcal{I}}\|_2$ over time t under various activation functions. Tanh: hyperbolic tangent; GELU: Gaussian error linear unit; SBPAF: sign-bi-power activation function; Swish: Swish activation function; Mish: Mish activation function. ε is a parameter of BSAF and a and q are parameters of PSAF and SPSF

Figs. 7a–7c show the convergence behavior under the LAF across various values of γ, λ, and k, respectively. The parameter configurations are as follows: $(\gamma,\lambda,k)=(*,3,2)$ in Fig. 7a, $(\gamma,\lambda,k)=(10,*,2)$ in Fig. 7b, and $(\gamma,\lambda,k)=(3,5,*)$ in Fig. 7c. Larger parameter values accelerate convergence (hardware permitting); all cases converge to the same solution.

To examine the influence of initialization, Fig. 8 shows that with parameters fixed at $(\gamma,\lambda,k)=(10,3,2)$, all tested initializations result in a vanishing error, confirming insensitivity to the initial conditions and reliable convergence.

Fig. 9 compares equivalent resistances computed using Eq. (7) with those obtained by FZNNHRN at the selected node pairs, validating the accuracy of the model for resistor network solutions.

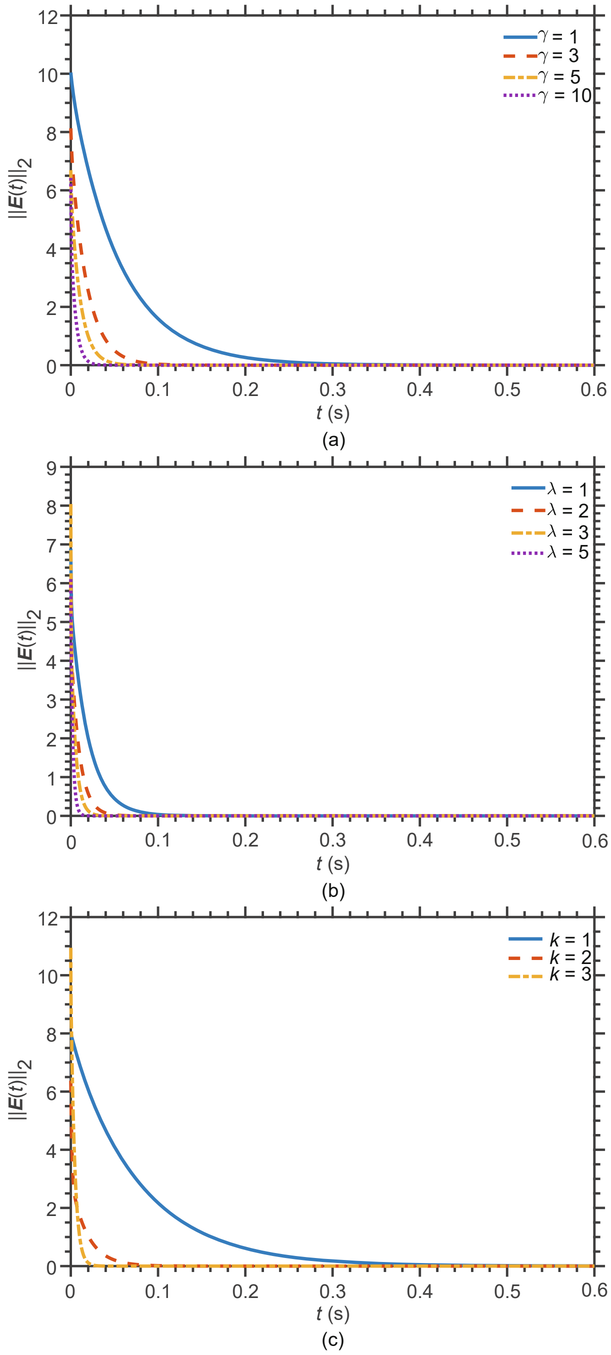

Fig. 7 Time evolution of $\|\boldsymbol{E}(t)\|_2$ across various values of γ (a), λ (b), and k (c)

7 Applications

7.1 Applications in path planning

Path planning is a key research area in the fields of AI and robotics (Koul and Horiuchi, 2019; Patle et al., 2019; Pan et al., 2022; Ji et al., 2023; Liu et al., 2023; Yang et al., 2023; Zhu et al., 2023). In real-world scenarios, path planning is typically constrained by specific environmental conditions (Fu et al., 2006; Aybars, 2008; Kulathunga, 2022; Mazaheri et al., 2024), which introduce additional complexity to the planning process. This subsection presents preliminary path planning simulations using the hammock resistor network structure.

Hammock resistor networks feature a distinct topology and nodal arrangement, where the current flows from the highest to the lowest potential nodes, forming a nonuniform potential gradient. The potential generally decreases from high- to low-potential regions. Obstacles are incorporated by assigning high potential values to their regions, creating repulsive barriers that prevent path penetration. For path planning, the start and target nodes correspond to the highest and lowest potentials, respectively, reflecting the boundary conditions and

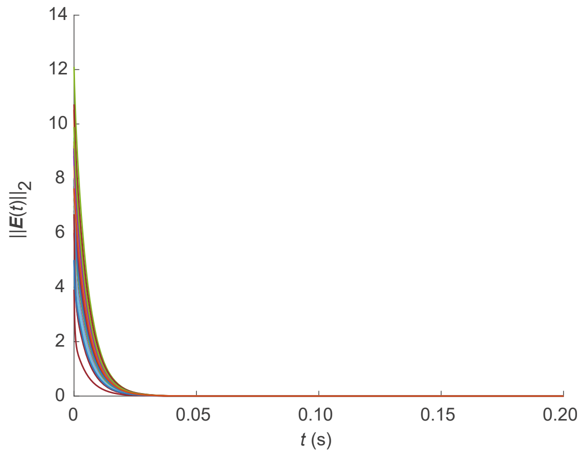

Fig. 8 Evolution of $\|L_{mn}^{\text{hammock}}(t)V(t)-\mathcal{I}\|_2$ over time t, with each curve starting from a randomly generated initial condition

connectivity of the network and highlighting its suitability for such tasks.

To exploit this potential distribution, a directionally guided path search algorithm, termed directional potential field path planning (DPFPP), is proposed based on a precomputed two-dimensional potential field (obsv). The proposed algorithm adopts a step-wise greedy strategy, initially selecting the neighbor closest to the target within the Euclidean distance. At each step, three candidates, forward and $\pm 45^\circ$ relative to the previous direction, are evaluated, and the next node is determined according to the following scoring function:

$$\text{Score}(c)=\text{obsv}(c)+w(1-\cos\theta), \tag{35}$$

where c represents the candidate node, θ denotes the angle between its direction and the target direction, and w penalizes directional deviation.

The absence of a feasible candidate triggers backtracking to explore alternative paths. The process continues until the target is reached or a maximum iteration limit is met, yielding a feasible path guided by the potential field. Algorithm S1 and Fig. S3 in the supplementary materials include the complete algorithm pseudocode and flowchart, respectively.

Fig. 10 shows a 10×10 hammock resistor network, with the resistance parameters set as $s=r=1$. The current flows from the input node $d_1(4,3)$ to the output node $d_2(9,9)$, and several arbitrary obstacles are manually added to the network. Fig. 10a presents the initial network structure. Subsequently, the potential distribution of the network is computed using Eq. (1) and the FZNNHRN method (Fig. 11). Based on this potential field, Fig. 12 illustrates the path planning process under the presence of obstacles, where the robot begins from the initial point and moves along the direction of the steepest potential descent until reaching the goal. Fig. 10b visualizes the final path based on the result illustrated in Fig. 12.

7.2 Efficiency assessment of path planning algorithms

In the domain of path planning, algorithms typically generate paths with distinct characteristics due to differences in their optimization objectives and computational strategies. Therefore, this subsection compares the computational efficiency of the proposed path planning algorithm with several classic path planning methods, including Dijkstra, A-star (A*), Floyd, and jump point search (JPS).

Initially, we perform discretization processing on the input image by dividing it into grid sizes that are compatible with the resistor network structure, thereby achieving spatial mapping from the image to circuit nodes. Based on this discretized representation, we implement and demonstrate the paths generated by the proposed and comparative algorithms under identical environmental conditions (Fig. 13). Although each algorithm successfully produces effective paths by leveraging its unique strengths, they exhibit substantial differences in computational efficiency.

To ensure a fair assessment of the time performance between the proposed and comparative algorithms, we establish the following uniform simulation conditions:

1. Simulation platform. All simulations are conducted using MATLAB R2021a. The hardware platform comprises an Intel® Core™ i5-10300H CPU @ 2.50 GHz and an NVIDIA GeForce GTX 1650 Ti GPU.

2. Grid scale. The number of simulation nodes ranges from 100 to 10 000. The grid sizes include 10×10, 15×15, 20×20, 30×30, 40×40, 50×50, 60×60, 70×70, 80×80, 90×90, and 100×100.

3. Obstacle density. A fixed obstacle density of 25% relative to the total number of nodes in the resistor network is used to ensure comparable computational complexity across different grid scales.

4. Start and end positions. The start and target nodes are consistently maintained across all grid scales and algorithms to ensure fair comparisons.

5. Evaluation protocol. Each algorithm is executed 30 times for each grid scale. The average path planning time is recorded as the performance indicator to mitigate the influence of randomness.

The specific simulation results are presented in Fig. 14. At a small network scale, all five algorithms complete path planning within a short time, and the performance differences are negligible. Notably, the JPS algorithm is the most efficient in this regime. As the network scale increases, the proposed DPFPP algorithm progressively exhibits its advantage and ultimately becomes the fastest method. Although the A* and JPS algorithms exhibit comparable performance on small-to-medium networks and remain relatively efficient as the scale increases, they are still slightly slower than the DPFPP algorithm. In contrast, the performance of the Floyd and Dijkstra algorithms degrades markedly with increasing scale, making them unsuitable for large-scale scenarios. In particular, Floyd requires 118 s at 50×50 and is excluded from larger network evaluations. This is because it computes the shortest paths between all node pairs, which incurs a substantially higher computational cost. Overall, these results indicate that DPFPP is well-suited to large-scale emergency path planning. Dynamic path planning simulations with real-time moving obstacles are presented in Section S2 of the supplementary materials, including a final snapshot (Fig. S2) and a complete animated graphics interchange format (GIF), further validating the robustness of the proposed framework.

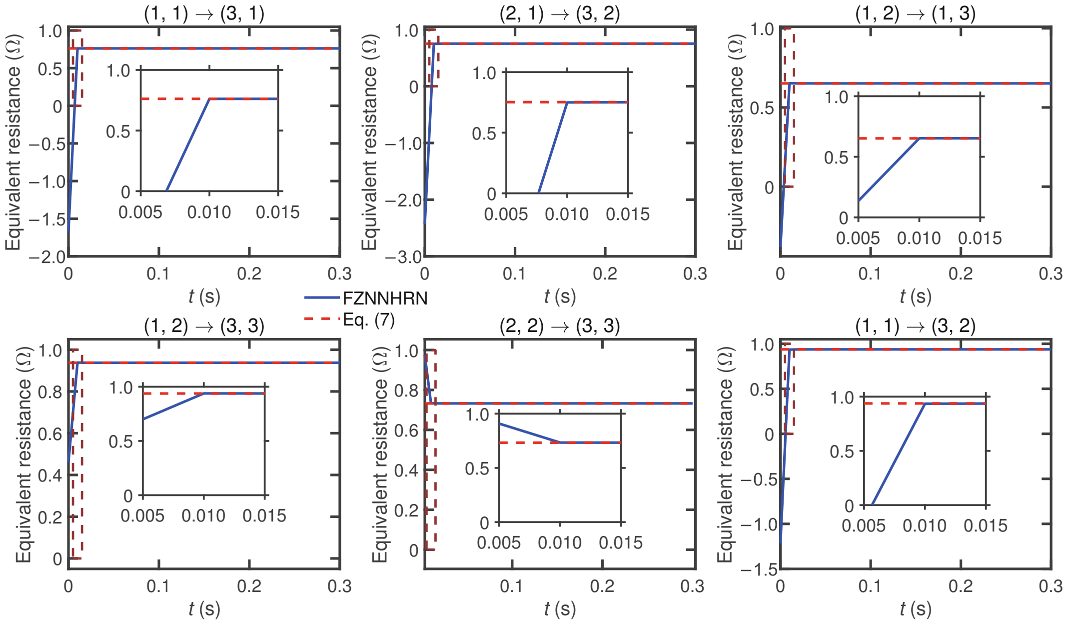

Fig. 9 Comparison of equivalent resistances computed using Eq. (7) and FZNNHRN for a 3 × 3 network with $(\gamma, \lambda, k) = (10, 3, 2)$ and an LAF

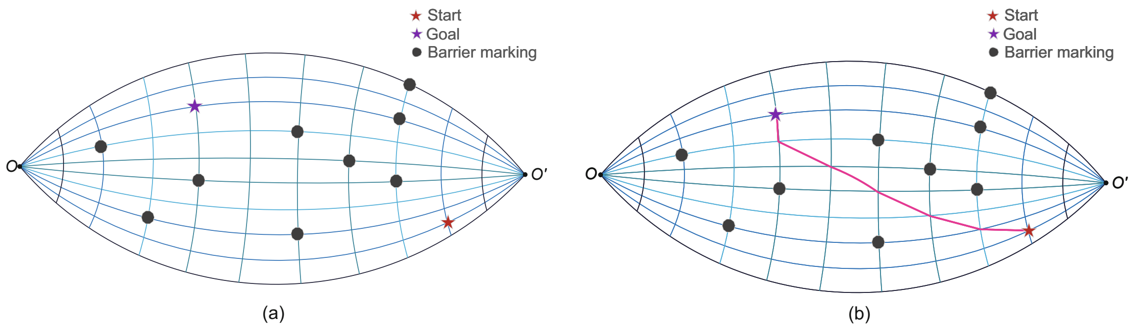

Fig. 10 Hammock network maps before and after path planning: (a) the initial hammock network map with obstacles; (b) the final hammock network map with the marked path

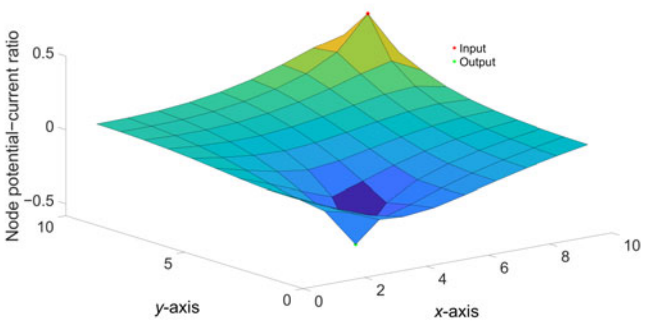

Fig. 11 Potential distribution calculated by the FZNNHRN model

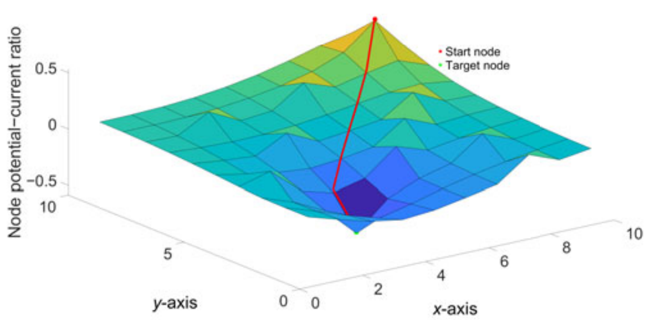

Fig. 12 Path planning result on the hammock network map

8 Discussion and conjectures

8.1 Avoiding local potential minima

Local minima are a primary challenge in potential field-based planning. Hammock resistor networks inherently produce a smooth potential distribution, which notably reduces the likelihood of spurious traps. To further ensure robustness, the proposed algorithm integrates directional constraints and a backtracking mechanism. These components allow robots to escape suboptimal regions and maintain consistent progress toward targets, even in the presence of complex obstacles or time-varying disturbances.

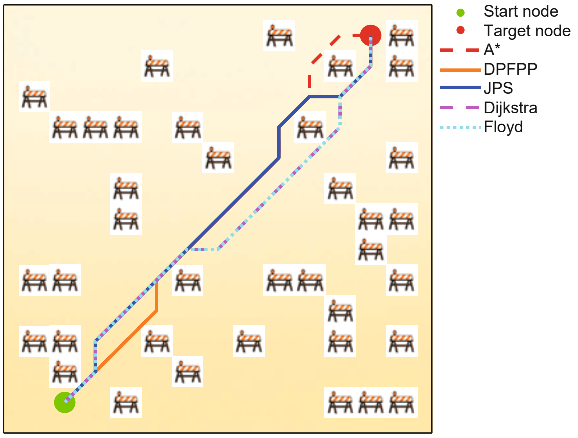

Fig. 13 Path planning diagrams for five algorithms

8.2 Optimal path planning problems

In Riemannian geometry, geodesics represent locally shortest paths, which are analogous to free-fall trajectories in general relativity. Motivated by this, we present the following conjectures:

1. In a hammock-shaped surface resistor network, the potential field induced by its intrinsic electrical properties is analogous to a gravitational field. The current trajectory from the source to the sink, governed by natural potential descent, is equivalent to the shortest path on the hammock surface connecting these two points.

If conjecture 1 does not hold, an alternative physical analogy is instructive. Consider two balls released from the same point: one follows a gentle slope, whereas the other follows a steeper curved track. The latter one may arrive earlier, tracing the classical brachistochrone, which minimizes travel time rather than distance. Analogously, in a hammock-shaped surface resistor network, the current may follow a natural potential descent trajectory that corresponds to a brachistochrone on the curved surface. This highlights an important point: in nonuniform environments, the optimal path is not necessarily the shortest in terms of distance but the path that minimizes an overall cost, such as time or energy.

2. Based on the natural potential descent principle, the path from the source to the target on a hammock-shaped surface is equivalent to the fastest route, thereby enabling efficient navigation in complex spatial environments.

In nonuniform scenarios, such as ground robots avoiding rough terrain or unmanned aerial vehicles bypassing adverse winds, minimizing the total cost (time/energy) is typically more critical than minimizing the distance. The proposed strategy remains optimal if either of the conjectures holds.

9 Conclusions and prospects

This study proposes a fast ZNN algorithm to solve time-varying Laplacian linear systems in hammock surface metastructured quadrilateral resistor networks. By leveraging the inherent structure of the Laplacian matrix, the proposed approach substantially reduces computational complexity and enables efficient real-time calculation of electric potential distributions in dynamic scenarios. Theoretical analysis demonstrates that the proposed algorithm achieves global exponential convergence, and numerical simulations verify its high computational efficiency and robustness with respect to the initial conditions across networks of various scales. Furthermore, the electric potential distribution computed using the proposed algorithm is incorporated into robot path planning, yielding a potential field-based path planning method. Simulation results indicate that the proposed algorithm maintains a path quality comparable to that of classical algorithms while incurring lower computational costs. Nevertheless, due to hardware limitations, the proposed algorithm is incapable of handling ultra-large-scale resistor networks. Moreover, as this algorithm is specifically designed for hammock resistor network models, it is difficult to extend to other types of resistor networks. Future research will focus on generalizing the proposed approach and exploring fast solvers that can be applied to a broader range of resistor network configurations. This study aims to foster interdisciplinary integration and convergence among physicists, mathematicians, computer scientists, and network scientists while outlining key challenges and directions for future research.

Supplementary information

The online version contains supplementary material available at https://doi.org/10.1631/ENG.ITEE.2026.0055.

Acknowledgments

This work was supported by the Natural Science Foundation of Shandong Province (No. ZR2022MA092) and the Shandong Provincial Higher Education Youth Innovation Team Development Plan (No. 2023KJ214).

Author contributions

Yanpeng ZHENG and Xiaoyu JIANG conceived the project and performed and analyzed the calculations. Guijie ZHANG and Sung-Kwun OH validated the results and generated the figures. Zhaolin JIANG developed the fast algorithm. All the authors revised and finalized the paper.

Conflict of interest

All the authors declare that they have no conflict of interest.

Data availability

The data that support the findings of this study are available from the corresponding authors upon reasonable request.

Declaration on the use of generative AI tools

During the preparation of this work, the authors used ChatGPT to improve language. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

Aybars U, 2008. Path planning on a cuboid using genetic algorithms. *Inform Sci*, 178(16):3275-3287. https://doi.org/10.1016/j.ins.2008.04.005

Bapst V, Keck T, Grabska-Barwińska A, et al., 2020. Unveiling the predictive power of static structure in glassy systems. *Nat Phys*, 16(4):448-454. https://doi.org/10.1038/s41567-020-0842-8

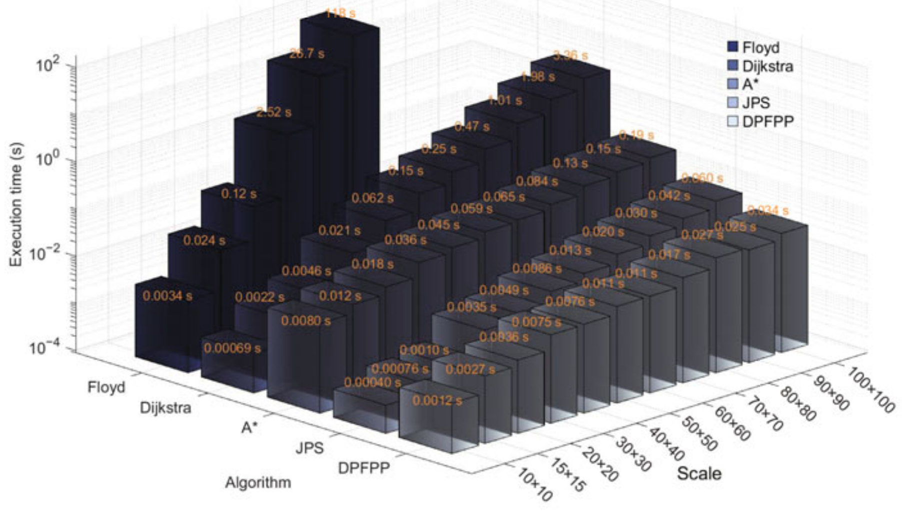

Fig. 14 Path planning efficiency of each algorithm at different grid scales

Bianconi G, 2021. Higher-Order Networks. Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/9781108770996

Bommes D, Lévy B, Pietroni N, et al., 2013. Quad-mesh generation and processing: a survey. *Comput Graph Forum*, 32(6):51-76. https://doi.org/10.1111/cgf.12014

Campen M, 2017. Partitioning surfaces into quadrilateral patches: a survey. *Comput Graph Forum*, 36(8):567-588. https://doi.org/10.1111/cgf.13153

Chair N, 2014. The effective resistance of the N-cycle graph with four nearest neighbors. *J Stat Phys*, 154(4):1177-1190. https://doi.org/10.1007/s10955-014-0916-z

Chen JQ, Ji WY, Tan ZZ, 2024. Electrical properties of a $2 \times n$ non-regular hammock network. *Indian J Phys*, 98(8):2851-2860. https://doi.org/10.1007/s12648-023-03027-w

Darwish M, Boysan H, Liewald C, et al., 2018. A resistor network simulation model for laser-scanning photo-current microscopy to quantify low conductance regions in organic thin films. *Org Electron*, 62:474-480. https://doi.org/10.1016/j.orgel.2018.08.002

Essam JW, Izmailyan NS, Kenna R, et al., 2015. Comparison of methods to determine point-to-point resistance in nearly rectangular networks with application to a 'hammock' network. *R Soc Open Sci*, 2(4):140420. https://doi.org/10.1098/rsos.140420

Fu Z, Zhao YZ, Qian ZY, et al., 2006. Wall-climbing robot path planning for testing cylindrical oilcan weld based on Voronoi diagram. Proc IEEE/RSJ Int Conf on Intelligent Robots and Systems, p.2749-2753. https://doi.org/10.1109/IROS.2006.282001

Gao Z, Liu Q, Hesthaven JS, et al., 2021. Non-intrusive reduced order modeling of convection dominated flows using artificial neural networks with application to Rayleigh-Taylor instability. *Commun Comput Phys*, 30(1):97-123. https://doi.org/10.4208/cicp.OA-2020-0064

Garcia SR, Yih S, 2018. Supercharacters and the discrete Fourier, cosine, and sine transforms. *Commun Algebr*, 46(9):3745-3765. https://doi.org/10.1080/00927872.2018.1424866

Greiner W, 1998. Classical Electrodynamics. Springer, New York, USA. https://doi.org/10.1007/978-1-4612-0587-6

Hadad Y, Soric JC, Khanikaev AB, et al., 2018. Self-induced topological protection in nonlinear circuit arrays. *Nat Electron*, 1(3):178-182. https://doi.org/10.1038/s41928-018-0042-z

Han Y, Cheng GF, Qiu BB, 2025. A triple-integral noise-resistant RNN for time-dependent constrained nonlinear optimization applied to manipulator control. *IEEE Trans Ind Electron*, 72(6):6124-6133. https://doi.org/10.1109/TIE.2024.3482112

Hum SV, Du BZ, 2017. Equivalent circuit modeling for reflectarrays using Floquet modal expansion. *IEEE Trans Antenn Propag*, 65(3):1131-1140. https://doi.org/10.1109/TAP.2017.2657483

Izmailian NS, Kenna R, Wu FY, 2013. The two-point resistance of a resistor network: a new formulation and application to the cobweb network. *J Phys A Math Theor*, 47(3):035003. https://doi.org/10.1088/1751-8113/47/3/035003

Jeon SJ, Lee JS, Kim YJ, et al., 2021. Electrical analysis method for gamma-ray imaging system based on resistive network readout. *IEEE Trans Nucl Sci*, 68(9):2392-2399. https://doi.org/10.1109/TNS.2021.3104004

Ji YX, Ni LT, Zhao C, et al., 2023. TriPField: a 3D potential field model and its applications to local path planning of autonomous vehicles. *IEEE Trans Intell Transp Syst*, 24(3):3541-3554. https://doi.org/10.1109/TITS.2022.3231259

Jiang XY, Zhang GJ, Zheng YP, et al., 2024. Explicit potential function and fast algorithm for computing potentials in $\alpha \times \beta$ conic surface resistor network. *Expert Syst Appl*, 238:122157. https://doi.org/10.1016/j.eswa.2023.122157

Jiang XY, Meng DX, Zheng YP, et al., 2025a. A dynamic cobweb resistance network solution based on a structured zeroing neural network and its applications. *Sci Rep*, 15(1):5222. https://doi.org/10.1038/s41598-025-89471-6

Jiang XY, Dai JW, Zheng YP, et al., 2025b. Modeling of omnidirectional antenna space radiation shaped resistor networks and efficient path planning. *Appl Phys Rev*, 12(4):041405. https://doi.org/10.1063/5.0291210

Jiang XY, Guan Y, Zheng YP, et al., 2026. Robot path planning based on potential function of a resistor network. *Appl Math Model*, 151:116534. https://doi.org/10.1016/j.apm.2025.116534

Jiang ZL, Zhou YF, Jiang XY, et al., 2023. Analytical potential formulae and fast algorithm for a horn torus resistor network. *Phys Rev E*, 107(4):044123. https://doi.org/10.1103/PhysRevE.107.044123

Jin L, Ying LF, Lu HY, et al., 2021. Saturation-allowed neural dynamics applied to perturbed time-dependent system of linear equations and robots. *IEEE Trans Ind Electron*, 68(10):9844-9854. https://doi.org/10.1109/TIE.2020.3029478

Jin L, Li YT, Zhang XY, et al., 2024. Fuzzy k-winner-take-all network for competitive coordination in multirobot systems. *IEEE Trans Fuzzy Syst*, 32(4):2005-2016. https://doi.org/10.1109/TFUZZ.2023.3339654

Koul S, Horiuchi TK, 2019. Waypoint path planning with synaptic-dependent spike latency. *IEEE Trans Circ Syst I Reg Pap*, 66(4):1544-1557. https://doi.org/10.1109/TCSI.2018.2882818

Kulathunga G, 2022. A reinforcement learning based path planning approach in 3D environment. *Proc Comput Sci*, 212:152-160. https://doi.org/10.1016/j.procs.2022.10.217

Lian YF, Xiao XT, Zhang JL, et al., 2024. Neural dynamics for cooperative motion control of omnidirectional mobile manipulators in the presence of noises: a distributed approach. *IEEE/CAA J Autom Sin*, 11(7):1605-1620. https://doi.org/10.1109/JAS.2024.124425
Liu LX, Wang X, Yang X, et al., 2023. Path planning techniques for mobile robots: review and prospect. *Expert Syst Appl*, 227:120254. https://doi.org/10.1016/j.eswa.2023.120254
Lüpke F, Eschbach M, Heider T, et al., 2017. Electrical resistance of individual defects at a topological insulator surface. *Nat Commun*, 8(1):15704. https://doi.org/10.1038/ncomms15704
Mazaheri H, Goli S, Nourollah A, 2024. Path planning in three-dimensional space based on butterfly optimization algorithm. *Sci Rep*, 14(1):2332. https://doi.org/10.1038/s41598-024-52750-9
Millán AP, Sun HL, Giambagli L, et al., 2025. Topology shapes dynamics of higher-order networks. *Nat Phys*, 21(3):353-361. https://doi.org/10.1038/s41567-024-02757-w
Novoselov KS, Geim AK, Morozov SV, et al., 2004. Electric field effect in atomically thin carbon films. *Science*, 306(5696):666-669. https://doi.org/10.1126/science.1102896
Novoselov KS, Geim AK, Morozov SV, et al., 2005. Two-dimensional gas of massless Dirac fermions in graphene. *Nature*, 438(7065):197-200. https://doi.org/10.1038/nature04233
Pan ZH, Zhang CX, Xia YQ, et al., 2022. An improved artificial potential field method for path planning and formation control of the multi-UAV systems. *IEEE Trans Circ Syst II Exp Briefs*, 69(3):1129-1133. https://doi.org/10.1109/TCSII.2021.3112787
Patle BK, Babu LG, Pandey A, et al., 2019. A review: on path planning strategies for navigation of mobile robot. *Def Technol*, 15(4):582-606. https://doi.org/10.1016/j.dt.2019.04.011
Pei SC, Yeh MH, 2001. The discrete fractional cosine and sine transforms. *IEEE Trans Signal Process*, 49(6):1198-1207. https://doi.org/10.1109/78.923302
Qiu BB, Guo JJ, Li XD, et al., 2022. Discrete-time advanced zeroing neurodynamic algorithm applied to future equality-constrained nonlinear optimization with various noises. *IEEE Trans Cybern*, 52(5):3539-3552. https://doi.org/10.1109/TCYB.2020.3009110
Qiu BB, Guo JJ, Mao MZ, et al., 2024. A fuzzy-enhanced robust DZNN model for future multiconstrained nonlinear optimization with robotic manipulator control. *IEEE Trans Fuzzy Syst*, 32(1):160-173. https://doi.org/10.1109/TFUZZ.2023.3293834
Rhazaoui K, Cai Q, Adjiman CS, et al., 2013. Towards the 3D modeling of the effective conductivity of solid oxide fuel cell electrodes: I. Model development. *Chem Eng Sci*, 99:161-170. https://doi.org/10.1016/j.ces.2013.05.030
Shi Y, Zhang YN, 2020. New discrete-time models of zeroing neural network solving systems of time-variant linear and nonlinear inequalities. *IEEE Trans Syst Man Cybern Syst*, 50(2):565-576. https://doi.org/10.1109/TSMC.2017.2751259
Shi Y, Jin L, Li S, et al., 2022. Novel discrete-time recurrent neural networks handling discrete-form time-variant multi-augmented Sylvester matrix problems and manipulator application. *IEEE Trans Neur Netw Learn Syst*, 33(2):587-599. https://doi.org/10.1109/TNNLS.2020.3028136
Sun ZB, Tang SJ, Jin L, et al., 2023a. Nonconvex activation noise-suppressing neural network for time-varying quadratic programming: application to omnidirectional mobile manipulator. *IEEE Trans Ind Inform*, 19(11):10786-10798. https://doi.org/10.1109/TII.2023.3241683
Sun ZB, Tang SJ, Zhang JL, et al., 2023b. Nonconvex noise-tolerant neural model for repetitive motion of omnidirectional mobile manipulators. *IEEE/CAA J Autom Sin*, 10(8):1766-1768. https://doi.org/10.1109/JAS.2023.123273
Tan Z, Tan ZZ, Zhou L, et al., 2018. Electrical properties of an $m \times n$ hammock network. *Commun Theor Phys*, 69(5):610-616. https://doi.org/10.1088/0253-6102/69/5/610
Tan ZZ, 2015. Recursion-transform method for computing resistance of the complex resistor network with three arbitrary boundaries. *Phys Rev E*, 91(5):052122. https://doi.org/10.1103/PhysRevE.91.052122
Tan ZZ, Tan Z, 2020. The basic principle of $m \times n$ resistor networks. *Commun Theor Phys*, 72(5):055001. https://doi.org/10.1088/1572-9494/ab7702
Wang XZ, Che ML, Wei YM, 2019. Neural networks based approach solving multi-linear systems with $\mathcal{M}$-tensors. *Neurocomputing*, 351:33-42. https://doi.org/10.1016/j.neucom.2019.03.025
Wei P, Wang XZ, Wei YM, 2023. Neural network models for time-varying tensor complementarity problems. *Neurocomputing*, 523:18-32. https://doi.org/10.1016/j.neucom.2022.12.008
Wen X, Don WS, Gao Z, et al., 2020. An edge detector based on artificial neural network with application to hybrid compact-WENO finite difference scheme. *J Sci Comput*, 83(3):49. https://doi.org/10.1007/s10915-020-01237-6
Willke P, Kotzott T, Pruschke T, et al., 2017. Magnetotransport on the nano scale. *Nat Commun*, 8(1):15283. https://doi.org/10.1038/ncomms15283
Wu FY, 2004. Theory of resistor networks: the two-point resistance. *J Phys A Math Gen*, 37(26):6653-6673. https://doi.org/10.1088/0305-4470/37/26/004
Wu WQ, Zheng B, 2020. Improved recurrent neural networks for solving Moore-Penrose inverse of real-time full-rank matrix. *Neurocomputing*, 418:221-231. https://doi.org/10.1016/j.neucom.2020.08.026
Xu GY, Eleftheriades GV, Hum SV, 2021. Analysis and design of general printed circuit board metagratings with an equivalent circuit model approach. *IEEE Trans Antenn Propag*, 69(8):4657-4669. https://doi.org/10.1109/TAP.2021.3060084
Yang LW, Li P, Qian S, et al., 2023. Path planning technique for mobile robots: a review. *Machines*, 11(10):980. https://doi.org/10.3390/machines11100980
Zhang D, Yang B, Tan JP, et al., 2021. Impact damage localization and mode identification of CFRPs panels using an electric resistance change method. *Compos Struct*, 276:114587. https://doi.org/10.1016/j.compstruct.2021.114587
Zhang YN, Jiang DC, Wang J, 2002. A recurrent neural network for solving Sylvester equation with time-varying coefficients. *IEEE Trans Neur Netw*, 13(5):1053-1063. https://doi.org/10.1109/TNN.2002.1031938
Zhang ZJ, Sun XL, Liu YQ, et al., 2025. A meta-interactive neural network for solving time-varying quadratic programming problems. *Nat Commun*, 16(1):10284. https://doi.org/10.1038/s41467-025-65192-2
Zhao H, 2025. A scientist question: research on the impact of super structured quadrilateral meshes on convergence and accuracy of finite element analysis. https://arxiv.org/abs/2507.17184
Zhao WJ, Zheng YP, Jiang XY, et al., 2023. Two optimized novel potential formulas and numerical algorithms for $m \times n$ cobweb and fan resistor networks. *Sci Rep*, 13(1):12417. https://doi.org/10.1038/s41598-023-39478-8
Zhu ZX, Yin Y, Lyu H, 2023. Automatic collision avoidance algorithm based on route-plan-guided artificial potential field method. *Ocean Eng*, 271:113737. https://doi.org/10.1016/j.oceaneng.2023.113737