

Optimized PID neural network closed-loop control for basal ganglia network in Parkinson's disease

Hengxi Zhang(张恒熙), Honghui Zhang(张红慧), Shuang Liu(柳爽), and Lin Du(都琳)
Citation: Chin. Phys. B, 2025, 34 (**12**): 120701. DOI: 10.1088/1674-1056/ae0d55
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Inverse design of directional hybrid nanoantennas using neural networks

Ru-Lin Guan(管如林), Deng-Chao Huang(黄登朝), Ya-Qiong Li(李雅琼), Chen Wang(王晨), and Bin-Zi Xu(徐彬梓)
Chin. Phys. B, 2025, 34 (**9**): 090702. DOI: 10.1088/1674-1056/adeb5c

Graph neural networks unveil universal dynamics in directed percolation

Ji-Hui Han(韩继辉), Cheng-Yi Zhang(张程义), Gao-Gao Dong(董高高), Yue-Feng Shi(石月凤), Long-Feng Zhao(赵龙峰), and Yi-Jiang Zou(邹以江)
Chin. Phys. B, 2025, 34 (**8**): 080702. DOI: 10.1088/1674-1056/add505

Analysis and design of multivalued many-to-one associative memory driven by external inputs

Qiang Fang(方强) and Hao Zhang(张浩)
Chin. Phys. B, 2025, 34 (**8**): 080701. DOI: 10.1088/1674-1056/add24b

Deep learning-enabled inverse design of polarization-selective structural color based on coding metasurface

Haolin Yang(杨昊霖), Bo Ni(倪波), Junhong Guo(郭俊宏), Hua Zhou(周华), and Jianhua Chang(常建华)
Chin. Phys. B, 2025, 34 (**5**): 050702. DOI: 10.1088/1674-1056/adb94a

Text-guided diverse-expression diffusion model for molecule generation

Wenchao Weng(翁文超), Hanyu Jiang(蒋涵羽), Xiangjie Kong(孔祥杰), and Giovanni Pau
Chin. Phys. B, 2025, 34 (**5**): 050701. DOI: 10.1088/1674-1056/adbedd

---

SPECIAL TOPIC — Biophysical circuits: Modeling & applications in neuroscience

Optimized PID neural network closed-loop control for basal ganglia network in Parkinson's disease

Hengxi Zhang(张恒熙)[1,2], Honghui Zhang(张红慧)[1,2,†], Shuang Liu(柳爽)[3], and Lin Du(都琳)[1,2]

[1] *School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an* 710129, *China*
[2] *MIIT Key Laboratory of Dynamics and Control of Complex Systems, Xi'an* 710129, *China*
[3] *Shanghai Engineering Research Center of Physical Vapor Deposition (PVD) Superhard Coating and Equipment, Shanghai Institute of Technology, Shanghai* 201418, *China*

(Received 11 July 2025; revised manuscript received 29 September 2025; accepted manuscript online 30 September 2025)

Conventional open-loop deep brain stimulation (DBS) systems with fixed parameters fail to accommodate inter-individual pathological differences in Parkinson's disease (PD) management while potentially inducing adverse effects and causing excessive energy consumption. In this paper, we present an adaptive closed-loop framework integrating a Yogi-optimized proportional–integral–derivative neural network (Yogi-PIDNN) controller. The Yogi-augmented gradient adaptation mechanism accelerates the convergence of general PIDNN controllers in high-dimensional nonlinear control systems while reducing control energy usage. In addition, a system identification method establishes input–output dynamics for pre-training stimulation waveforms, bypassing real-time parameter-tuning constraints and thereby enhancing closed-loop adaptability. Finally, a theoretical analysis based on Lyapunov stability criteria establishes a sufficient condition for closed-loop stability within the identified model. Computational validations demonstrate that our approach restores thalamic relay reliability while reducing energy consumption by $(81.0 \pm 0.7)\%$ across multi-frequency tests. This study advances adaptive neuromodulation by synergizing data-driven pre-training with stability-guaranteed real-time control, offering a novel framework for energy-efficient and personalized Parkinson's therapy.

Keywords: Parkinson's disease, closed-loop deep brain stimulation, PID neural network, adaptive control

PACS: 07.05.Mh, 87.19.lj, 87.19.lr, 87.85.dq

DOI: 10.1088/1674-1056/ae0d55 **CSTR:** 32038.14.CPB.ae0d55

1. Introduction

Deep brain stimulation (DBS), a therapeutic approach that delivers high-frequency ($>$ 100 Hz) electrical pulses to targeted regions within the basal ganglia (BG), has been widely recognized as an effective treatment for alleviating motor symptoms of Parkinson's disease (PD).[1–3] Conventional open-loop DBS systems rely on static stimulation parameters manually adjusted by clinicians through qualitative assessments. This process necessitates prolonged parameter-adjustment periods, during which fixed high-frequency stimulation can accelerate battery depletion and potentially result in adverse patient outcomes.[4–6] To address these limitations, the adoption of closed-loop control strategies, which adapt stimulation parameters in real time based on biological signals, has emerged as a promising approach to enhance therapeutic efficacy and reduce energy consumption.[7,8]

Among closed-loop control methods, the proportional–integral–derivative (PID) control algorithm has been widely adopted as the primary control strategy for designing closed-loop DBS systems, owing to its structural simplicity, ease of implementation, and broad applicability across various control systems. Experimental and computational studies have consistently demonstrated its effectiveness.[9–14] Nevertheless, the application of PID control algorithms in DBS systems remains challenged by the need to further optimize parameter-tuning strategies to accommodate the dynamic requirements of individual patients.

The PID neural network (PIDNN) is a controller that integrates the core principles of PID algorithms with the adaptive capabilities of neural networks.[15] By encoding proportional, integral, and derivative operations into trainable layers, PIDNN retains the error-driven control logic of PID while overcoming its limitations in parameter tuning. Recent studies have demonstrated its potential in controlling complex nonlinear systems: Zeng *et al.*[16] developed a multivariate PIDNN framework based on adaptive population extremal optimization, achieving robust transient and steady-state performance in such systems. Similarly, Ding *et al.*[17] proposed a fractional-order PIDNN enhanced by beetle swarm optimization, demonstrating superior control in two nonlinear systems. Although neural network controllers have been widely applied to nonlinear control problems,[18–21] their use in DBS remains limited.[22] Liu *et al.*[23] employed a backpropagation neural network (BPNN) to modulate DBS for different

†Corresponding author. E-mail: haozhucy@nwpu.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

PD states. However, this approach requires pre-training using data generated from a well-tuned PID controller, which limits its generalizability. In addition, Wang *et al.*[24] designed an adaptive radial basis function neural network (ARBFNN) controller based on a subthalamic nucleus (STN)–globus pallidus externa (GPe) model, effectively suppressing pathological oscillations in PD. Nevertheless, this model-dependent approach faces significant challenges in high-dimensional and nonlinear brain networks, where designing appropriate control laws becomes computationally intractable.

Motivated by these challenges, we propose a closed-loop DBS strategy based on a Yogi-optimized PIDNN. While previous research primarily focuses on initial weight optimization,[16,17] our methodology introduces the Yogi optimization algorithm[25] to dynamically adjust PIDNN learning rates through gradient-moment estimation, thereby enhancing the controller's ability to handle complex nonlinear systems. The Yogi optimizer is similar to Adam,[26] using an exponentially weighted moving average to estimate the gradient's second moment while employing an independent adaptive learning rate. However, it reformulates the estimation of the second moment to achieve improved performance while maintaining comparable theoretical convergence guarantees.[25,27] It has also been applied in conjunction with other methods in previous research.[28]

As will be discussed in Section 2, the introduction of the Yogi algorithm significantly enhances PIDNN control performance while reducing energy consumption. However, initial simulations revealed higher energy costs than expected compared to conventional open-loop DBS systems, necessitating further optimization. To mitigate this issue, inspired by the AI Pontryagin framework proposed by Böttcher *et al.*,[29] we introduce an enhanced PIDNN training strategy. We first construct an input–output model of the BG network using system identification based on recursive least squares. Building on this model, we replace conventional time-discretized weight updates in PIDNN with a multi-iteration gradient-descent pre-training approach to generate control signals. Our results demonstrate that this architecture achieves energy optimization of control signals without explicitly incorporating an energy-regularization term in the loss function. Furthermore, by leveraging the discrete Lyapunov stability theorem, we derive sufficient conditions (Theorem 1) to guarantee convergence of the estimated state errors.

The rest of this paper is organized as follows. Section 2 presents an overview of the BG model and develops the PIDNN-based control framework, including system identification and stability analysis. Section 3 provides numerical simulation results verifying the effectiveness of the proposed closed-loop control strategy. Finally, the discussion and conclusions are given in Section 4.

2. Model and methods

2.1. Basal ganglia network model

The basal ganglia (BG) network plays a critical role in motor control, and its dysfunction in PD leads to pathological oscillations, such as beta-band activity, which impair thalamic relay reliability. The BG model utilized here is a modification of the classic Rubin–Terman model, developed by So *et al.*[30] Compared with the Rubin–Terman model,[31,32] So *et al.*[30] enhanced the representation of calcium currents and post-hyperpolarization currents in the STN nucleus, as well as the reversal potentials of hyperpolarization currents and leak currents in the globus pallidus (GP) nucleus, to better match experimental data on neuronal firing characteristics.

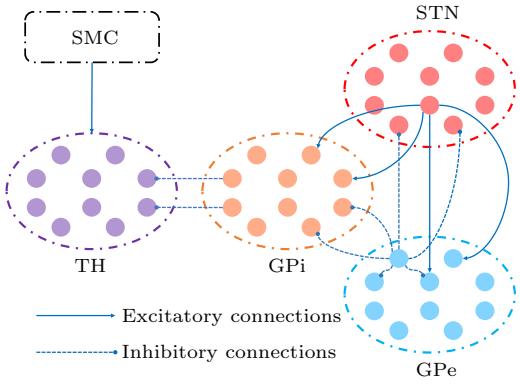


Fig. 1. Structure of the BG network model. Each nucleus consists of 10 neuronal cells. Solid lines indicate excitatory connections, and dashed lines indicate inhibitory connections.

The BG model consists of four nuclei: thalamus (TH), STN, GPe, and globus pallidus interna (GPi). These nuclei are interconnected through both excitatory and inhibitory synaptic connections, forming a network that responds to inputs from the sensorimotor cortex (SMC). As shown in Fig. 1, the connectivity of neurons in the model adopts a sparse connection pattern similar to the Rubin–Terman model, wherein each STN cell projects excitatory signals to two GPe and two GPi neurons; each GPe cell sends inhibitory signals to two STN, two GPi, and two GPe neurons; and each TH neuron receives inhibitory input from a single GPi neuron and excitatory input from the SMC. A constant current is applied to GPe neurons to simulate striatal input. The number of neurons in the model is sufficient to reproduce the dynamic behaviors of both normal and Parkinsonian states and to investigate the effects of DBS on the Parkinsonian state, without the need for larger populations.[30,33] The model incorporates 10 Hodgkin–Huxley-type neurons per nucleus, which adequately reproduces PD-related dynamics while avoiding unnecessary computational overhead.[30,32]

We summarize the meanings of the key variables in the equations as follows. v represents the membrane potential of a cell. The membrane capacitance is $C_m = 1$ pF/μm^2. I_{L}, I_{Na}, I_{K}, I_{T}, I_{Ca}, and I_{ahp} denote the leak, sodium, potassium, low-threshold T-type calcium, high-threshold calcium, and afterhyperpolarization potassium currents, respectively.

$I_{\text{app}_x}(x \in \{\text{STN}, \text{GPe}, \text{GPi}\})$ is a positive constant bias current that represents the net synaptic input to these cells from other brain regions. h, n, r, and c are gating variables governed by first-order kinetic equations, and [Ca] represents intracellular Ca^{2+} concentration. The time constants τ_i, $i \in \{h, n, r, c\}$, correspond to the dynamics of these variables.

The synaptic current from a presynaptic cell α to a postsynaptic cell β, with $\alpha, \beta \in \{\text{STN}, \text{GPe}, \text{GPi}, \text{TH}\}$, is defined as

$$I_{\alpha\to\beta} = g_{\alpha\to\beta}(V_\beta - E_{\alpha\to\beta})\sum_j s_\alpha^j, \tag{1}$$

where $g_{\alpha\to\beta}$ is the maximum synaptic conductance, $E_{\alpha\to\beta}$ is the reversal potential, and $\sum_j s_\alpha^j$ denotes the sum of synaptic conductances from the presynaptic cell j. The synaptic gating variable s_α is defined as follows.

(i) For STN→GPe, STN→GPi, and GPi→TH synapses, s_α obeys the second-order differential equation

$$\frac{\mathrm{d}s}{\mathrm{d}t} = z, \qquad \frac{\mathrm{d}z}{\mathrm{d}t} = 0.234\,u(t) - 0.4z - 0.04s, \tag{2}$$

where $u(t) = 1$ if the presynaptic membrane potential crosses -10 mV at time t, indicating an action potential, and $u(t) = 0$ otherwise.

(ii) For GPe→STN, GPe→GPi, and GPe→GPe synapses, s_α follows

$$\frac{\mathrm{d}s}{\mathrm{d}t} = 2(1-s)H_\infty(v-20) - 0.04s, \tag{3}$$

$$H_\infty(v) = \frac{1}{1+\exp\left(-\frac{v+57}{2}\right)}. \tag{4}$$

Throughout the model, membrane potentials are measured in mV, conductances in $\text{nS}/\mu\text{m}^2$, currents in $\text{pA}/\mu\text{m}^2$, and the time constants τ_i in ms.

The membrane potential dynamics of an STN neuron are given by

$$\begin{aligned}
C_m\frac{\mathrm{d}v}{\mathrm{d}t} &= -I_{\text{L}} - I_{\text{Na}} - I_{\text{K}} - I_{\text{T}} - I_{\text{Ca}} - I_{\text{ahp}} \\
&\quad - I_{\text{GPe}\to\text{STN}} + I_{\text{app}_\text{STN}}, \\
\frac{\mathrm{d}h}{\mathrm{d}t} &= 0.75\,\frac{h_\infty(v)-h}{\tau_h(v)}, \\
\frac{\mathrm{d}n}{\mathrm{d}t} &= 0.75\,\frac{n_\infty(v)-n}{\tau_n(v)}, \\
\frac{\mathrm{d}r}{\mathrm{d}t} &= 0.2\,\frac{r_\infty(v)-r}{\tau_r(v)}, \\
\frac{\mathrm{d}c}{\mathrm{d}t} &= 0.08\,\frac{c_\infty(v)-c}{\tau_c(v)}, \\
\frac{\mathrm{d}[\text{Ca}]}{\mathrm{d}t} &= 3.75\times10^{-5}\left(-I_{\text{Ca}} - I_{\text{T}} - 22.5[\text{Ca}]\right).
\end{aligned} \tag{5}$$

GPe and GPi neurons are modeled similarly. Their membrane potentials satisfy

$$\begin{aligned}
C_m\frac{\mathrm{d}v}{\mathrm{d}t} &= -I_{\text{L}} - I_{\text{Na}} - I_{\text{K}} - I_{\text{T}} - I_{\text{Ca}} - I_{\text{ahp}} \\
&\quad - I_{\text{STN}\to\text{GP}} + I_{\text{GPe}\to\text{GPe/GPi}} + I_{\text{app}_\text{GPe/GPi}}, \\
\frac{\mathrm{d}h}{\mathrm{d}t} &= 0.75\,\frac{h_\infty(v)-h}{\tau_h(v)}, \\
\frac{\mathrm{d}n}{\mathrm{d}t} &= 0.75\,\frac{n_\infty(v)-n}{\tau_n(v)}, \\
\frac{\mathrm{d}r}{\mathrm{d}t} &= 0.2\,\frac{r_\infty(v)-r}{30}, \\
\frac{\mathrm{d}[\text{Ca}]}{\mathrm{d}t} &= 10^{-4}\left(-I_{\text{Ca}} - I_{\text{T}} - 15[\text{Ca}]\right).
\end{aligned} \tag{6}$$

Both STN and GPi are clinically validated DBS targets.[3,34] Inhibitory synaptic currents from GPi to TH can induce abnormal TH firing patterns, contributing to PD pathophysiology. Direct stimulation of GPi allows direct modulation of GPi output and simplifies the control strategy. Furthermore, GPi is anatomically larger than STN, offering safer surgical access.[10] Considering these advantages, we choose GPi as the stimulation target for closed-loop DBS, consistent with previous studies.[10,33,35]

The DBS current I_{DBS} injected into GPi is modeled as

$$\begin{aligned}
C_m\frac{\mathrm{d}v}{\mathrm{d}t} &= -I_{\text{L}} - I_{\text{Na}} - I_{\text{K}} - I_{\text{T}} - I_{\text{Ca}} - I_{\text{ahp}} - I_{\text{STN}\to\text{GP}} \\
&\quad + I_{\text{GPe}\to\text{GPi}} + I_{\text{app}_\text{GPi}} + I_{\text{DBS}}.
\end{aligned} \tag{7}$$

In general, I_{DBS} takes the form

$$\begin{aligned}
I_{\text{DBS}} &= i_{\text{DBS}} H\left(\sin\left(\frac{2\pi t}{\rho_{\text{DBS}}}\right)\right) \\
&\quad \times\left(1 - H\left(\sin\left(2\pi\frac{t+\delta_{\text{DBS}}}{\rho_{\text{DBS}}}\right)\right)\right),
\end{aligned} \tag{8}$$

where H is the Heaviside step function,

$$H(v) = \begin{cases} 0, & v < 0, \\ 1, & v \geqslant 0. \end{cases} \tag{9}$$

Here, i_{DBS} is the pulse amplitude, ρ_{DBS} is the period, and $\delta_{\text{DBS}} = 0.6$ ms is the fixed pulse duration. The stimulation frequency is $f_{\text{DBS}} = 1/\rho_{\text{DBS}}$. Since the stimulation amplitude serves as the control variable and varies with time, the fixed-amplitude expression in Eq. (8) is not directly applicable in the control strategy.

Membrane potentials of the TH cells were governed by the equations

$$\begin{aligned}
C_m\frac{\mathrm{d}v}{\mathrm{d}t} &= -I_{\text{L}} - I_{\text{Na}} - I_{\text{K}} - I_{\text{T}} - I_{\text{GPi}\to\text{TH}} + I_{\text{SMC}}, \\
\frac{\mathrm{d}h}{\mathrm{d}t} &= \frac{h_\infty(v)-h}{\tau_h(v)}, \quad \frac{\mathrm{d}r}{\mathrm{d}t} = \frac{r_\infty(v)-r}{\tau_r(v)},
\end{aligned} \tag{10}$$

where I_{SMC} represents excitatory input from the SMC, modeled as a series of unipolar current pulses with an amplitude of 3.5 $\mu\text{A}/\text{cm}^2$ and a duration of 5 ms. The instantaneous frequency is determined by a gamma distribution with an average rate of 14 Hz and a coefficient of variation of 0.2. The details of the equations and parameters are obtained from Ref. [30] and provided in Table 1.

Table 1. Equations and parameters of the STN, GPe, GPi, and TH neuron models.

Variable	STN neuron	GPe(GPi) neuron	TH neuron
I_{L}	$2.2(v+60)$	$0.1(v+65)$	$0.05(v+70)$
I_{K}	$45n^4(v+80)$	$30n^4(v+80)$	$5(0.75(1-h))(v+75)$
I_{Na}	$37m_\infty(v)^3h(v-55)$	$120m_\infty(v)^3h(v-55)$	$3m_\infty(v)^3h(v-50)$
I_{T}	$0.5a_\infty(v)^3b_\infty(r)^2v$	$0.5a_\infty(v)^3rv$	$5p_\infty(v)^2rv$
I_{Ca}	$2c^2(v-140)$	$0.15s_\infty(v)^2(v-120)$	–
I_{ahp}	$20(v+80)\frac{[Ca]}{[Ca]+15}$	$10(v+80)\frac{[Ca]}{[Ca]+10}$	–
$I_{\mathrm{GPe}\to\mathrm{STN}}$	–	$0.5s_{\mathrm{GPe}}(v+85)$	–
$I_{\mathrm{GPe}\to\mathrm{GPe}}$	–	$0.5s_{\mathrm{GPe}}(v+85)$	–
$I_{\mathrm{GPe}\to\mathrm{GPi}}$	–	$0.5s_{\mathrm{GPe}}(v+85)$	–
$I_{\mathrm{STN}\to\mathrm{GPe}}$	$0.15s_{\mathrm{STN}}v$	–	–
$I_{\mathrm{STN}\to\mathrm{GPi}}$	$0.15s_{\mathrm{STN}}v$	–	–
$I_{\mathrm{GPi}\to\mathrm{TH}}$	–	$0.17s_{\mathrm{GPi}}(v+85)$	–
$I_{\mathrm{app_STN}}$	33/23(Healthy/PD)	–	–
$I_{\mathrm{app_GPe}}$	–	21/8(Healthy/PD)	–
$I_{\mathrm{app_GPi}}$	–	22/16(Healthy/PD)	–
$n_\infty(v)$	$\frac{1}{(1+\exp(\frac{-v-32}{8}))}$	$\frac{1}{(1+\exp(\frac{-v-50}{1}4))}$	–
$m_\infty(v)$	$\frac{1}{(1+\exp(\frac{-v-30}{1}5))}$	$\frac{1}{(1+\exp(\frac{-v-37}{1}0))}$	$\frac{1}{(1+\exp(\frac{-v-37}{7}))}$
$h_\infty(v)$	$\frac{1}{1+\exp(\frac{-v-39}{3.1})}$	$\frac{1}{1+\exp(\frac{v+58}{12})}$	$\frac{1}{1+\exp(\frac{v+41}{7})}$
$a_\infty(v)$	$\frac{1}{1+\exp(\frac{-v-63}{7.8})}$	$\frac{1}{1+\exp(\frac{-v-57}{2})}$	–
$b_\infty(r)$	$\frac{1}{1+\exp(\frac{-r+0.4}{0.1})}-\frac{1}{1+\exp(4)}$	–	–
$r_\infty(v)$	$\frac{1}{1+\exp(\frac{v+67}{2})}$	–	$\frac{1}{1+\exp(\frac{v+84}{4})}$
$c_\infty(v)$	$\frac{1}{1+\exp(\frac{-v-20}{8})}$	–	–
$s_\infty(v)$	–	$\frac{1}{1+\exp(\frac{-v-35}{2})}$	–
$p_\infty(v)$	–	–	$\frac{1}{1+\exp(\frac{-v-60}{6.2})}$
$\tau_n(v)$	$1+\frac{100}{1+\exp(\frac{v-80}{26})}$	$0.05+\frac{0.27}{1+\exp(\frac{v-40}{12})}$	–
$\tau_h(v)$	$1+\frac{500}{1+\exp(\frac{v-57}{3})}$	$0.05+\frac{0.27}{1+\exp(\frac{v-40}{12})}$	$\frac{1}{1+\exp(\frac{v+41}{4})}$
$\tau_r(v)$	$7.1+\frac{17.5}{1+\exp(\frac{v-68}{2.2})}$	–	$\frac{0.15}{28+\exp(\frac{-v-25}{10.5})}$
$\tau_c(v)$	$1+\frac{10}{1+\exp(\frac{v+80}{26})}$	–	–

2.2. Control problem description

The ability of the thalamus to accurately respond to signals from the SMC is a crucial factor for evaluating both the development of PD states and the effectiveness of treatment in the model. Therefore, the state of the BG network is generally assessed by measuring the accuracy with which TH neurons respond to inputs from the SMC. The thalamic relay reliability, denoted as Rel, was defined by Feng *et al.*[36] based on a modified error index (EI) introduced by Rubin and Terman.[32]

If each input pulse from the SMC elicits a single action potential in each TH neuron, and that action potential surpasses the −40 mV threshold in membrane voltage, it is considered a correct response; otherwise, it is deemed incorrect. Let N_{error} be the number of incorrect responses within an interval and N_{SMC} the total number of SMC inputs. Then Rel is defined as

$$\text{Rel} = 1 - \frac{N_{\text{error}}}{N_{\text{SMC}}}. \tag{11}$$

In this paper, we use Rel as the primary indicator for evaluating the effectiveness of PD state control; the closer Rel is to 1, the more effective the control strategy. Besides thalamus relay reliability, the energy consumed during stimulation is also important for evaluating controller performance. Ideally, the closed-loop control scheme should lead to a lower energy cost. To quantitatively assess energy consumption, we adopt the approach outlined in Ref. [36], which measures energy cost using the root mean square (RMS) of the stimulation current

$$I_{\text{RMS}} = \sqrt{\sum_{n=1}^{N} \frac{|I_{\text{DBS}}(n)|^2}{N}}, \tag{12}$$

where n is the time-step index and N is the total number of time steps.

To achieve closed-loop control, we choose the synaptic variable s_{GPi} from $I_{\text{GPi}\to\text{TH}}$ as the feedback variable in the control strategy. s_{GPi} is treated as a computable quantity in the model, and clinically, it can be estimated from local field potentials or spike timing, making it a feasible feedback signal for closed-loop DBS.[33,34] This ensures the rationality of using it as a feedback variable. Additionally, relying on a single feedback source allows both the stimulator and sensing hardware to be located within the same brain region, potentially reducing surgical complexity while enabling more direct modulation of abnormal TH firing patterns.[10]

Based on the above considerations, our control objective is to design a closed-loop controller that uses s_{GPi} as feedback, ensuring that Rel remains close to the normal state while minimizing I_{RMS}.

2.3. Yogi-PIDNN controller design

PIDNN is a three-layer feedforward neural network comprising proportional (P), integral (I), and derivative (D) neurons in the hidden layers of each subnetwork, enabling it to function similarly to a PID controller. For a controlled system with n inputs and s outputs, the PIDNN consists of $2n$, $3n$, and s neurons in its respective layers. Let $r_1, r_2, \cdots, r_n$ denote the reference inputs of the control system, $y_1, y_2, \cdots, y_n$ denote the outputs of the controlled system, and $u_1, u_2, \cdots, u_s$ represent the control outputs of the PIDNN controller. The control output is determined by various control strategies. The specific PIDNN structure is shown in Fig. 2.[16]

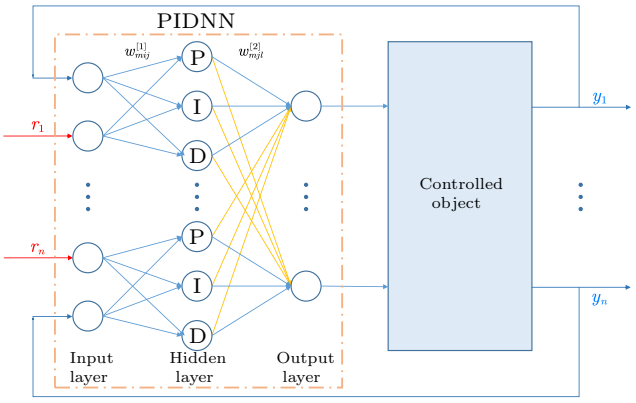


Fig. 2. Structure of PID neural network.

Integrate $r_m, y_m, m = 1,2,3\ldots,n$ as the output of the input layer, denoted as x_{m1}, x_{m2}, where m represents the m-th subnet, which is expressed as follows:

$$x_{m1}(k) = r_m(k), \quad m = 1,2,3,\ldots,n, \tag{13}$$

$$x_{m2}(k) = \begin{cases} y_{m\max}, & \text{if } y_m(k) \geq y_{m\max}, \\ y_m(k), & \text{if } y_{m\min} < y_m(k) < y_{m\max}, \\ y_{m\min}, & \text{if } y_m(k) \leq y_{m\min}, \end{cases}$$
$$m = 1,2,3,\ldots,n, \tag{14}$$

where k is the sampling time. In particular, $y_{m\min}$ and $y_{m\max}$ denote the lower and upper bounds of $y_m(k)$ to prevent unstable control outputs caused by extreme inputs.

The input of the hidden layer is denoted as $H_{mj}(k)$, which can be directly obtained by

$$H_{mj}(k) = \sum_{i=1}^{2} w_{mij}^{(1)}(k) x_{mi}(k),$$
$$j = 1,2,3, \quad m = 1,2,3,\ldots,n, \tag{15}$$

where $w_{mij}^{(1)}$ represents the connection weight between the input layer and the hidden layer.

The outputs corresponding to the P, I, and D neurons in the hidden layer of the m-th subnetwork are denoted as $h_{m1}(k)$, $h_{m2}(k)$, and $h_{m3}(k)$, and are given by

$$h_{m1}(k) = \begin{cases} h_{m1\max}, & \text{if } h_{m1}(k) \geq h_{m1\max}, \\ H_{m1}(k), & \text{if } h_{m1\min} < h_{m1}(k) < h_{m1\max}, \\ h_{m1\min}, & \text{if } h_{m1}(k) \leq h_{m1\min}, \end{cases} \tag{16}$$

$$h_{m2}(k) = \begin{cases} h_{m2\max}, & \text{if } h_{m2}(k) \geq h_{m2\max}, \\ H_{m2}(k) + h_{m2}(k-1), & \text{if } h_{m2\min} < h_{m2}(k) < h_{m2\max}, \\ h_{m2\min}, & \text{if } h_{m2}(k) \leq h_{m2\min}, \end{cases} \tag{17}$$

$$h_{m3}(k) = \begin{cases} h_{m3\max}, & \text{if } h_{m3}(k) \geq h_{m3\max}, \\ H_{m3}(k) - h_{m3}(k-1), & \text{if } h_{m3\min} < h_{m3}(k) < h_{m3\max}, \\ h_{m3\min}, & \text{if } h_{m3}(k) \leq h_{m3\min}, \end{cases} \tag{18}$$

where $h_{mj\max}$ and $h_{mj\min}$ $(j = 1,2,3)$ represent the upper and lower bounds of h_{mj}.

The input of each output neuron is defined as

$$u_l(k) = \sum_{m=1}^{n}\sum_{j=1}^{3} w_{mjl}^{(2)} h_{mj}, \quad l = 1,2,3,\cdots,s, \tag{19}$$

where $w_{mjl}^{(2)}$ represents the connection weight between the hidden layer and the output layer.

The learning performance of PIDNN is quantified by $L(k)$, defined as

$$L(k) = \frac{1}{2}\sum_{m=1}^{n}[e_m(k)]^2 = \frac{1}{2}\sum_{m=1}^{n}[r_m(k) - y_m(k)]^2. \tag{20}$$

The weight update of a general PIDNN is based on

$$\frac{\partial L(k)}{\partial w_{mjl}^{(2)}(k)} = -h_{mj}(k)\sum_{m=1}^{n} e_m(k)\,\mathrm{sign}\left(\frac{\Delta y_m(k)}{\Delta u_l(k)}\right), \tag{21}$$

$$\frac{\partial L(k)}{\partial w_{mij}^{(1)}(k)} = -x_{mi}(k)\,w_{mjl}^{(2)}(k)\mathrm{sign}\left(\frac{\Delta h_{mj}(k)}{\Delta H_{mj}(k)}\right) \times \sum_{m=1}^{n} e_m(k)\,\mathrm{sign}\left(\frac{\Delta y_m(k)}{\Delta u_l(k)}\right), \tag{22}$$

where $\Delta y_m(k) = y_m(k) - y_m(k-1)$, $\Delta u_l(k) = u_l(k) - u_l(k-1)$, $\Delta h_{mj}(k) = h_{mj}(k) - h_{mj}(k-1)$, and $\Delta H_{mj}(k) = H_{mj}(k) - H_{mj}(k-1)$.

The Yogi optimizer, a variant of Adam that employs an additive update to the second-moment estimate of gradients, is well suited to addressing convergence challenges in high-dimensional nonlinear systems.[25] We incorporate the Yogi algorithm into PIDNN for weight updates to improve the computational efficiency and accuracy of PIDNN. The pseudocode for the Yogi algorithm applied to weights $w(k)$ is provided in Algorithm 1.

Based on Eqs. (21) and (22), together with Algorithm 1, the updated weight adjustment rule can be written as

$$w_{mjl}^{(2)}(k+1) = w_{mjl}^{(2)}(k) - \eta_2 \frac{\hat{v}_2(k)}{\sqrt{\hat{s}_2(k)+\varepsilon}},$$

$$w_{mij}^{(1)}(k+1) = w_{mij}^{(1)}(k) - \eta_1 \frac{\hat{v}_1(k)}{\sqrt{\hat{s}_1(k)+\varepsilon}}. \tag{23}$$

Algorithm 1: Yogi

Input: $w(0) \in \mathbb{R}^d$, learning rate η, parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-6}$

Set $v(0) = 0$, $s(0) = 0$

for $k = 1$ to K **do**

Compute $g(k) = \nabla L_{w(k-1)}(k-1)$.

$v(k) = \beta_1 v(k-1) + (1-\beta_1)g(k)$,

$s(k) = s(k-1) - (1-\beta_2)\mathrm{sign}(s(k-1) - g^2(k))g^2(k)$,

$\hat{v}(k) = \frac{v(k)}{1-\beta_1^k}$, $\hat{s}(k) = \frac{s(k)}{1-\beta_2^k}$,

$w(k+1) = w(k) - \eta\frac{\hat{v}(k)}{\sqrt{\hat{s}(k)+\varepsilon}}$.

end for

We first employed PIDNN as a controller for feedback control design. In contrast to previous studies on closed-loop control, PIDNN can simultaneously incorporate multiple variables into the feedback process. Therefore, we define the reference input as $r_m = \bar{s}'_{m\mathrm{GPi}}$, $m = 1,2,3,\ldots,10$, where $\bar{s}'_{m\mathrm{GPi}}$, $m = 1,2,3,\ldots,10$ denotes the time-averaged value of the m-th GPi neuron's s_{GPi} in the normal state. Let $y_m(k) = s_{m\mathrm{GPi}}(k)$, $m = 1,2,3,\ldots,10$, where $s_{m\mathrm{GPi}}(k)$ represents the s_{GPi} activity of the corresponding GPi neuron in the PD network at time k. Based on the structures shown in Figs. 1 and 2, the basic closed-loop feedback system is presented in Fig. 3.

Figure 4 illustrates the closed-loop control results for the BG network using both PIDNN and Yogi-PIDNN controllers with different learning rates. Due to the high-dimensional and nonlinear characteristics of the BG network, conventional PIDNN fails to achieve satisfactory control performance. Moreover, because the initial membrane potential in the BG model is randomly generated and Gaussian noise is present, the PIDNN controller does not consistently yield effective results across multiple simulations. Although introducing the Yogi algorithm improves PIDNN performance to some extent, the energy consumption required to achieve effective control remains unacceptably high.

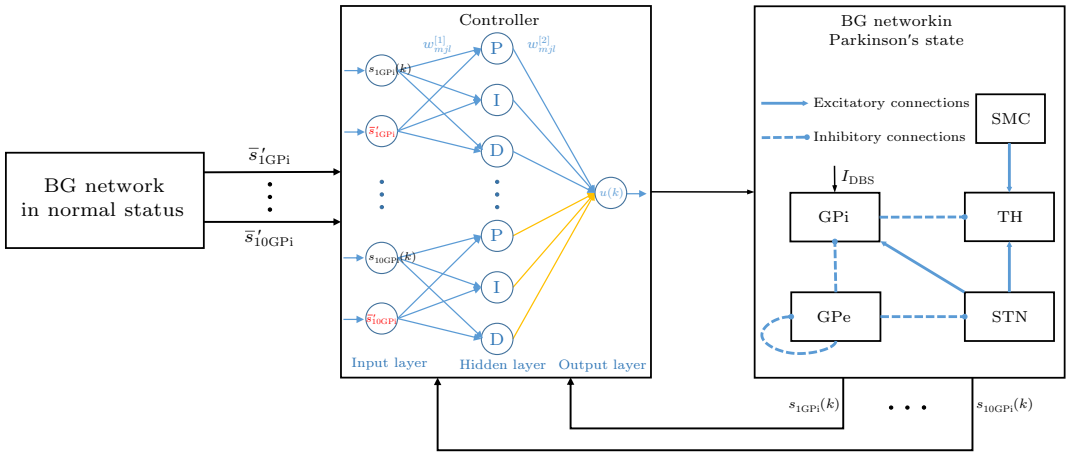


Fig. 3. Block diagram of the basic closed-loop control architecture based on PIDNN.

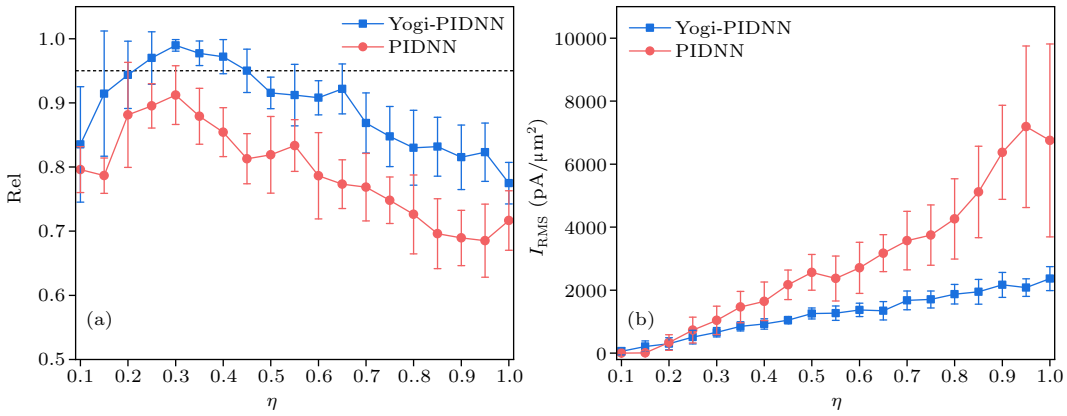


Fig. 4. Performance comparison of PIDNN and Yogi-PIDNN controllers in closed-loop DBS. The horizontal axis represents the base learning rate η, applied uniformly via the Yogi optimizer, which adapts effective rates per weight to ensure robust performance. (a) Thalamic relay reliability Rel under different learning rates η. The dashed line indicates the thalamic relay reliability threshold for healthy conditions (Rel > 0.95). (b) Energy consumption I_{RMS} under corresponding learning rates η. Error bars represent standard deviations from 10 independent simulations.

2.4. Closed-loop adaptive Yogi-PIDNN control strategy based on system identification

Böttcher *et al.*[29] demonstrated that the AI Pontryagin framework produces low-energy control signals when initialized with small weights and trained using smooth gradient descent updates, even without an explicit energy regularization term in the loss function. Inspired by this finding, we propose conducting multi-iteration gradient descent pre-training before each control signal output to reduce energy consumption. Specifically, let ι denote the gradient descent iteration index, with neural network weights at time step k denoted as $w(k,\iota)$ and the corresponding network output as $\hat{u}(w(k,\iota))$. For a fixed k, given a small weight update $\Delta w(k,\iota)$, the control signal can be approximated as

$$\hat{u}(w(k,\iota+1)) = \hat{u}(w(k,\iota)) + \mathcal{J}_{\hat{u}(k,\iota)}\Delta w(k,\iota), \tag{24}$$

where $\mathcal{J}_{\hat{u}(k,\iota)}$ is the Jacobian matrix of $\hat{u}(k,\iota)$, with elements $\frac{\partial \hat{u}(k,\iota)}{\partial w(k,\iota)}$. Since the weight update follows $\Delta w(k,\iota) = -\eta \nabla_{w(k,\iota)} L$ and $\nabla_{w(k,\iota)} L = \mathcal{J}^{\top}_{\hat{u}(k,\iota)} \nabla_{\hat{u}(k,\iota)} L$, we obtain

$$\hat{u}(w(k,\iota+1)) = \hat{u}(w(k,\iota)) - \eta \mathcal{J}_{\hat{u}(k,\iota)} \mathcal{J}^{\top}_{\hat{u}(k,\iota)} \nabla_{\hat{u}(k,\iota)} L. \tag{25}$$

By initializing with small weights $w(k,0)$, a small initial control signal $\hat{u}(k,0)$ is obtained. Subsequently, by constraining the learning rate η, the control signal $\hat{u}(k,\iota)$ is smoothly adjusted toward the target state, thereby reducing energy consumption. However, because the state of the controlled system evolves irreversibly, repeated gradient descent training using actual system states is infeasible.

To enable online pre-training of the PIDNN, we employ system identification methods to establish the input–output relationship of the BG network. This relationship serves as an input–output model for the online training process, as described below:

$$\begin{aligned}&(1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{n_a} z^{-n_a}) y_m(k) \\ =\ & (b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{n_b} z^{-n_b}) u(k) + \xi(k),\end{aligned} \tag{26}$$

where $y_m(k)$, $u(k)$, and $\xi(k)$ correspond to $s_{m\mathrm{GPi}}(k)$, the stimulation current I_{DBS}, and Gaussian noise in the BG model, respectively. The integers n_b and n_a denote the input and output orders. Therefore, we define

$$\begin{aligned}\theta_m &= [a_{m1}\ \ a_{m2}\ \cdots\ a_{mn}\ \ b_{m0}\ \ b_{m1}\ \cdots\ \ b_{mn}]^{\mathrm{T}}, \\ \varphi_m(k) &= [-s_{m\mathrm{GPi}}(k-1) \cdots\ -s_{m\mathrm{GPi}}(k-n_a)\ \ I_{\mathrm{DBS}}(k) \\ &\quad \cdots\ \ I_{\mathrm{DBS}}(k-n_b)]^{\mathrm{T}}.\end{aligned} \tag{27}$$

According to Eq. (26), we obtain

$$\hat{s}_{m\mathrm{GPi}}(k) = \varphi_m^{\mathrm{T}}(k)\theta_m + \xi(k). \tag{28}$$

We use recursive least squares (RLS)[37] to obtain the estimated parameter $\hat{\theta}_m$, so the estimate of $\hat{s}_{m\mathrm{GPi}}$ for the k-th observation of the BG network is

$$\hat{s}_{m\mathrm{GPi}}(k) = \varphi_m^{\mathrm{T}}(k)\hat{\theta}_m. \tag{29}$$

Using Eq. (29), we estimate $s_{m\mathrm{GPi}}$ based on the current input signal $u(k)$ and compute the estimation error $\hat{e}_m(k) = \vec{s}'_{m\mathrm{GPi}} - \hat{s}_{m\mathrm{GPi}}(k)$. Consequently, the performance evaluation function $\hat{L}(k)$ for the current system is given by

$$\hat{L}(k) = \frac{1}{2}\sum_{m=1}^{n} (\hat{e}_m(k))^2. \tag{30}$$

We then apply the Yogi-PIDNN to obtain the control input signal that minimizes $\hat{L}(k)$. After reaching the maximum iteration number $\iota_{\max}$, the optimal control signal is applied to the BG network. Because system identification and neural network training are performed concurrently throughout the control process, the closed-loop control strategy exhibits inherent adaptability. This approach effectively mitigates noise and nonlinear effects in the BG network. Ultimately, the closed-loop adaptive Yogi-PIDNN control strategy is obtained, as depicted in Fig. 5.

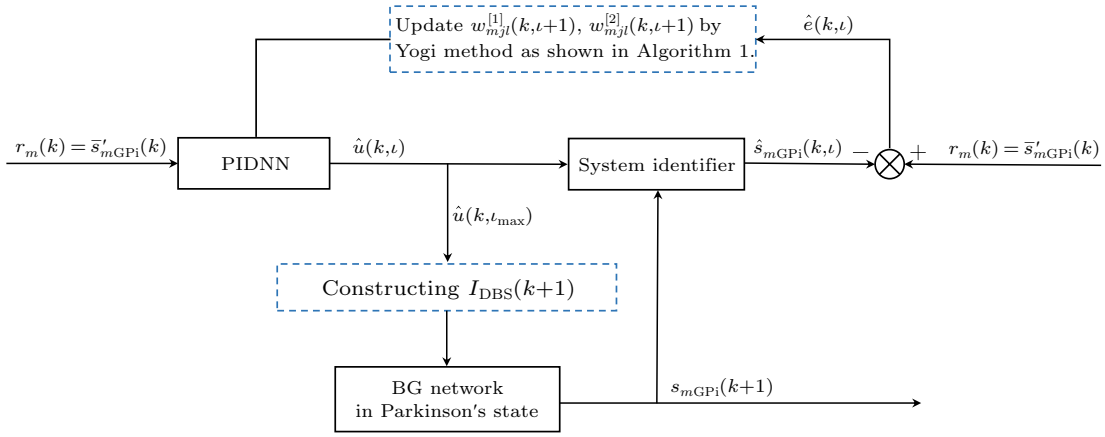


Fig. 5. Schematic diagram of the closed-loop adaptive Yogi-PIDNN control strategy for the BG network based on system identification. The PIDNN weights are updated using the Yogi algorithm, with the estimated error $\hat{e}(k)$ employed for iterative training of the system identified via Eq. (26).

To demonstrate the reliability of the designed closed-loop system, it is necessary to ensure the convergence of the estimation error $\hat{e}(k)$, indicating that the input–output system obtained through system identification remains stable when driven by the control signal generated by the Yogi-PIDNN.

Theorem 1 The control signals obtained from Yogi-PIDNN training can stabilize the controlled input–output system if the learning rates η_1 and η_2 take the same value η and satisfy the following conditions:

(i) if $\sum_{m=1}^{n}\left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)\hat{e}_m(k,\iota-1)>0$, then

$$0<\eta<\frac{-2\sum_{m=1}^{n}\left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)\hat{e}_m(k,\iota-1)}{\sum_{m=1}^{n}\left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)}, \tag{31}$$

(ii) if $\sum_{m=1}^{n}\left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)\hat{e}_m(k,\iota-1)<0$, then

$$0>\eta>\frac{-2\sum_{m=1}^{n}\left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)\hat{e}_m(k,\iota-1)}{\sum_{m=1}^{n}\left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)}, \tag{32}$$

where

$$Q^{(i)}(k,\iota-1)=-\sum_{m=1}^{n}\left[\sum_{j=1}^{3}\sum_{i=1}^{2}Q_i(k,\iota-1)\frac{\hat{v}^{(i)}(k,\iota-1)}{\sqrt{\hat{s}^{(i)}(k,\iota-1)+\varepsilon}}\right], \quad i=1,2, \tag{33}$$

$$Q_{1m}(k,\iota-1)=-x_{mi}(k)\,w^{(2)}_{mjl}(k)\mathrm{sign}\left(\frac{\Delta h_{mj}(k)}{\Delta H_{mj}(k)}\right)\times\mathrm{sign}\left(\frac{\Delta y_m(k)}{\Delta u_l(k)}\right), \tag{34}$$

$$Q_{2m}(k,\iota-1)=-h_{mj}(k)\,\mathrm{sign}\left(\frac{\Delta y_m(k)}{\Delta u_l(k)}\right). \tag{35}$$

Proof First we define the Lyapunov function as

$$V(k,\iota)=\frac{1}{2}\|\hat{e}_m(k,\iota)\|^2=\frac{1}{2}\sum_{m=1}^{n}(\hat{e}_m(k,\iota)^2). \tag{36}$$

Thus, the change of the Lyapunov function is

$$\begin{aligned}\Delta V(k,\iota)&=V(k,\iota)-V(k,\iota-1)\\&=\frac{1}{2}\sum_{m=1}^{n}(\hat{e}_m(k,\iota)^2-\hat{e}_m(k,\iota-1)^2).\end{aligned} \tag{37}$$

We define the variation of the error as follows:

$$\Delta\hat{e}_m(k,\iota)=\hat{e}_m(k,\iota)-\hat{e}_m(k,\iota-1). \tag{38}$$

Hence, equation (37) can be rewritten as

$$\Delta V(k,\iota)=\sum_{m=1}^{n}\left(\Delta\hat{e}_m(k,\iota)\hat{e}_m(k,\iota-1)+\frac{1}{2}\Delta\hat{e}_m(k,\iota)^2\right). \tag{39}$$

We can rewrite Eq. (38) as

$$\begin{aligned}\Delta\hat{e}_m(k,\iota)=&\sum_{m=1}^{n}\left(\sum_{j=1}^{3}\sum_{i=1}^{2}\frac{\partial\hat{e}_m(k,\iota-1)}{\partial w^{(1)}_{mij}(k,\iota-1)}\Delta w^{(1)}_{mij}(k,\iota-1)\right)\\&+\sum_{m=1}^{n}\left(\sum_{j=1}^{3}\sum_{i=1}^{2}\frac{\partial\hat{e}_m(k,\iota-1)}{\partial w^{(2)}_{mjl}(k,\iota-1)}\Delta w^{(2)}_{mjl}(k,\iota-1)\right).\end{aligned} \tag{40}$$

From Eq. (23), we have $\Delta w^{(1)}_{mij}(k,\iota-1)=-\eta_1\frac{\hat{v}^{(1)}(k,\iota-1)}{\sqrt{\hat{s}^{(1)}(k,\iota-1)+\varepsilon}}$, $\Delta w^{(2)}_{mjl}(k,\iota-1)=-\eta_2\frac{\hat{v}^{(2)}(k,\iota-1)}{\sqrt{\hat{s}^{(2)}(k,\iota-1)+\varepsilon}}$. To simplify the expression formally, we introduce variables $Q_{1m}(k,\iota-1)$ and $Q_{2m}(k,\iota-1)$ as follows:

$$\begin{aligned}Q_{1m}(k,\iota-1)&=\frac{\partial\hat{e}_m(k,\iota-1)}{\partial w^{(1)}_{mij}(k,\iota-1)}\\&=-x_{mi}(k)\,w^{(2)}_{mjl}(k)\mathrm{sign}\left(\frac{\Delta h_{mj}(k)}{\Delta H_{mj}(k)}\right)\mathrm{sign}\left(\frac{\Delta y_m(k)}{\Delta u_l(k)}\right),\end{aligned} \tag{41}$$

$$\begin{aligned}Q_{2m}(k,\iota-1)&=\frac{\partial\hat{e}_m(k,\iota-1)}{\partial w^{(2)}_{mjl}(k,\iota-1)}\\&=-h_{mj}(k)\,\mathrm{sign}\left(\frac{\Delta y_m(k)}{\Delta u_l(k)}\right).\end{aligned} \tag{42}$$

Then, Eq. (40) can be represented as

$\Delta\hat{e}_m(k,\iota)$

$$= -\eta_1 \sum_{m=1}^{n} \left[\sum_{j=1}^{3} \sum_{i=1}^{2} Q_{1m}(k,\iota-1) \frac{\hat{v}^{(1)}(k,\iota-1)}{\sqrt{\hat{s}^{(1)}(k,\iota-1)+\varepsilon}} \right]$$
$$-\eta_2 \sum_{m=1}^{n} \left[\sum_{j=1}^{3} \sum_{i=1}^{2} Q_{2m}(k,\iota-1) \frac{\hat{v}^{(2)}(k,\iota-1)}{\sqrt{\hat{s}^{(2)}(k,\iota-1)+\varepsilon}} \right]. \quad (43)$$

Let

$$Q^{(i)}(k,\iota-1) = -\sum_{m=1}^{n} \left[\sum_{j=1}^{3} \sum_{i=1}^{2} Q_{im}(k,\iota-1) \frac{\hat{v}^{(i)}(k,\iota-1)}{\sqrt{\hat{s}^{(i)}(k,\iota-1)+\varepsilon}} \right], \quad i=1,2.$$

According to Eq. (43), we rewrite Eq. (39) as

$$\Delta V(k,\iota) = \sum_{m=1}^{n} \{[\eta_1 Q^{(1)}(k,\iota-1)+\eta_2 Q^{(2)}(k,\iota-1)]\hat{e}_m(k,\iota-1) + \frac{1}{2}[\eta_1 Q^{(1)}(k,\iota-1)+\eta_2 Q^{(2)}(k,\iota-1)]^2\}. \quad (44)$$

Let the learning rates η_1 and η_2 take the same value η. If we want $\Delta V < 0$, the following conditions should be satisfied: (i) if $\sum_{m=1}^{n} \left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)\hat{e}_m(k,\iota-1) > 0$, then

$$0 < \eta < \frac{-2\sum_{m=1}^{n} \left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)\hat{e}_m(k,\iota-1)}{\sum_{m=1}^{n} \left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)}, \quad (45)$$

(ii) if $\sum_{m=1}^{n} \left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)\hat{e}_m(k,\iota-1) < 0$, then

$$0 > \eta > \frac{-2\sum_{m=1}^{n} \left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)\hat{e}_m(k,\iota-1)}{\sum_{m=1}^{n} \left(Q^{(1)}(k,\iota-1)+Q^{(2)}(k,\iota-1)\right)}. \quad (46)$$

3. Results

3.1. Preliminary simulations

The BG network model transitions from the normal state to the PD state by reducing I_{app}, resulting in firing-rate changes consistent with experimental observations. As shown in Fig. 6(a), in the normal state, s_{GPi} exhibits slight irregular oscillations, while the STN, GPe, and GPi neurons show tonic spiking behavior. The excitatory input from the SMC is represented by red pulses. TH accurately responds to SMC input, meaning that each SMC pulse induces a single action potential in the thalamic neuron. In this state, the thalamic relay reliability is Rel = 0.9861. The neuronal firing characteristics in the Parkinsonian state and the corresponding changes in s_{GPi} are shown in Fig. 6(b). In this state, GPi and GPe exhibit synchronized burst firing, while the dynamics of s_{GPi} evolve into large-amplitude periodic oscillations that disrupt thalamic function. As a result, TH neurons cannot reliably follow excitatory SMC inputs, reducing the relay reliability to Rel = 0.5722.

Figure 6(c) shows the effects of classical open-loop DBS targeting the GPi. We injected a pulse sequence with duration $\delta = 0.6$ ms, frequency $f = 130$ Hz, and amplitude $i_{\text{DBS}} = 200$ pA/μm^2 into the GPi nucleus.[32] Under these conditions, the pathological GPi firing rhythm is replaced by a stimulation-locked high-frequency discharge pattern. The large-amplitude oscillations of s_{GPi} collapse into small-range high-frequency oscillations. Thalamic relay reliability is fully restored, with Rel = 1. The energy consumption of open-loop DBS under these parameters is $I_{\text{RMS}} = 55.9284$ pA/μm^2.

3.2. System identification

We selected $n_a = 4$ and $n_b = 2$, enabling the identified system to fit the BG network data with high accuracy. The orders $n_a = 4$ and $n_b = 2$ were chosen empirically based on validation datasets from basal ganglia simulations, selecting the configuration that minimized the average estimation error $\bar{E}$ while preserving computational efficiency. For the GPi-DBS state of interest, the identification results are shown in Fig. 7(a). The dotted blue trace represents the output generated by the identified system, closely matching the ground-truth BG network output shown in red. During identification, Figs. 7(b) and 7(c) display the evolution of the corresponding parameters in Eq. (26).

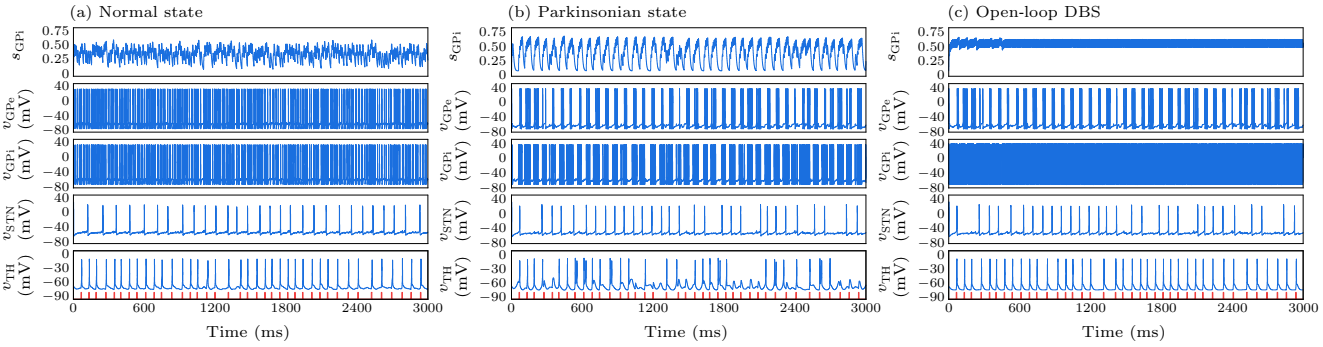


Fig. 6. The s_{GPi} and the membrane potential dynamic changes of neuronal cells in different core cluster of the BG network. The red pulsed signal indicates excitatory input from the SMC to the TH. (a) s_{GPi} and membrane potentials of neurons in different nuclei in the normal state, with Rel = 0.9861. (b) s_{GPi} and membrane potentials of neurons in different nuclei in the PD state, with Rel = 0.5722. (c) s_{GPi} and membrane potentials of neurons in different nuclei under open-loop DBS stimulation, with Rel = 1 and $I_{\text{RMS}} = 55.9284$ pA/μm^2.

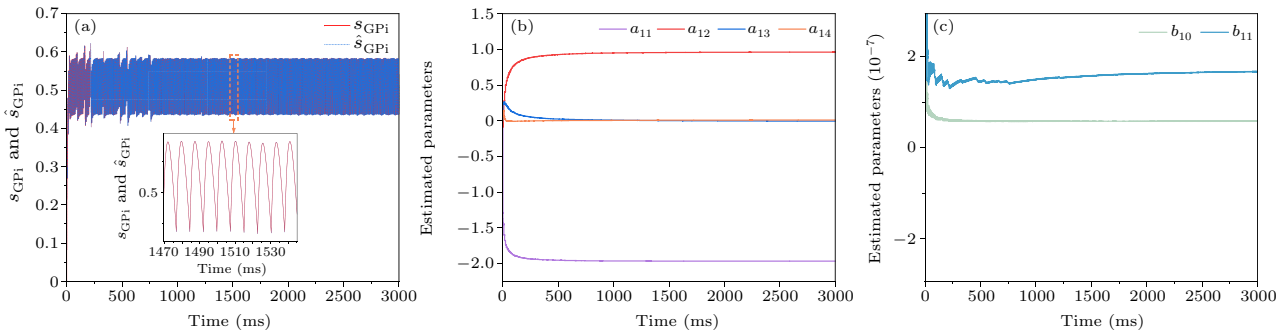


Fig. 7. System identification results of s_{GPi}. (a) Under the GPi-DBS state, the estimated $\hat{s}_{GPi}$ values are represented by the blue dotted line and closely follow the true s_{GPi} values of the BG network, indicated by the red solid line. (b)–(c) Changes in the corresponding parameters during the identification process.

In order to quantify the accuracy of the identification, we adopt the same method as in Ref. [33] to calculate the estimation error, given by

$$E_m = \sqrt{\sum_{k=1}^{N} \frac{[s_{m\mathrm{GPi}}(k) - \hat{s}_{m\mathrm{GPi}}(k)]^2}{(N-1)}}, \; m = 1,2,3,\ldots,10, \quad (47)$$

where N denotes the total length of the data. Under the GPi-DBS condition, the average value of E_m $(m = 1,2,3,\ldots,10)$ is $\bar{E} = 4.5970 \times 10^{-5}$, indicating a good identification performance.

3.3. Adaptive Yogi-PIDNN control strategy

Figure 8 demonstrates the effectiveness of the closed-loop control strategy implemented using the adaptive Yogi-PIDNN. The DBS signal has a frequency of $f = 130$ Hz, duration $\delta = 0.6$ ms, and the amplitude is determined by the output of the adaptive Yogi-PIDNN controller. During the initialization of the neural network, the learning rate was set to $\eta = 0.05$, the maximum number of iterations to $\iota_{\max} = 50$, and the initial weights were determined using the Xavier initialization method.[38]

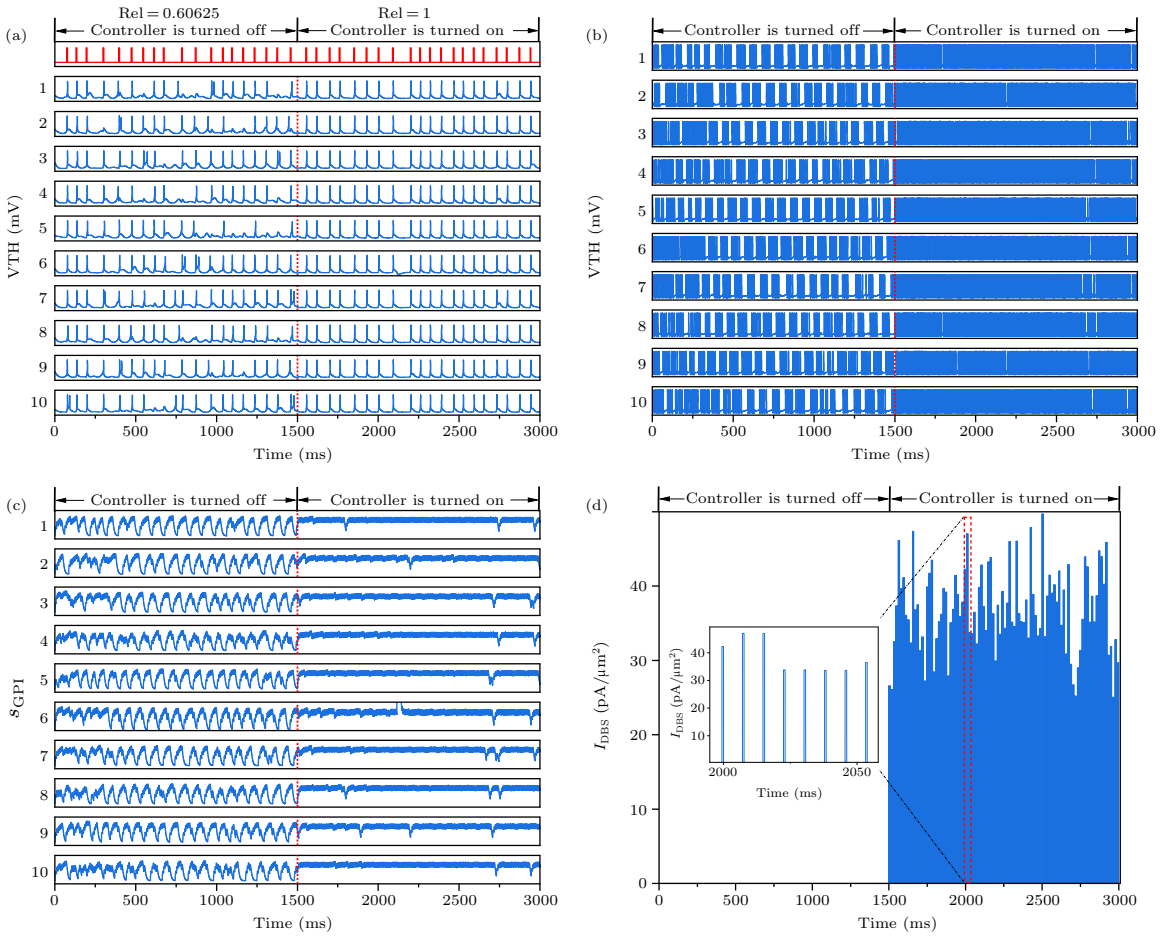


Fig. 8. Results of the adaptive Yogi-PIDNN control strategy. (a) Changes in the state of TH neurons before and after control, showing recovery of the thalamic relay reliability from Rel = 0.6025 to Rel = 1. (b) Comparison of GPi neurons before and after control. (c) Comparison of waveform changes in s_{GPi} before and after control. (d) Output of stimulation signals generated by closed-loop DBS under the adaptive Yogi-PIDNN control strategy. The controller is activated at 1500 ms, as indicated by the red dashed line in the figure. Neural network parameters are set as follows $\eta = 0.05$, $\iota_{\max} = 50$.

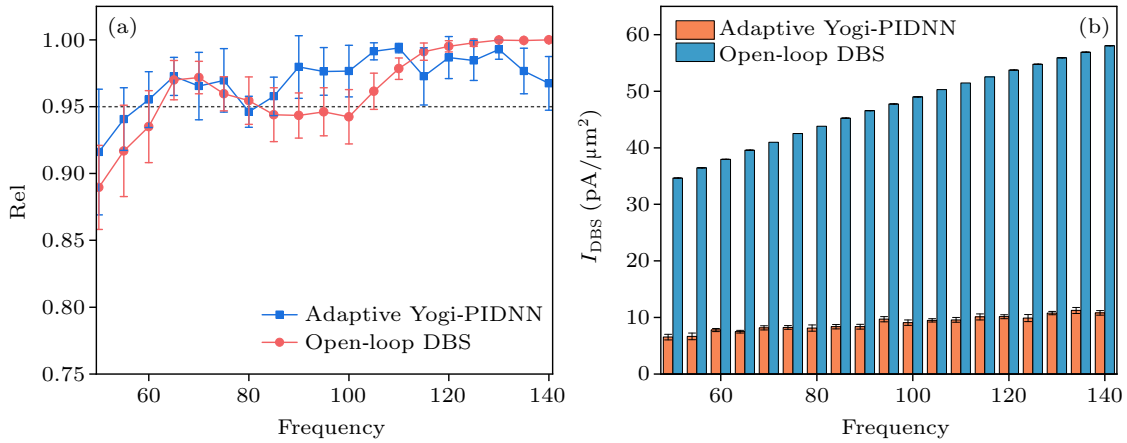


Fig. 9. Comparison of open-loop DBS and closed-loop DBS with adaptive Yogi-PIDNN control at various frequencies. (a) Thalamic relay reliability Rel under different frequencies. The dashed line indicates the thalamic relay reliability threshold under healthy conditions (Rel >0.95). (b) Energy consumption I_{RMS} under corresponding frequencies. Stimulation results are obtained from 10 independent simulations, with standard deviations represented by error bars.

The controller was activated at 1500 ms, and stimulation was applied to the GPi, as indicated by the red dashed line. As shown in Fig. 8(a), upon activation of the adaptive Yogi-PIDNN controller, the pathological state in which TH neurons cannot accurately respond to SMC input rapidly dissipates. This effectively restores the thalamic relay reliability to Rel $=1$. Concurrently, as shown in Fig. 8(b), the firing pattern of the GPi neurons transitions from pathological oscillations to a tonic spike state, resembling the behavior under open-loop DBS. The s_{GPi} oscillates within a narrow range, indicating that the pathological oscillations have been effectively suppressed, as illustrated in Fig. 8(c). These results suggest that the adaptive Yogi-PIDNN control strategy can effectively restore thalamic relay reliability.

Next, we examine the energy expenditure associated with the stimulation signal to assess the efficiency of the approach. The specific control output is shown in Fig. 8(d). It can be observed that, when the frequency is fixed, this control strategy produces a pulse sequence with relatively small amplitude, not exceeding 50 pA/μm^2 in this simulation, leading to reduced energy consumption. Using the same parameters as in the previous experiments, 10 trials were conducted. The energy consumption I_{RMS} of the adaptive Yogi-PIDNN control reached a maximum of 10.73 pA/μm^2, representing an 80.8% reduction compared to the energy consumption of open-loop DBS at the same frequency and duration.

Since the current system state is used as feedback for the iterative training updates of the PIDNN, and the effect of frequency on the system takes time to become apparent, adjusting the frequency as a control variable is challenging. Therefore, we implement our control strategies under varied frequencies to validate their robustness and to identify optimal neuromodulation parameters. Given that DBS signals below 50 Hz fail to sufficiently enhance thalamic relay reliability, while supraphysiological stimulation frequencies may induce latent neuronal injury and unnecessary energy expenditure,[30,39] we confined the stimulation frequency to the 50–140 Hz range.

As shown in Fig. 9(a), the adaptive Yogi-PIDNN controller can restore thalamic relay reliability with effects comparable to those of open-loop DBS. Notably, within the frequency range of 85 Hz to 110 Hz, our control strategy exhibits superior modulation effects. Specifically, at a stimulation frequency of 110 Hz, the recovery of thalamic relay reliability reaches its peak, with Rel $=0.9938$. Concurrently, the standard deviation of the thalamic relay reliability is minimized at this frequency, indicating enhanced stability and precision.

The corresponding energy loss is depicted in Fig. 9(b). Across various frequencies, the stimulus signals generated by our control strategy require, on average, $(81.0\pm0.7)\%$ less energy than those produced by an open-loop DBS operating at the same frequency. Particularly at $f=110$ Hz, our strategy not only achieves more effective recovery of thalamic relay reliability than an open-loop DBS but also reduces energy consumption by an additional 81.4%. This finding underscores that for square waveforms of DBS pulse currents, higher amplitude is not invariably necessary for restoring thalamic relay reliability. Even at lower frequencies, our strategy adeptly identifies stimuli that consume less energy while still delivering superior therapeutic outcomes.

To investigate the effects of the learning rate η and maximum iteration number $\iota_{\max}$ on control performance, we conducted a parametric study and visualized the results in two-dimensional plots. As shown in Fig. 10(a), a larger η enables the adaptive Yogi-PIDNN controller to generate amplitude parameters that effectively control the PD state. Specifically, for $\eta>0.1$, only 30 iterations suffice to achieve a thalamic relay reliability of Rel >0.95, with the system maintaining stability across a wide range of $\iota_{\max}$. Conversely, for smaller η, effective control (Rel >0.95) can still be achieved by increasing $\iota_{\max}$, as higher iterations compensate for slower convergence.

This aligns with the convergence properties of the Yogi optimizer, which, as defined in Algorithm 1, employs an adaptive second-moment estimate. Theorem 1 ensures robust convergence, particularly at smaller η, thereby stabilizing control outputs.

From an energy perspective, Fig. 10(b) illustrates that the energy loss of the adaptive Yogi-PIDNN control strategy increases with both η and $\iota_{\max}$. At higher η, the update term $\eta \mathcal{J}_{\hat{u}(k,\iota)} \mathcal{J}^{\top}_{\hat{u}(k,\iota)} \nabla_{\hat{u}(k,\iota)} L$ in Eq. (25) induces larger changes in $\hat{u}(k,\iota)$, resulting in higher pulse currents and elevated energy dissipation. The Yogi optimizer mitigates this effect at lower η by constraining gradient updates through its adaptive second-moment estimate, ensuring smoother $\hat{u}$ adjustments. Consequently, lower η with higher $\iota_{\max}$ yields control signals with Rel > 0.95 and reduced energy consumption, albeit at the cost of increased computational complexity. Conversely, higher η necessitates careful tuning of $\iota_{\max}$ to prevent excessive energy loss and potential neuronal stress from high-amplitude stimuli.

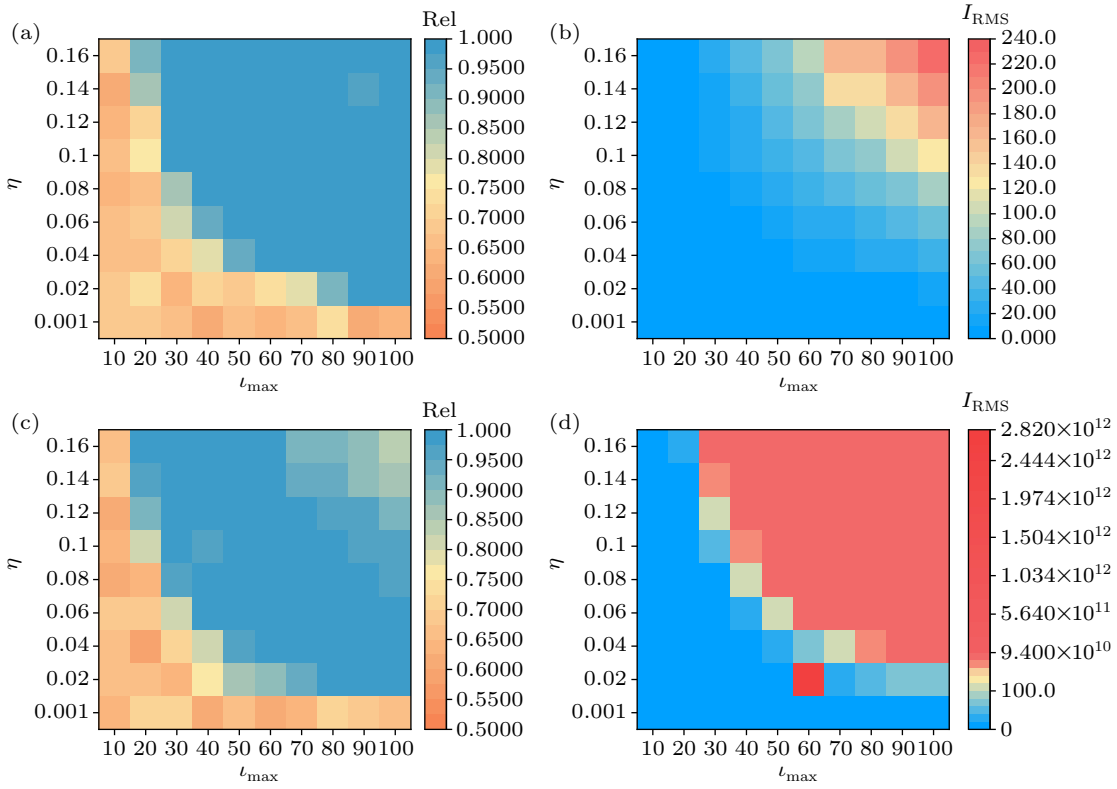


Fig. 10. Heatmaps illustrating the performance of two adaptive control strategies across the parameter space of learning rate (η) and maximum iteration number ($\iota_{\max}$). The color indicates the metric value, averaged over ten independent tests. Panels show the performance of the adaptive Yogi-PIDNN strategy in terms of (a) thalamic relay reliability (Rel) and (b) stimulus energy consumption (I_{RMS}). The same metrics for the adaptive PIDNN strategy are shown in panels (c) and (d), respectively.

To assess the necessity of the Yogi optimizer, we compared the adaptive Yogi-PIDNN with a standard adaptive PIDNN strategy, trained iteratively with varying η and $\iota_{\max}$, as shown in Figs. 10(c) and 10(d). Figure 10(c) reveals that the parameter range for achieving effective control (Rel > 0.95) is significantly narrower for the standard adaptive PIDNN, complicating practical parameter tuning. Moreover, Fig. 10(d) demonstrates that the energy loss of the standard adaptive PIDNN escalates rapidly with increasing η and $\iota_{\max}$. For instance, at $\eta = 0.002$ and $\iota_{\max} = 60$, the control signal exhibits explosive energy growth, indicating instability. This instability arises because the standard adaptive PIDNN, lacking the adaptive second-moment regulation of the Yogi algorithm, is prone to gradient explosion in complex nonlinear systems. In contrast, the adaptive Yogi-PIDNN leverages the Yogi algorithm to stabilize gradient updates, ensuring lower energy fluctuations and a broader stable parameter range. These findings underscore the critical role of the Yogi optimizer, inspired by the AI Pontryagin framework,[29] in achieving robust, energy-efficient closed-loop DBS control.

4. Discussion and conclusion

To evaluate the efficacy of our adaptive Yogi-PIDNN framework, we compare it with recently established closed-loop DBS control algorithms, all tested on computational models of PD pathophysiology. The comparative results are summarized in Table 2. On the premise of ensuring equivalent therapeutic efficacy, we focus on the technical characteristics of each control strategy and its energy consumption reduction ratio relative to open-loop strategies. Overall, the control strategy proposed in this work demonstrates competi-

tive energy optimization performance, particularly when compared with the strategies tested in the same model.[33,35] Fleming *et al.*[14] implemented a PI controller on a basal ganglia-thalamo-cortical model, achieving a 75% reduction in energy consumption relative to open-loop DBS and an 83% reduction in beta oscillation error. However, its reliance on linear control limits adaptability to nonlinear dynamics, such as those induced by medication fluctuations. In the realm of PID-dominated control strategies, the precise tuning of gain coefficients remains a critical technical challenge. Although the control strategies proposed by Zhao *et al.*[40] based on reinforcement learning achieve high adaptability and significantly reduce energy consumption, their implementation relies on extensive data pre-training. Moreover, further research is required to enhance their generalization ability and guarantee system stability. Notably, current investigations into neural network-based control strategies are in their nascent stage, accompanied by notable technical limitations. For instance, the closed-loop control strategy developed by Liu *et al.*[23] using backpropagation neural networks (BPNN) necessitates a pre-configured PID controller to generate a high-quality training dataset. This approach renders the control performance of BPNN overly dependent on the pre-set PID parameters, resulting in limited generalization capabilities and prolonged training durations. Similarly, the ARBFNN control architecture designed by Wang *et al.*[24] requires model-dependent strategies for neural network weight adjustment, and its effectiveness in complex large-scale network scenarios remains inadequately validated.

Table 2. Comparison of closed-loop DBS control algorithms for PD. The energy optimization values represent the percentage of energy consumption reduction relative to open-loop DBS. The proposed strategy achieves an 80.8% reduction compared to conventional open-loop DBS (frequency 130 Hz, amplitude 200 pA/μm^2, duration 0.6 ms).

References	Model used	Control strategy	Energy optimization
Liu *et al.* (2016)[33]	Basal ganglia model[30]	Generalized predictive control (GPC) algorithm	67.8%
Lu *et al.* (2019)[35]	Basal ganglia model[30]	Reinforcement learning	36.7%
Fleming *et al.* (2020)[14]	Cortico-basal ganglia network model[14]	PI controller	75%
Liu *et al.* (2020)[23]	Modified basal ganglia-thalamo-cortical computational model[23]	Backpropagation neural network	58.26%
Wang *et al.* (2022)[24]	A mean firing-rate model[41]	Adaptive radial basis function neural network	Not specified
Zhao *et al.* (2025)[40]	Basal ganglia-thalamic neuronal network model[42]	Reinforcement learning	80%
This work	Basal ganglia model[30]	Adaptive Yogi-PIDNN	80.8%

In contrast, our framework leverages recursive least squares system identification to construct a patient-specific input–output model of the basal ganglia network, enabling robust adaptation to nonlinear and time-varying dynamics, as discussed in Section 2. By adopting multi-iteration gradient descent pre-training, inspired by the AI Pontryagin framework,[29] the adaptive Yogi-PIDNN achieves implicit energy optimization without explicit regularization, reducing energy consumption by $(81.0 \pm 0.7)\%$ compared to open-loop DBS during multi-frequency testing, as demonstrated in Section 3. The application of the discrete Lyapunov stability theorem ensures convergence of estimation errors, providing a theoretical guarantee of control stability, a feature absent in reinforcement learning approaches and less robust in PI and state feedback methods.

Despite its advantages, the adaptive Yogi-PIDNN framework has limitations. Although the BG network model effectively represents both the healthy state and the PD state, it is still a simplified representation of complex neural networks involved in PD pathology.[43] Consequently, to further validate the efficacy and practicality of our control strategy, future research should apply it to different, more refined, and biologically accurate models[44,45] for testing. In addition, for the PIDNN itself, its reliance on current error for feedback regulation precludes the formation of an effective control strategy for waveform optimization involving multi-parameter adjustments. This overlooks the synergistic effects of multi-parameter modulation, potentially limiting adaptability to inter-individual variability in PD or advanced waveforms. Future work will integrate these variables through expanded PIDNN architectures to further optimize personalization, albeit at the cost of increased computational demands. Additionally, the reliance on system identification requires high-quality training data, which may be challenging to acquire in clinical environments with noisy or incomplete neural signals. Moreover, while the Yogi optimizer mitigates the computational burden of multi-iteration pre-training, further optimization is needed for resource-constrained DBS devices. Future research could enhance the framework by integrating multi-modal biomarkers, such as kinematic or non-motor symptom data, to improve adaptability across diverse PD phenotypes. Exploring model-free approaches based on clinical data and designing system identification methods with both rapid response capabilities and real-data adaptability could

further enhance performance. This endeavor seeks to enhance the applicability and effectiveness of our framework in real-world clinical settings. Moreover, extending simulations to incorporate additional PD symptoms and conducting large-scale clinical validations will be essential for broader adoption.

In summary, this study introduces a novel adaptive Yogi-PIDNN framework for closed-loop DBS, addressing the critical challenges of energy efficiency and personalized control in PD treatment. By integrating system identification with recursive least squares, multi-iteration gradient descent pre-training, and implicit energy optimization inspired by the AI Pontryagin framework,[29] our approach not only significantly restores thalamic functional reliability but also exhibits a notable reduction in energy consumption during multi-frequency testing. Specifically, the energy consumption of our method shows an average decrease of $(81.0 \pm 0.7)\%$ compared with that of open-loop DBS at the same frequency. The application of the discrete Lyapunov stability theorem provides a rigorous guarantee of control stability, distinguishing our framework from existing model-based algorithms. Future work will focus on validating the adaptive Yogi-PIDNN framework in clinical settings, integrating multi-modal biomarkers to enhance adaptability, and optimizing computational efficiency for real-time implementation. By addressing these challenges, our approach has the potential to advance personalized neuromodulation therapies, offering improved therapeutic outcomes for PD patients.

Acknowledgments

This project was supported by the National Natural Science Foundation of China (Grant Nos. 12372064 and 12172291) and the Youth and Middle-Aged Science and Technology Development Program of Shanghai Institute of Technology (Grant No. ZQ2024-10).

References

[1] Benabid A L 2003 *Curr. Opin. Neurobiol.* **13** 696
[2] Okun M S 2012 *N. Engl. J. Med.* **367** 1529
[3] Hariz M and Blomstedt P 2022 *J. Intern. Med.* **292** 764
[4] Limousin P and Foltynie T 2019 *Nat. Rev. Neurol.* **15** 234
[5] Lozano A M, Lipsman N, Bergman H, *et al.* 2019 *Nat. Rev. Neurol.* **15** 148
[6] Guidetti M, Marceglia S, Loh A, Harmsen I E, Meoni S, Foffani G, Lozano A M, Moro E, Volkmann J and Priori A 2021 *Brain Stimul.* **14** 1238
[7] Bouthour W, Mégevand P, Donoghue J, Lüscher C, Birbaumer N and Krack P 2019 *Nat. Rev. Neurol.* **15** 343
[8] Krauss J K, Lipsman N, Aziz T, Boutet A, Brown P, Chang J W, Davidson B, Grill W M, Hariz M I and Horn A 2021 *Nat. Rev. Neurol.* **17** 75
[9] Rosa M, Arlotti M, Ardolino G, Cogiamanian F, Marceglia S, Di Fonzo A, Cortese F, Rampini P M and Priori A 2015 *Mov. Disord.* **30** 1003
[10] Gorzelic P, Schiff S and Sinha A 2013 *J. Neural Eng.* **10** 026016
[11] Chen L, Jiang W, Bin D, *et al.* 2016 *IEEE Trans. Neural Syst. Rehabil. Eng.* **24** 1109
[12] Coronel-Escamilla A, Gomez-Aguilar J F, Stamova I and Santamaria F 2020 *Chaos Solitons Fractals* **140** 110149
[13] Liu C, Zhao G, Meng Z, *et al.* 2021 *Cogn. Neurodyn.* **15** 1157
[14] Fleming J E, Dunn E and Lowery M M 2020 *Front. Neurosci.* **14** 166
[15] Shu H and Pi Y 2000 *Comput. Chem. Eng.* **24** 859
[16] Zeng G Q, Xie X Q, Chen M R and Weng J 2019 *Swarm Evol. Comput.* **44** 320
[17] Ding J, Wu M and Xiao M 2022 *IEEE Trans. Neural Netw. Learn. Syst.* **35** 8715
[18] Psaltis D, Sideris A and Yamamura A A 1988 *IEEE Control Syst. Mag.* **8** 17
[19] Yang H and Liu J 2018 *IEEECAA J. Autom. Sin.* **5** 457
[20] Liu Y J, Zeng Q, Tong S, Chen C P and Liu L 2019 *IEEE Trans. Ind. Electron.* **66** 9458
[21] Liu Y J, Zhao W, Liu L, Li D, Tong S and Chen C P 2021 *IEEE Trans. Neural Netw. Learn. Syst.* **34** 2732
[22] Khan S, Khan F, Sikander Q U A, Alam M M and Su'Ud M M 2021 *IEEE Access* **9** 136929
[23] Liu C, Zhao G, Wang J, Wu H, Li H, Fietkiewicz C and Loparo K A 2020 *IEEE Access* **8** 161067
[24] Wang K, Wang J, Zhu Y, Li H, Liu C, Fietkiewicz C and Loparo K A 2022 *Biomed. Signal Process. Control* **77** 103776
[25] Zaheer M, Reddi S, Sachan D, Kale S and Kumar S 2018 *Adv. Neural Inf. Process. Syst.* **31**
[26] Kingma D P 2014 arXiv:1412.6980
[27] Abdulkadirov R, Lyakhov P, Bergerman M and Reznikov D 2024 *Chaos Solitons Fractals* **179** 114432
[28] Chang Z, Zhang Y and Chen W 2019 *Energy* **187** 115804
[29] Böttcher L, Antulov-Fantulin N and Asikis T 2022 *Nat. Commun.* **13** 333
[30] So R Q, Kent A R and Grill W M 2012 *J. Comput. Neurosci.* **32** 499
[31] Terman D, Rubin J E, Yew A and Wilson C 2002 *J. Neurosci.* **22** 2963
[32] Rubin J E and Terman D 2004 *J. Comput. Neurosci.* **16** 211
[33] Liu C, Wang J, Li H, Lu M, Deng B, Yu H, Wei X, Fietkiewicz C and Loparo K A 2016 *IEEE Trans. Neural Netw. Learn. Syst.* **28** 371
[34] Schiff S J 2010 *Philos. Trans. R. Soc. A* **368** 2269
[35] Lu M, Wei X, Che Y, Wang J and Loparo K A 2019 *IEEE Trans. Neural Syst. Rehabil. Eng.* **28** 339
[36] Feng X J, Shea-Brown E, Greenwald B, Kosut R and Rabitz H 2007 *J. Comput. Neurosci.* **23** 265
[37] Lin N, Chi R and Huang B 2020 *Appl. Math. Model.* **81** 787
[38] Glorot X and Bengio Y 2010 *Proc. 13th Int. Conf. Artif. Intell. Stat.* (JMLR Workshop and Conference Proceedings) pp. 249–256
[39] Birdno M J and Grill W M 2008 *Neurotherapeutics* **5** 14
[40] Zhao T, Faustino B L, Jagatheesaperumal S K, Rolim F d P S and de Albuquerque V H C 2025 *Expert Syst. Appl.* **267** 126154
[41] Holgado A J N, Terry J R and Bogacz R 2010 *J. Neurosci.* **30** 12340
[42] Liu C, Wang J, Yu H, Deng B, Wei X, Li H, Loparo K A and Fietkiewicz C 2015 *Commun. Nonlinear Sci. Numer. Simul.* **28** 10
[43] Yu Y, Wang X, Wang Q and Wang Q 2020 *Appl. Math. Mech.* **41** 1747
[44] Kumaravelu K, Brocker D T and Grill W M 2016 *J. Comput. Neurosci.* **40** 207
[45] Yu Y and Wang Q 2019 *Nonlinear Dyn.* **98** 1065