CAAI Transactions on Intelligence Technology

WILEY

ORIGINAL RESEARCH OPEN ACCESS

FPGA-Based Real-Time Implementation of Closed-Loop Brain-Controlled Eye Movement System

Guodao Zhang[1,2,3] | Yupeng Li[4] | Yanjie Lu[5] | Genfu Yang[1] | Xiaotian Pan[1,2,3] | Jing Wang[6]

[1]Institute of Intelligent Media Computing, Hangzhou Dianzi University, Hangzhou, China | [2]Key Laboratory of Micro-Nano Sensing and IoT of Wenzhou, Wenzhou Institute of Hangzhou Dianzi University, Wenzhou, China | [3]Shangyu Institute of Science and Engineering Co. Ltd., Hangzhou Dianzi University, Shaoxing, China | [4]School of Automation, Hangzhou Dianzi University, Hangzhou, China | [5]School of Computer Science, Hangzhou Dianzi University, Hangzhou, China | [6]Department of Ophthalmology, Zhongshan Hospital, Fudan University, Shanghai, China

Correspondence: Jing Wang (wang.jing7@zs-hospital.sh.cn) | Xiaotian Pan (xiaotianpan@hdu.edu.cn)

Received: 12 December 2025 | **Revised:** 1 July 2026 | **Accepted:** 22 July 2026

Keywords: brain–eye interaction | closed-loop control | FPGA implementation | neuromorphic hardware | oculomotor system | real-time system

ABSTRACT

This paper presents a comprehensive closed-loop brain-eye movement framework, spanning parametric modelling, dynamic analysis and real-time FPGA implementation. A simplified first-order oculomotor model was derived from the classical Robinson formulation to preserve the essential neural-mechanical feedback structure while enabling efficient hardware realization. Through systematic parametric variations of neural and mechanical properties and multiple classes of brain-generated inputs (step, ramp, sinusoidal and noisy), the system exhibited stable closed-loop behaviour with fast convergence and low steady-state errors. Time-domain and phase-plane analyses confirmed that the eye accurately tracked diverse target trajectories across wide parameter ranges, with all equilibrium points classified as asymptotically stable. The discrete-time model was implemented on a Xilinx Virtex-4 FPGA using only adders, subtractors and shifters, completely avoiding multipliers. The design achieved a maximum clock frequency of 118 MHz, operated at 100 MHz with a closed-loop latency of nine cycles (90 ns), and consumed approximately 280 mW total power while utilising less than 4% of the device resources. To validate the system in a practical setting, a real-time hardware platform with a 3 × 3 target grid (nine fixation points) was developed. Experimental results demonstrated that the closed-loop brain-eye architecture reliably steered the eye towards each target with minimal tracking error and smooth dynamic response. Overall, this work bridges physiological oculomotor models and neuromorphic hardware, offering a scalable resource-efficient platform for real-time brain-eye closed-loop emulation and control.

1 | Introduction

Precise and low-latency control of eye movements plays a fundamental role in visual perception, neuromorphic vision and brain–machine interfaces [1–3]. The human oculomotor system integrates neural commands and biomechanical dynamics to enable accurate gaze stabilisation and target tracking under rapidly changing visual stimuli. Emulating these closed-loop brain–eye interactions in real time is essential not only for understanding sensorimotor integration but also for advancing artificial vision systems [4]. Moreover, the recent synergy between advanced artificial intelligence and brain–computer interfaces, alongside innovations in neurocomputing architectures, has opened new frontiers for real-time neuromorphic emulation and control [5].

The classical Robinson model provides a well-established second-order representation of the oculomotor system, capturing both neural feedback and ocular plant dynamics [6, 7].

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2026 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

https://doi.org/10.1049/cit2.70172

Although this formulation accurately describes saccadic and smooth pursuit behaviours, its direct digital implementation is computationally demanding due to coupled differential equations and multiple dynamic states.

To address this, simplified or reduced-order models have been proposed to retain the essential neural–mechanical feedback structure while significantly reducing implementation complexity. Such models enable systematic analysis of neural and mechanical parameters while remaining compatible with fixed-point arithmetic and real-time digital realisations on embedded hardware platforms [8].

Despite extensive research on dynamic system modelling, high-speed hardware implementations on FPGA and digital platforms have proven highly beneficial across multiple domains, including neural circuits, auditory systems, ocular motor control, and other sensorimotor systems. For example, some neuron models such as Izhikevich, Adex, Hodgkin-Huxley and Morris–Lecar have been implemented on different FPGA platforms using different mathematical approaches [9–11]. On the other hand, retinal cell modelling has been realized on FPGA board in refs. [12–14]. In particular, CORDIC-based digital implementation of the Hodgkin–Huxley neuron model is presented in ref. [15]. In addition, in refs. [16, 17] digital realization of HR model is proposed in single and coupled forms. Chen in [18] presents FPGA realization of Quartic Neuron Model (QNM) using cost-effective mathematical modifications. Moreover, memristive-based modelling of Morris–Lecar neuron is investigated in ref. [19] in case of FPGA implementation. On the other hand, in case of biological cell implementation and other organs, cardiac Purkinje fibre cells for biomedical engineering approaches are discussed in ref. [20]. In addition, artificial cochlea implementation is presented in refs. [21, 22]. These implementations enable real-time operation, low latency, and deterministic performance, which are crucial for both experimental neuroscience and embedded applications.

Building on these advancements, several studies have also explored hardware implementations or simulations of brain–eye systems, focussing on oculomotor dynamics, saccadic and smooth pursuit models, and closed-loop neural control.

A biomimetic oculomotor controller based on spiking neural networks is developed and trained for a realistic biomechanical eye model in ref. [3]. A comprehensive survey of FPGA-based visual object tracking (VOT) systems is presented in ref. [23]. In ref. [24], an RTL-based FPGA implementation of a real-time eye tracking system is presented. A real-time FPGA-based pupil tracking system is presented in ref. [25], where a vision-based black box technique is implemented on a Xilinx Spartan-6 device to achieve robust and low-power pupil localization. The proposed FPGA-based eye movement detection system efficiently identifies saccades and blinks from EOG signals using FIR filtering and Haar wavelet decomposition [26]. An FPGA-implemented CNN is proposed for accurate and portable classification of saccadic eye movements [27]. In ref. [28], a humanoid robotic implementation of a neurophysiological model of brainstem saccadic circuitry was proposed. A recent survey [29] reviewed state-of-the-art methods for eye tracking and head movement detection, highlighting their applications in human–computer interaction, assistive technologies, and driving assistance systems. In addition, in ref. [30], a physiologically motivated framework was presented for model-based separation, detection, and classification (MBSDC) of eye movements.

The core contribution of this work is the development of a complete closed-loop brain–eye tracking system that integrates physiological modelling, parametric dynamic analysis, and real-time neuromorphic implementation. We construct a fully operational closed-loop framework that emulates neural control and oculomotor response in real time. At the modelling level, a reduced-order formulation is proposed to preserve the essential neural–mechanical feedback while remaining amenable to efficient digital implementation. Extensive time-domain and phase-plane analyses across a broad range of neural and mechanical parameters and multiple input classes (step, ramp, sinusoidal and noisy) confirm stable tracking behaviour and predictable dynamic responses. This model is then realized on FPGA through a multiplier-free architecture composed exclusively of adders, subtractors and shifters, resulting in low resource utilization, sub-microsecond closed-loop latency, and modest power consumption. The hardware platform is further enhanced with a 3×3 visual target grid, enabling real-time experimental evaluation of the closed-loop system. The brain-generated commands drive the eye model to sequentially track each target, completing fixation movements with high accuracy and smooth trajectories. This integrated modelling-hardware approach represents a demonstration of a fully functional closed-loop brain-eye tracking system implemented on FPGA, bridging physiological insight with practical neuromorphic control.

It should be explicitly noted that the primary contribution of this work is not the introduction of a novel physiological or mathematical model of the oculomotor system. Instead, the core novelty lies in the realm of algorithm–hardware codesign. The specific choice of this reduced-order formulation is motivated by its structural compatibility with fixed-point discretisation, avoiding complex operations that other theoretical reduced-order approaches might still require. Unlike prior reduced-order models, our formulation is explicitly restructured to remove nonessential dynamics for direct hardware mapping. Specifically, the model avoids multiplication and division operations in its discrete implementation, relying solely on addition, subtraction, and shift operations for state updates. This hardware-aware approach bridges the gap between biological modelling and physical implementation, enabling a highly resource-efficient, high-speed, and zero-DSP FPGA realization that operates entirely on multiplier-less (shift-add) logic.

Furthermore, to distinguish the proposed framework from existing FPGA-based eye movement and neuromorphic control systems, it is crucial to highlight its methodological novelty. Conventional FPGA eye-tracking systems predominantly function as vision-based applications, utilising computationally expensive algorithms (e.g., convolutional neural networks or morphological operations) to externally observe pupil dynamics. Conversely, recent neuromorphic controllers often rely on resource-heavy spiking neural networks (SNNs) acting as computational black-boxes. In stark contrast, our framework operates as a direct real-time hardware emulator of the biological

closed-loop oculomotor mechanism. By mathematically restructuring a deterministic reduced-order physiological model into a hardware-aware zero-DSP architecture, this approach fundamentally eliminates the need for hardware multipliers (utilising pure shift-add logic). This paradigm shift bridges precise physiological modelling with digital hardware synthesis, achieving an ultra-low latency and highly resource-efficient neuromorphic emulation.

The remainder of this paper is organised as follows. Section 2 presents the oculomotor modelling framework and the associated parametric dynamic analyses. Section 3 describes the evaluation of model features. Section 4 presents the proposed FPGA architecture and its implementation details and also reports the experimental validation using the multi-target tracking platform and discusses the performance results. Finally, Section 5 concludes the paper and outlines potential future research directions.

2 | Simplified Modelling of the Oculomotor System

The human oculomotor system can be fully described by the classical Robinson model [6, 7], which represents the eye and extraocular muscles as a second-order mass–spring–damper system driven by neural feedback. To enable real-time hardware implementation while retaining the essential features of eye movement control, the system can be approximated as a simplified first-order model.

The selection of this specific simplified formulation in our work is strictly driven by the stringent constraints of digital hardware design rather than proposing a novel physiological paradigm. In order to achieve a high-speed, cost-effective closed-loop platform, the mathematical model must be explicitly tailored for hardware synthesis. Classical and contemporary studies have frequently employed reduced-order models to analyse the oculomotor system. For instance, Robinson established foundational first- and second-order linear models to represent the oculomotor plant [31], whereas Sylvestre and Cullen utilised similar simplifications to evaluate neuron discharge dynamics [32]. More recently, Radisavljevic-Gajic decoupled a second-order oculomotor model into pure-slow and pure-fast dynamic components [33], and Broucke introduced an adaptive internal model for theoretical control analysis [8]. Furthermore, when neurophysiological and biomimetic models are implemented on robotic platforms, they typically rely on computationally heavy frameworks executed on standard microprocessors or PCs. Examples include the humanoid robot vision head by Alitappeh et al. [4], the anthropomorphic head driven by leaky-integrator networks proposed by Manfredi et al. [28] and the six-degree-of-freedom cable-driven robotic eye employing nonlinear autoregressive networks (NARX) presented by Alitappeh et al. [4].

However, these existing approaches—whether reduced-order theoretical models or complex 3-D kinematics—are entirely software-based. They heavily rely on exact simulations involving continuous differential equations, floating-point arithmetic, and heavy matrix multiplications, which do not address the stringent constraints of digital hardware synthesis. Unlike these conventional methodologies, our specific novelty lies in translating a reduced-order dynamic model into a hardware-aware RTL design. Unlike conventional reduced-order models derived primarily for biological fidelity, our formulation explicitly removes nonessential dynamics and is mathematically restructured to ensure direct fixed-point realisability. Specifically, the model avoids multiplication and division operations in its discrete implementation, relying solely on addition, subtraction and shift operations for state updates. By applying these cost-effective mathematical modifications to avoid complex operations, we explicitly optimise the model for a resource-efficient zero-DSP FPGA realization utilising purely multiplier-less (shift-add) logic.

2.1 | First-Order Oculomotor Model

The continuous-time first-order model is expressed as follows:

$$\begin{cases} \dot{\theta}(t) = v(t), \\ \dot{v}(t) = -\dfrac{B}{J}v(t) - \dfrac{K}{J}\theta(t) + \dfrac{1}{J}T_m(t), \\ \dot{T}_m(t) = -\dfrac{1}{\tau_n}T_m(t) + k_p\big(\theta_{\text{target}}(t) - \theta(t)\big), \end{cases} \tag{1}$$

where:

1. $\theta(t)$ is the eye angular position [rad],
2. $v(t)$ is the eye angular velocity [rad/s],
3. $T_m(t)$ is the neural muscle torque generated to drive the eye,
4. $\theta_{\text{target}}(t)$ is the desired eye angle commanded by the brain,
5. J is the effective moment of inertia of the eyeball, reflecting resistance to acceleration,
6. B is the viscous damping coefficient, representing resistance from orbital tissues,
7. K is the elastic stiffness of the ocular plant, modelling the restoring torque,
8. k_p is the neural proportional gain, determining the strength of corrective action by the brain,
9. τ_n is the neural time constant, representing the delay in neural signal generation.

To formally justify the simplified state-space model employed in this study, the explicit mathematical assumptions used to derive this first-order formulation from the classical Robinson model must be established. The classical Robinson formulation characterises the oculomotor plant as a nonlinear, second-order system driven by parallel pulse-step neural generators. The derivation of our reduced-order model relies on two primary abstractions. First, regarding the ocular mechanics, we explicitly assume that the nonlinear force-velocity relationships of individual agonist and antagonist muscles can be lumped into a single, linearised net torque (T_m). Because the physical inertia of the eyeball is negligible relative to the high orbital viscosity and stiffness, the rapid inertial transients are heavily overdamped, allowing the plant to be approximated linearly. Second, regarding the neural drive, the discrete, parallel circuitry of the burst generator and neural integrator is abstracted into a continuous, single-path proportional error controller with a lumped neural time constant (τ_n). Consequently, the ultra-fast microscopic accelerations generated at the exact onset of saccades are simplified, effectively acting as a low-pass filter over the neural commands.

This degree of dynamic simplification is strictly acceptable and necessary for the goal of real-time digital hardware implementation. Retaining the complete nonlinear, second-order differential equations would mathematically prohibit a multiplier-less design, mandating resource-heavy floating-point ALUs, CORDIC-based nonlinear approximations and extensive DSP utilization. By applying these explicit abstractions, the model is strictly constrained to a linear format, enabling direct synthesis into a fixed-point, zero-DSP architecture (utilising only shift-and-add logic) without compromising the macroscopic stability and target-tracking accuracy required for practical neuromorphic emulation.

Although the proposed simplified formulation provides significant hardware advantages, it inherently involves a trade-off regarding physiological fidelity. By reducing the classical Robinson model, certain microscopic biological dynamics are bypassed. Specifically, the simplified model lumps the complex, nonlinear force-velocity relationships of individual agonist and antagonist extraocular muscles into a single linearised net torque. Furthermore, it omits the precise high-frequency pulse-slide-step neural burst commands, meaning the ultra-fast microscopic accelerations at the very onset of saccades are smoothed out. However, preserving these nonlinear, higher-order dynamics in digital hardware would mandate floating-point arithmetic, complex nonlinear approximations and hardware multipliers (DSPs), thereby destroying the low-latency and low-power objectives. This simplification is strictly justified because the macro-level dynamic behaviours—such as saccadic target acquisition, smooth pursuit tracking, and closed-loop stability—remain fully intact, making it highly effective for real-time neuromorphic control applications.

2.2 | System Dynamics and Brain Role

The reference input for the system is provided by the brain in the form of a desired eye angle, $\theta_{\text{target}}(t)$. The error signal, defined as $e(t) = \theta_{\text{target}}(t) - \theta(t)$, quantifies the difference between the target and the actual eye position and serves as the driving input for the neural controller.

According to the first-order dynamics of the neural controller, the error is converted into a corrective muscle torque $T_m(t)$ through the following equation:

$$\dot{T}_m(t) = -\frac{1}{\tau_n} T_m(t) + k_p\, e(t),$$

This torque represents the primary mechanism through which the brain influences the ocular plant.

As the eye moves in response to $T_m(t)$, the angular position $\theta(t)$ approaches the desired target $\theta_{\text{target}}(t)$. The error $e(t)$ consequently diminishes, causing the neural torque $T_m(t)$ to adjust dynamically in real time. This continuous interaction forms a closed-loop feedback system, in which the brain constantly monitors the position of the eye and corrects any deviation from the target, ensuring accurate and smooth tracking of the desired trajectory.

Figure 1 illustrates the simulated behaviour of the oculomotor system under three distinct brain-generated target signals: pulsed, sinusoidal, and smoothed noisy trajectories. The results highlight the closed-loop dynamics of the system, demonstrating how the brain continuously monitors the eye position and generates corrective neural commands to achieve accurate tracking. The figure illustrates three types of target signals $\theta_{\text{target}}(t)$: a pulsed input representing saccadic commands, a sinusoidal input modelling smooth pursuit, and a smoothed noisy input emulating irregular brain activity. The second subplot shows the corresponding neural torques $T_m(t)$, where large responses occur for sudden target changes, smoother variations for sinusoidal input, and fluctuating corrections for noisy input. The third subplot depicts the resulting eye positions $\theta(t)$, demonstrating that the oculomotor system accurately tracks all targets: rapid jumps for pulsed input (saccades), oscillatory tracking for sinusoidal input, and robust following of the noisy input without collapse.

Overall, these results emphasise the central role of the brain in closed-loop control of eye movements. The brain continuously computes the position error and generates muscle torques to drive the eye towards the desired target, maintaining accurate alignment under various input types.

3 | Evaluation of Model Features

Before proceeding to hardware implementation, it is essential to thoroughly examine the dynamic and functional characteristics of the simplified oculomotor model. Such an evaluation provides insight into the system's performance, robustness and response to various brain-generated commands.

To ensure rigorous biological interpretability, it is imperative to explicitly map the mathematical control variables of this framework to their physiological counterparts. In the physical oculomotor system, the "brain-generated commands" are not continuous mechanical torques; rather, they are discrete neural spike trains originating from motoneurons within the brainstem nuclei (e.g., the oculomotor and abducens nuclei). According to classical neurophysiology, this efferent neural drive operates on a "pulse-step" rate-coding mechanism. A high-frequency burst of spikes (the neural pulse) is dispatched to overcome orbital viscosity during rapid saccadic transitions, followed by a sustained steady-state firing rate (the neural step) to counteract the elastic restoring forces of extraocular muscles during target

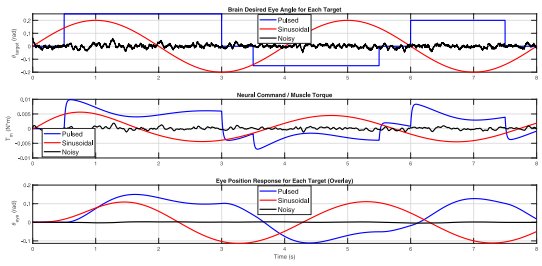


FIGURE 1 | Simulation of the oculomotor system under three different brain-generated target signals.

fixation. In the proposed hardware-aware framework, the continuous control variable (net torque) serves as a macroscopic, lumped approximation of this discrete neural drive. Conceptually, the transient error-driven effort in the feedback loop mathematically emulates the biological neural pulse, whereas the steady-state control signal mirrors the neural step. This abstraction aggregates the complex spiking activities of individual agonist and antagonist muscles into a unified deterministic variable, successfully bridging the gap between microscopic physiological reality and resource-efficient digital hardware emulation.

3.1 | Time-Domain Analysis

To further investigate the dynamic behaviour of the simplified oculomotor model, time-domain simulations [34, 35] were performed under three distinct types of brain-generated target signals: pulsed, sinusoidal, and smoothed noisy trajectories. For each target, the neural proportional gain k_p was varied across three physiologically plausible values to observe its effect on the closed-loop response.

Figure 2 presents the resulting simulations. Each row corresponds to a different type of target input: the top row illustrates the pulsed target, the middle row the sinusoidal target and the bottom row the smoothed noisy input. Within each row, three columns depict (i) the target signal and the corresponding eye position $\theta(t)$, (ii) the instantaneous position error $e(t) = \theta_{\text{target}}(t) - \theta(t)$ and (iii) the neural command or muscle torque $T_m(t)$ generated by the brain.

The results clearly demonstrate the closed-loop nature of the brain–eye system. For all target types, the brain continuously monitors the error between the desired and actual eye positions, generating corrective neural torques that drive the ocular plant towards the target. Increasing k_p results in faster correction and reduced steady-state error, as evidenced by the smaller deviations in $\theta(t)$ and $e(t)$. The pulsed target row shows rapid saccade-like transitions, with neural torque spikes corresponding to sudden changes in the target angle. The sinusoidal input produces smooth oscillatory eye movements, and the torque adapts continuously to follow the slow-varying reference. For the noisy input, the system maintains robust tracking despite irregular fluctuations, highlighting the resilience of the closed-loop control even under unpredictable brain signals.

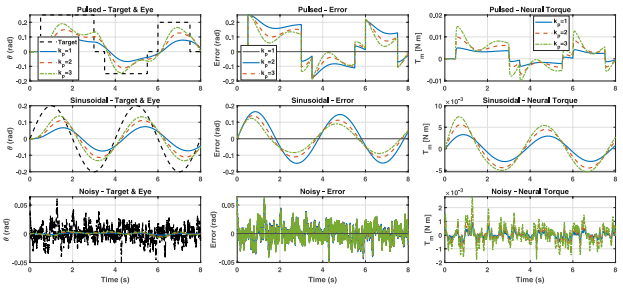


FIGURE 2 | Time-domain simulations of the closed-loop oculomotor system for three types of brain-generated target signals (pulsed, sinusoidal, smoothed noisy) and three different neural proportional gains k_p.

Also, Table 1 summarises the system response metrics for systematic variations of the eyeball inertia J, viscous damping B, elastic stiffness K, neural time constant τ_n and neural proportional gain k_p. As shown in the table, increasing the inertia J results in slower eye acceleration and larger final displacement θ_{final}, reflecting the increased resistance to angular motion. Higher viscous damping B reduces peak eye angles $\theta_{\max}$ and smooths oscillations, indicating that damping mitigates overshoot while slightly slowing the response. The elastic stiffness K exerts a restoring torque proportional to displacement, such that increasing K decreases both θ_{final} and θ_{mean}, while modestly increasing the average neural torque $T_{m,\text{mean}}$ required to achieve the target. The neural time constant τ_n controls the speed of corrective torque generation: larger τ_n delays neural response, leading to increased transient error and higher mean deviation from the target, as reflected in $\text{error}_{\text{mean}}$. Finally, the proportional gain k_p scales the neural corrective action; higher k_p amplifies the torque $T_m(t)$, producing faster error reduction and higher peak eye velocities, yet may introduce overshoot if excessively large.

It is essential to establish the physiological and engineering rationale behind the selected ranges for the system parameters (J, B, K, k_p, τ_n). The parameter selection in this study is governed by algorithm–hardware codesign methodology. For the mechanical parameters of the ocular plant (inertia J, viscous damping B and elastic stiffness K), direct utilization of absolute physical biomechanical values in a discrete fixed-point architecture is mathematically impractical and susceptible to numerical instability. Therefore, these parameters are mapped into a dynamically equivalent scaled domain. This normalisation ensures fixed-point arithmetic stability—crucial for maintaining the zero-DSP nature of the architecture—while meticulously preserving the relative proportions that govern the fundamental transient and steady-state dynamic characteristics of the biological oculomotor plant [6, 31]. Conversely, the selected ranges for the neural controller parameters are directly grounded in physiological measurements. The neural time constant range, $\tau_n \in [0.01, 0.04]$ s, accurately models the 10–40 ms neural transmission, synaptic processing, and muscle activation delays experimentally verified in primate and human oculomotor systems [7, 32]. Additionally, the variations in the proportional gain (k_p) represent the physiologically plausible modulations in neural command strength required to effectively overcome orbital viscosity during saccadic and smooth pursuit movements. Thus, the evaluated parameter ranges achieve a rigorous balance between biological dynamic fidelity and real-time digital hardware viability.

Specifically, for the baseline dynamic analysis, the fixed nominal values of $J = 0.01\ \text{kg·m}^2$, $B = 0.02\ \text{N·m·s/rad}$ and $K = 0.05\ \text{N·m/rad}$ are employed. Although mathematically scaled to prevent numerical underflow in the discrete fixed-point datapath, these explicit values are meticulously tuned so that their relative dynamic ratios (i.e., B/J and K/J) strictly emulate the viscoelastic restoring forces and overdamped transient behaviour of the physiological oculomotor plant.

TABLE 1 | Effect of parameter variations on the closed-loop oculomotor system (time-domain metrics).

Parameter	Value	θ_{final}	θ_{mean}	θ_{max}	θ_{min}	$T_{m,mean}$	$Error_{mean}$
J	0.008	0.012898	0.036281	0.14461	−0.10349	0.001675	0.10124
J	0.010	0.017762	0.036886	0.14972	−0.11124	0.001652	0.10065
J	0.012	0.026227	0.037438	0.15383	−0.11590	0.001631	0.09966
J	0.014	0.035608	0.037895	0.15724	−0.11916	0.001613	0.09846
J	0.016	0.044765	0.038298	0.16013	−0.12256	0.001598	0.09729
B	0.015	0.003524	0.037577	0.16299	−0.12734	0.001623	0.09730
B	0.020	0.017762	0.036886	0.14972	−0.11124	0.001652	0.10065
B	0.025	0.028008	0.036191	0.13906	−0.098996	0.001681	0.10432
B	0.030	0.035440	0.035487	0.13052	−0.089303	0.001709	0.10806
B	0.035	0.040782	0.034789	0.12376	−0.081574	0.001738	0.11171
K	0.040	0.031135	0.041474	0.16540	−0.12212	0.001470	0.09554
K	0.050	0.017762	0.036886	0.14972	−0.11124	0.001652	0.10065
K	0.060	0.008722	0.033118	0.13690	−0.10230	0.001802	0.10506
K	0.070	0.003127	0.029978	0.12621	−0.094581	0.001927	0.10883
K	0.080	0.000236	0.027343	0.11714	−0.087632	0.002032	0.11203
τ_n	0.015	0.023137	0.031067	0.12335	−0.090827	0.001413	0.10994
τ_n	0.020	0.017762	0.036886	0.14972	−0.11124	0.001652	0.10065
τ_n	0.025	0.010699	0.041444	0.17230	−0.12936	0.001837	0.09309
τ_n	0.030	0.003762	0.045047	0.19213	−0.14567	0.001987	0.08681
τ_n	0.035	−0.001355	0.047923	0.20992	−0.16034	0.002116	0.08202
k_p	1	0.025633	0.023524	0.091911	−0.067372	0.001093	0.12178
k_p	2	0.017762	0.036886	0.14972	−0.11124	0.001652	0.10065
k_p	3	0.003441	0.045001	0.19048	−0.14341	0.001989	0.08643
k_p	4	−0.003478	0.050255	0.22129	−0.16592	0.002230	0.07698
k_p	5	−0.002144	0.054128	0.24570	−0.17889	0.002400	0.07028

3.2 | Dynamic Analysis of the Closed-Loop Brain–Eye System

Analysing the dynamics of the oculomotor system provides insight into how the brain controls eye movements beyond specific time-domain responses [36].

The dynamic behaviour of the closed-loop brain–eye system can be understood by examining the nullclines and phase-plane trajectories in the (θ, v) and (θ, T_m) spaces. Nullclines describe the loci of points where the time derivative of a state variable is zero, thereby separating regions of positive and negative flow and revealing the location and type of equilibrium points.

3.2.1 | *Velocity Nullcline*

Setting $\dot{v} = 0$ in the eye plant dynamics

$$\dot{v} = -\frac{B}{J}v - \frac{K}{J}\theta + \frac{1}{J}T_m, \tag{2}$$

yields the linear relation

$$T_m = K\theta + Bv, \tag{3}$$

which corresponds to all (θ, v) pairs where the eye acceleration vanishes for a given instantaneous neural torque T_m.

3.2.2 | *Neural Torque Nullcline*

Setting $\dot{T}_m = 0$ in the neural controller dynamics

$$\dot{T}_m = -\frac{1}{\tau_n}T_m + k_p(\theta_{target} - \theta), \tag{4}$$

gives

$$T_m = \tau_n k_p(\theta_{target} - \theta), \tag{5}$$

which is a straight line in the (θ, T_m) plane with slope $-\tau_n k_p$ and intercept $\tau_n k_p \theta_{target}$. Increasing k_p or τ_n steepens this nullcline, shifting its intersection with the plant nullcline and thereby modifying both steady-state error and transient dynamics.

3.2.3 | *Equilibrium Point*

The equilibrium (θ^*, v^*, T_m^*) is defined by the simultaneous solution of

$$\dot{\theta} = v = 0, \tag{6}$$

$$\dot{v} = 0, \tag{7}$$

$$\dot{T}_m = 0, \tag{8}$$

leading to

$$v^* = 0, \qquad K\theta^* = \tau_n k_p(\theta_{\text{target}} - \theta^*). \tag{9}$$

Solving for θ^* gives the unique closed-loop equilibrium

$$\theta^* = \frac{\tau_n k_p}{K + \tau_n k_p}\,\theta_{\text{target}}, \qquad T_m^* = \frac{K\tau_n k_p}{K + \tau_n k_p}\,\theta_{\text{target}}. \tag{10}$$

This result shows that the equilibrium eye position is a fraction of the target, determined by the balance between neural control strength and ocular stiffness.

3.2.4 | *Nullclines in the (θ, v) Plane*

Substituting the neural torque nullcline into the $\dot{v} = 0$ relation eliminates T_m, yielding a single v-nullcline in (θ, v):

$$-\frac{K}{J}\theta - \frac{B}{J}v + \frac{1}{J}\tau_n k_p(\theta_{\text{target}} - \theta) = 0, \tag{11}$$

which simplifies to

$$v(\theta) = \frac{\tau_n k_p}{B}\,\theta_{\text{target}} - \frac{K + \tau_n k_p}{B}\,\theta. \tag{12}$$

This line has slope $-(K + \tau_n k_p)/B$ and intercept $(\tau_n k_p/B)\theta_{\text{target}}$, clearly illustrating how both neural and mechanical parameters shape the phase-plane geometry. The θ-nullcline is given by $v = 0$.

To gain deeper insight into the intrinsic dynamic behaviour of the closed-loop brain–eye system, phase-plane and nullcline analyses are performed for different neural controller parameters. Figure 3 illustrates the θ–v and θ–T_m phase planes of the oculomotor system, showing the θ-, v-, and T_m-nullclines, the equilibrium point, and representative system trajectories starting from different initial conditions. The parameters of the ocular plant are fixed at $J = 0.01$ kg·m^2, $B = 0.02$ N·m·s/rad and $K = 0.05$ N·m/rad, whereas the neural gain k_p and time constant τ_n are varied across the subplots. A constant target angle $\theta_{\text{target}} = 0.1$ rad is used to isolate and examine the system's intrinsic closed-loop dynamics. This analysis provides an intuitive graphical representation of how neural and mechanical parameters shape the nullclines and determine the system's trajectories and equilibrium. It highlights how variations in k_p and τ_n alter the location of the equilibrium point and the transient convergence paths, thereby offering valuable insight into the mechanisms underlying stable and accurate oculomotor control.

The linearised dynamics around the equilibrium are obtained by evaluating the Jacobian matrix

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -\dfrac{K}{J} & -\dfrac{B}{J} & \dfrac{1}{J} \\ -k_p & 0 & -\dfrac{1}{\tau_n} \end{bmatrix}. \tag{13}$$

The eigenvalues λ_i of A determine local stability: if all $\text{Re}(\lambda_i) < 0$, the equilibrium is asymptotically stable. Complex eigenvalues with negative real parts correspond to stable spirals (focus), whereas real negative eigenvalues indicate stable nodes. Conversely, eigenvalues with positive real parts indicate instability, and mixed-sign real parts correspond to saddle points.

To further quantify the dynamic characteristics, we compute the following metrics from the eigenvalues:

1. Maximum real part of eigenvalues: $\text{ReMax} = \max_i \text{Re}(\lambda_i)$, indicating the slowest decay or fastest growth rate.
2. Damping ratio: For a pair of complex conjugate eigenvalues $\lambda = \sigma \pm j\omega$, $\zeta = -\sigma/\sqrt{\sigma^2 + \omega^2}$ measures the exponential decay relative to oscillation amplitude.

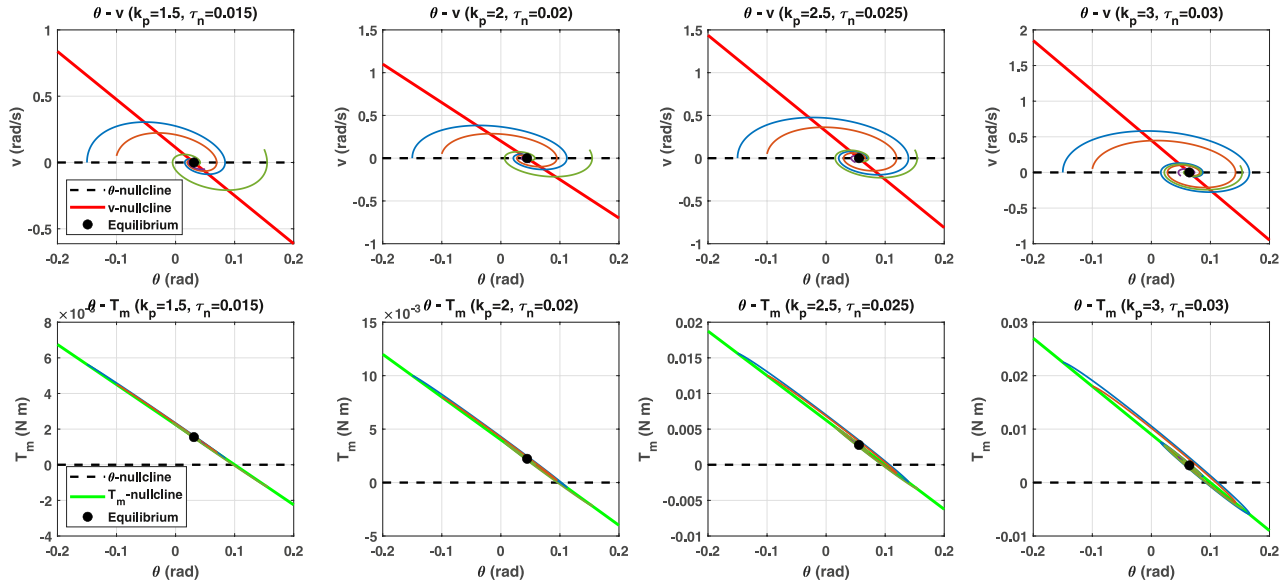


FIGURE 3 | Phase-plane analysis of the closed-loop brain–eye system for different neural controller parameters.

3. Natural frequency: $\Omega_n = \sqrt{\sigma^2 + \omega^2}$, describing the frequency of oscillatory response.

These dynamic quantities (ReMax, ζ, and Ω_n) provide direct quantitative insight into the convergence speed, overshoot, and oscillatory behaviour of the eye position near the equilibrium point.

To illustrate the effect of neural parameters on system equilibria and dynamics, we varied k_p, τ_n, and θ_{target} over physiologically plausible ranges. The resulting equilibrium points (θ^*, v^*, T_m^*) along with the dynamic indices ReMax, ζ, and Ω_n are summarised in Table 2. This parametric study reveals that increasing k_p or decreasing τ_n generally reduces steady-state error and accelerates convergence, while the natural frequency Ω_n and damping ratio ζ determine transient oscillatory behaviour.

Therefore, equilibrium and linear stability analyses are essential to quantify how variations in neural gains and time constants affect convergence speed, overshoot and oscillatory behaviour, ensuring that the system maintains precise and stable gaze control under different physiological conditions.

3.3 | Tracking Performance Analysis

To complement the time-domain and dynamical analyses, the oculomotor system is further evaluated under dynamic and physiologically realistic target signals. Unlike idealised inputs, real eye movements require continuous adjustment to moving targets with varying velocities and irregular trajectories.

In this case, we analysed the dynamic response of the eye tracking system to different types of target inputs. The model parameters were set as follows: eye inertia $J = 0.01$ kg·m^2, damping $B = 0.02$ N·m·s/rad, stiffness $K = 0.05$ N·m/rad, proportional gain $k_p = 1.5$, and neural time constant $\tau_n = 0.02$ s. The simulations were performed with a time step of $dt = 0.001$ s over a total duration of $T_{\text{sim}} = 2$ s. Multiple target angles $\theta_{\text{target}} = [0.05, 0.1, 0.15, 0.2]$ rad were considered.

The outputs of the simulations included the eye angle $\theta(t)$, eye velocity $\dot{\theta}(t)$ and muscle torque $T_m(t)$ over time. For quantitative analysis, key performance metrics were computed, including overshoot, settling time (within 2% of target), peak velocity, RMS tracking error, and torque range. The RMS error is defined as follows:

$$\text{RMS Error} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\theta(k) - \theta_{\text{target}}(k)\right)^2},$$

which represents the deviation between the reference target angle and the actual eye angle during tracking.

Figure 4 shows the simulation results for the first two target angles. Each row corresponds to a different target, and each column corresponds to one of the input types. The dashed black lines indicate the reference input $\theta_{\text{target}}(t)$, whereas the blue lines show the actual eye response $\theta(t)$.

The complete set of performance metrics for all targets and input types is summarised in Table 3. This table provides a quantitative assessment of the eye tracking accuracy and responsiveness for different reference signals.

To further evaluate the dynamic behaviour of the oculomotor system under various reference inputs, we computed a *performance index* (PI) that quantifies closed-loop tracking quality over time. The PI combines instantaneous tracking error and control effort, and is defined as follows:

TABLE 2 | Equilibria and dynamic characteristics of the closed-loop oculomotor system for various neural parameters.

k_p	τ_n	θ_{target}	θ_{eq}	v_{eq}	$T_{m,\text{eq}}$	ReMax	Damping ratio	Ω_n
0.5	0.01	0.05	0.0045455	0	0.00022727	−0.99745	0.42533	34.8985
0.5	0.01	0.10	0.0090909	0	0.00045455	−0.99745	0.42533	34.8985
0.5	0.02	0.05	0.0083333	0	0.00041667	−0.98961	0.40409	18.3062
0.5	0.02	0.10	0.016667	0	0.00083333	−0.98961	0.40409	18.3062
0.5	0.04	0.05	0.014286	0	0.00071429	−0.95720	0.36241	10.1227
0.5	0.04	0.10	0.028571	0	0.0014286	−0.95720	0.36241	10.1227
1.5	0.01	0.05	0.011538	0	0.00057692	−0.99235	0.38926	35.0380
1.5	0.01	0.10	0.023077	0	0.0011538	−0.99235	0.38926	35.0380
1.5	0.02	0.05	0.018750	0	0.00093750	−0.96889	0.34277	18.5719
1.5	0.02	0.10	0.037500	0	0.0018750	−0.96889	0.34277	18.5719
1.5	0.04	0.05	0.027273	0	0.0013636	−0.87336	0.26466	10.6177
1.5	0.04	0.10	0.054545	0	0.0027273	−0.87336	0.26466	10.6177
3.0	0.01	0.05	0.018750	0	0.00093750	−0.98471	0.34820	35.2289
3.0	0.01	0.10	0.037500	0	0.0018750	−0.98471	0.34820	35.2289
3.0	0.02	0.05	0.027273	0	0.0013636	−0.93794	0.28315	18.9164
3.0	0.02	0.10	0.054545	0	0.0027273	−0.93794	0.28315	18.9164
3.0	0.04	0.05	0.035294	0	0.0017647	−0.75169	0.18411	11.2207
3.0	0.04	0.10	0.070588	0	0.0035294	−0.75169	0.18411	11.2207

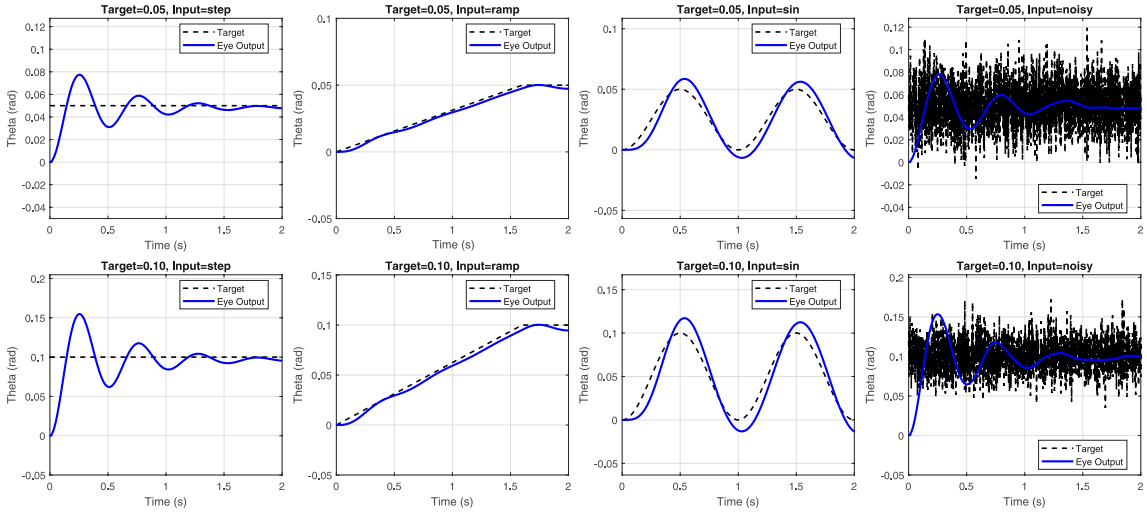


FIGURE 4 | Eye tracking simulation results for the first two target angles with step, ramp, sinusoidal, and noisy inputs.

TABLE 3 | Eye tracking performance metrics for different target angles and input types.

Target	Input type	Overshoot	Settling time (s)	Peak velocity	RMS error	Torque range
0.05	Step	0.02739	0.142	0.47931	0.012866	0.11675
0.05	Ramp	0.00015903	1.653	0.048369	0.0017589	0.0045673
0.05	Sin	0.0085704	0.429	0.21658	0.0065848	0.028354
0.05	Noisy	0.027841	0.145	0.46683	0.023528	0.24091
0.1	Step	0.054781	0.142	0.95862	0.025733	0.2335
0.1	Ramp	0.00031806	1.653	0.096738	0.0035179	0.0091345
0.1	Sin	0.017141	0.429	0.43316	0.01317	0.056709
0.1	Noisy	0.053351	0.14	0.96795	0.032545	0.3758
0.15	Step	0.082171	0.142	1.4379	0.038599	0.35025
0.15	Ramp	0.00047708	1.653	0.14511	0.0052768	0.013702
0.15	Sin	0.025711	0.429	0.64974	0.019754	0.085063
0.15	Noisy	0.083027	0.141	1.4861	0.043933	0.46605
0.2	Step	0.10956	0.142	1.9172	0.051465	0.467
0.2	Ramp	0.00063611	1.653	0.19348	0.0070358	0.018269
0.2	Sin	0.034281	0.429	0.86631	0.026339	0.11342
0.2	Noisy	0.10853	0.142	1.9171	0.054741	0.54953

$$\mathrm{PI}(t) = |\theta_{\mathrm{target}}(t) - \theta(t)| + \alpha |T_m(t)|, \tag{14}$$

where $\theta_{\mathrm{target}}(t)$ is the target angle, $\theta(t)$ is the eye output, $T_m(t)$ is the muscle torque, and α is a weighting coefficient set to 0.05 to balance error and control effort. Lower PI values correspond to better tracking performance, reflecting both accuracy and smoothness of the eye movement.

For each input type (step, ramp, sinusoidal and noisy), the system was simulated over a set of target amplitudes ranging from 0.05 to 0.2 rad with fine resolution. The simulations were performed using Euler integration with a time step of 0.001 s over a total duration of 4 s. Eye inertia, damping, stiffness, and neural gain were set to values representative of the oculomotor plant and neural controller.

The resulting PI was visualised as a heatmap for each input type (Figure 5). In these heatmaps, the horizontal axis corresponds to simulation time, the vertical axis to the target amplitude, and the colour represents the instantaneous PI value. Cooler colours indicate lower PI (better tracking), whereas warmer colours indicate higher PI (poorer performance). This visualisation allows simultaneous assessment of tracking accuracy across both time and target amplitude, revealing the interaction between input type, target magnitude and closed-loop dynamics.

The heatmaps demonstrate that the system maintains low PI values for small to moderate target amplitudes across all input types, indicating effective closed-loop control. Higher amplitudes and rapidly varying or noisy inputs increase the PI locally,

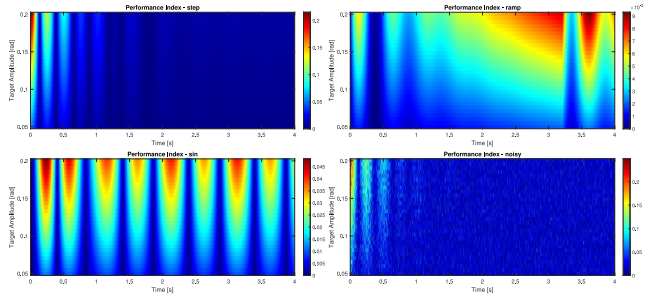


FIGURE 5 | Performance index (PI) heatmaps for closed-loop eye tracking. Each subplot corresponds to one input type: Step, ramp, sinusoidal, and noisy.

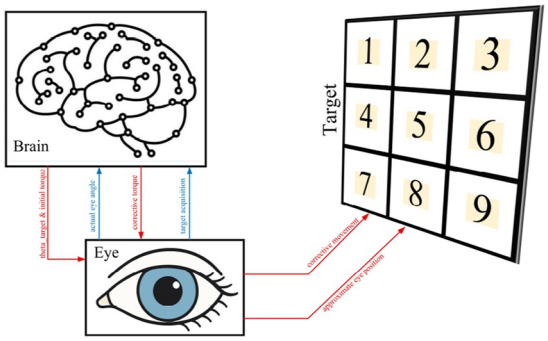


FIGURE 6 | Closed-loop brain-eye-target system. The brain issues a target angle θ_{target} and an initial torque T_{brain} to the eye. The eye rotates towards the desired target among the 3 × 3 grid.

highlighting transient performance limitations of the neural controller and plant dynamics.

4 | Hardware-Based Realistic Emulation of Eye–Brain Closed-Loop Dynamics

This section introduces a hardware framework that physically replicates the closed-loop behaviour of the human eye–brain system in real operational scenarios. Using a high-speed processing unit, the platform continuously translates desired visual target positions into ocular torques, driving a realistic eye model towards the target in real time. Each target within a defined visual field is represented as a moving point, and the system instantaneously adjusts eye orientation based on neural-inspired control laws. This setup allows immediate observation of how the eye responds to dynamic and irregular target motions, demonstrating both the transient and steady-state performance of the closed-loop system.

Figure 6 illustrates the proposed closed-loop brain-eye targeting system, composed of three main blocks: the brain, the eye, and the target screen. The screen contains a 3 × 3 grid of targets, providing nine potential positions for fixation. The system operates in a closed-loop manner, ensuring that the eye aligns with the intended target through continuous error correction.

Initially, the brain issues a command specifying the target angle θ_{target} that the eye should achieve. Simultaneously, the brain generates a torque command T_{init} to initiate eye movement towards the target. The eye receives this initial command and rotates accordingly, attempting to reach the designated target. Due to inherent delays and mechanical limitations, the eye may not precisely reach the target on the first attempt, resulting in a residual angular error.

The target screen acts as a reference, allowing the brain to detect any discrepancy between the desired target angle and the actual eye position. This detected error $\Delta\theta(t)$ is defined as follows:

$$\Delta\theta(t) = \theta_{target}(t) - \theta(t), \tag{15}$$

To correct this error, the brain computes a corrective torque $T_{\text{corr}}(t)$ proportional to the angular error:

$$T_{corr}(t) = k_p \Delta\theta(t) \tag{16}$$

where k_p denotes the neural proportional gain. The total torque applied to the eye is thus expressed as follows:

$$T_{\text{total}}(t) = T_{\text{init}}(t) + T_{\text{corr}}(t), \tag{17}$$

driving the eye towards the intended target. The eye's angular position is subsequently updated according to the system dynamics:

$$\theta(t + \Delta t) = \theta(t) + f(T_{\text{total}}(t)), \tag{18}$$

where $f(\cdot)$ models the eye's rotational response including any temporal delay and physical constraints. Note that $T_{total}(t)$ serves as the physical hardware realization of the macroscopic biological neural torque $T_m(t)$ defined in the system dynamics (Equation 1).

Through this feedback loop, the system continuously refines the eye's position, ensuring accurate alignment with the desired target. The brain-eye-target interaction, as illustrated in Figure 6, provides a clear representation of the information flow: the brain issues both the target angle and initial torque to the eye, the eye responds by moving towards the target, and the target screen allows the brain to detect residual errors and generate corrective torques. This closed-loop architecture highlights the practical implementation of real-time error correction, where the brain constantly monitors eye position relative to the target, adjusts torque commands accordingly, and maintains accurate tracking of multiple targets in the 3 × 3 grid.

It should be noted that the angular velocity v of the eye is implicitly considered through the system dynamics. Although the control strategy is mainly based on generating appropriate torque commands, these torques directly affect the eye's angular acceleration, which subsequently modifies the velocity and ultimately drives the eye angle $\theta(t)$ towards the desired target provided by the brain subsystem.

4.1 | Hardware Architecture of the Brain-Eye Controller

The continuous-time first-order oculomotor model is discretised using the Euler method with a sampling time Δt. The resulting discrete-time equations are given below:

$$\theta[n+1] = \theta[n] + \Delta t\, v[n], \tag{19}$$

$$v[n+1] = v[n] + \Delta t\left(-\frac{B}{J}v[n] - \frac{K}{J}\theta[n] + \frac{1}{J}T_m[n]\right), \tag{20}$$

$$T_m[n+1] = T_m[n] + \Delta t\left(-\frac{1}{\tau_n}T_m[n] + k_p\left(\theta_{\text{target}}[n] - \theta[n]\right)\right). \tag{21}$$

The hardware implementation of the closed-loop brain-eye system is illustrated in Figure 7. The architecture is designed to achieve high-speed performance without the use of multipliers, utilising only adders, subtractors and shifters. It consists of four main modules: the brain module, eye module, delta calculator and the 3 × 3 target grid. Each module is responsible for specific computations and communicates with other modules to maintain real-time closed-loop control.

1. Brain module: This module generates both the initial control commands and corrective actions. It receives the angular error, defined as $\Delta\theta = \theta_{\text{target}} - \theta$, from the delta calculator. Based on this error, it outputs the target angle θ_{target}, initial torque T_{init}, and corrective torque T_{corr} to the eye module. The corrective torque is computed as follows:

$$T_{\text{corr}} = k_p\Delta\theta,$$

where k_p is the proportional gain. The brain module continuously updates T_{corr} until the eye angle converges to the target.

2. Eye module: The eye module implements the oculomotor dynamics in hardware. It receives θ_{target}, T_{init}, and T_{corr} from the brain module, and updates the eye angle $\theta(t)$ according to the discretised first-order dynamics:

$$\begin{aligned}\theta[n+1] &= \theta[n] + v[n]\Delta t,\\ v[n+1] &= v[n] + \frac{\Delta t}{J}(-Bv[n] - K\theta[n] + T_{\text{total}}[n]).\end{aligned} \tag{22}$$

where v is the angular velocity and $T_{\text{total}} = T_{\text{init}} + T_{\text{corr}}$. The eye module outputs θ to the delta calculator, simulating eye movement in real time.

3. Delta calculator: This module computes the angular error between the current eye angle and the target angle. Inputs are θ_{target} from the brain module and θ from the eye module. The output, $\Delta\theta$, is sent back to the brain module. In hardware, this is efficiently implemented using a subtractor.

4. 3 × 3 target grid: This module represents nine permissible target positions and provides the selected target to the Brain Module. It does not perform complex computations; it simply signals θ_{target} to the brain module for closed-loop control.

The closed-loop operation proceeds as follows: the brain module sends θ_{target} and T_{init} to the eye module. The Eye Module updates θ based on the applied torque T_{total}. The delta calculator computes $\Delta\theta$ and returns it to the brain module, which calculates T_{corr} and sends it back to the eye module. This loop continues until θ converges to θ_{target} within a desired tolerance.

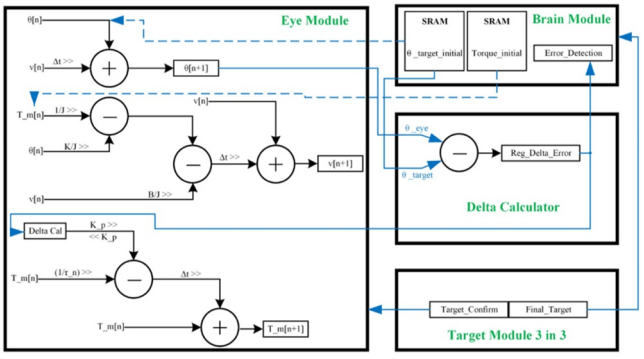


FIGURE 7 | Hardware implementation of the closed-loop brain-eye system.

4.2 | Digital Realization Results of Proposed Controller

For the hardware prototyping, we selected the Xilinx Virtex-4 XC4VLX25 FPGA device, which provides sufficient logic and memory resources for real-time operation. Table 4 reports the resource utilization of the proposed architecture on the Virtex-4 device. The results clearly demonstrate the hardware efficiency of the proposed implementation. This resource efficiency is particularly advantageous for embedded neuromorphic control systems, where real-time performance and low hardware footprint are both critical.

Because the architecture relies on continuous bidirectional interaction between the brain and eye modules (Figure 7), strict scheduling and clocking are essential: any misalignment between neural torque generation and ocular feedback updates may introduce numerical or timing artefacts. For the chosen device (Xilinx Virtex-4 XC4VLX25) the maximum clock frequency was measured at 118 MHz; the design is therefore operated nominally at 100 MHz, which provides a comfortable margin relative to timing closure while preserving low-power operation. The system scheduling is fully synchronous in a single global clock domain; the brain controller is implemented with two pipeline stages, the eye dynamics with three stages, and the feedback and target grid with one stage each (as in Table 5). The overall closed-loop path comprises seven pipeline stages and a total latency of nine clock cycles.

Using an initiation interval equal to the closed-loop latency (nine cycles) yields an internal maximum update capability of approximately 100 MHz/9 ≈ 11.11 Mupdates/s. Table 6 reports the main timing and throughput figures. Moreover, Table 7 reports power consumption of the implementation.

To contextualise the reported hardware performance, it is crucial to compare the system's internal update rate with the

TABLE 4 | FPGA resource utilization for the closed-loop brain–eye system on Virtex-4 XC4VLX25.

Resource	Used	Available	Utilization (%)
Slices	412	10,752	3.8
LUTs	765	21,504	3.6
Flip–flops	589	21,504	2.7
Block RAM (kbits)	64	1152	5.6
DSP48 multipliers	0	48	0.0

TABLE 5 | Clocking and scheduling structure of the closed-loop brain–eye system.

Module	Clock domain (MHz)	Pipeline stages	Latency (cycles)
Brain controller	100	2	3
Eye dynamics	100	3	5
Delta calculator	100	1	1
Target grid	100	1	1
Closed-loop total	100	7	9

TABLE 6 | Performance metrics of the implemented hardware.

Metric	Value	Notes
Maximum frequency	118 MHz	Synthesised
Operating frequency	100 MHz	Nominal (synthesis target)
Closed-loop latency	90 ns	9 cycles at 100 MHz
Internal max throughput	11.11 Mupdates/s	1 update every 9 cycles

TABLE 7 | Power consumption of the implementation.

Component	Power (mW)	Percentage (%)
Static power	160	57.1
Eye dynamics module	72	25.7
Brain controller	36	12.9
Delta calculator + grid	12	4.3
Total power	280	100

required operational bandwidth for natural biological eye movements. Fast human eye movements, such as saccades, typically reach peak rotational velocities of up to 700°/s. To effectively emulate and control these kinematics in real-time without introducing artificial delays, a closed-loop update rate in the kilohertz range (e.g., 1 kHz or 1 ms latency) is generally sufficient. Our proposed zero-DSP hardware architecture achieves an internal maximum throughput of 11.11 Mupdates/s and a closed-loop latency of just 90 ns. This indicates that the FPGA controller operates more than four orders of magnitude faster than the stringent requirements of real-time biological tracking. This substantial operational headroom not only guarantees lag-free, real-time closed-loop emulation of instantaneous target shifts but also demonstrates the architecture's scalability. The surplus bandwidth enables potential future integration with large-scale spiking neural networks or simultaneous multi-eye tracking systems on a single chip without bottlenecking the dynamic oculomotor response.

4.3 | Experimental Validation of the Eye-Tracking Closed-Loop System

The closed-loop eye-tracking system was tested by sequentially applying nine step target angles, denoted as θ_{target}. Each target was associated with an initial torque, T_{init}, applied to the eye module. The system tracks the instantaneous eye angle, θ, and computes the instantaneous tracking error. For each target, the system records the *reach time*, t_{reach}, defined as the first instance when the instantaneous error satisfies

$$|e(t)| < \text{tol}, \tag{23}$$

where tol is a predefined convergence threshold. In addition, the mean absolute error (MAE) over the full observation period is computed as follows:

$$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N}|e(n)|, \tag{24}$$

where N is the total number of time steps. These metrics provide quantitative measures of both the speed and accuracy of the eye-tracking response.

Figure 8 presents the eye-tracking performance for all nine targets. Each subplot shows the target angle, θ_{target}, the measured eye angle, θ, and the instantaneous error, $e(t)$. The top-right corner of each subplot indicates t_{reach} and MAE, summarising the system's ability to reach each target accurately and maintain minimal error throughout the observation period.

Furthermore, as velocity smoothness is a critical metric in biological motor control, the angular velocity profiles for each target transition have been explicitly extracted and superimposed on the tracking responses in Figure 8 (plotted on the secondary y-axis). As observed, the system successfully generates smooth, bell-shaped acceleration and deceleration velocity phases characteristic of natural saccades. The complete absence of erratic velocity spikes or computational jitter visually confirms that the proposed discrete-time fixed-point approximation—despite lacking hardware multipliers or floating-point precision—maintains absolute numerical stability and perfectly satisfies the rigorous dynamic smoothness requirements for real-time oculomotor emulation.

The closed-loop brain-eye system effectively demonstrates real-time tracking of sequential visual targets. For each target, the brain module issues a desired eye angle along with an initial torque command. The eye responds dynamically according to its inherent inertia, damping, and stiffness, whereas the continuous feedback of the angular error allows the brain to generate corrective torques. This continuous error correction ensures that the eye rapidly converges to the intended target and maintains precise alignment, despite mechanical and temporal limitations. The resulting trajectories, as shown in the experimental plots,

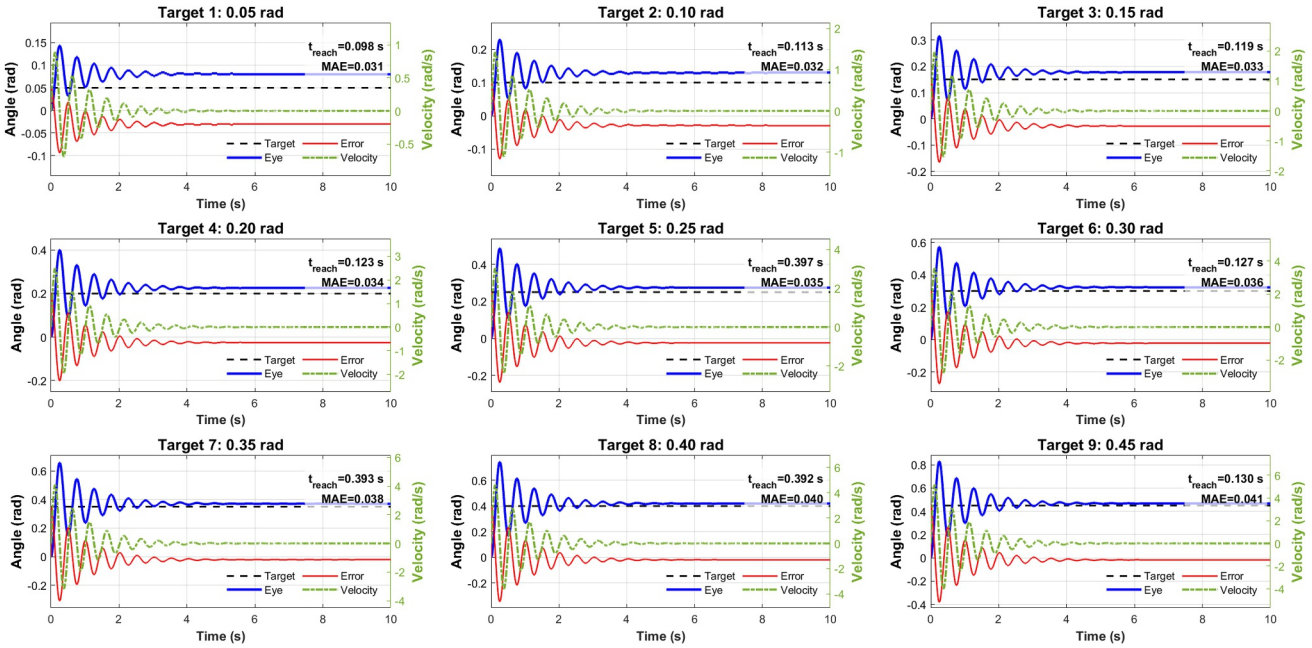


FIGURE 8 | Digital outputs for eye-tracking performance for nine step targets. Each subplot displays the target angle, measured eye angle, and instantaneous tracking error.

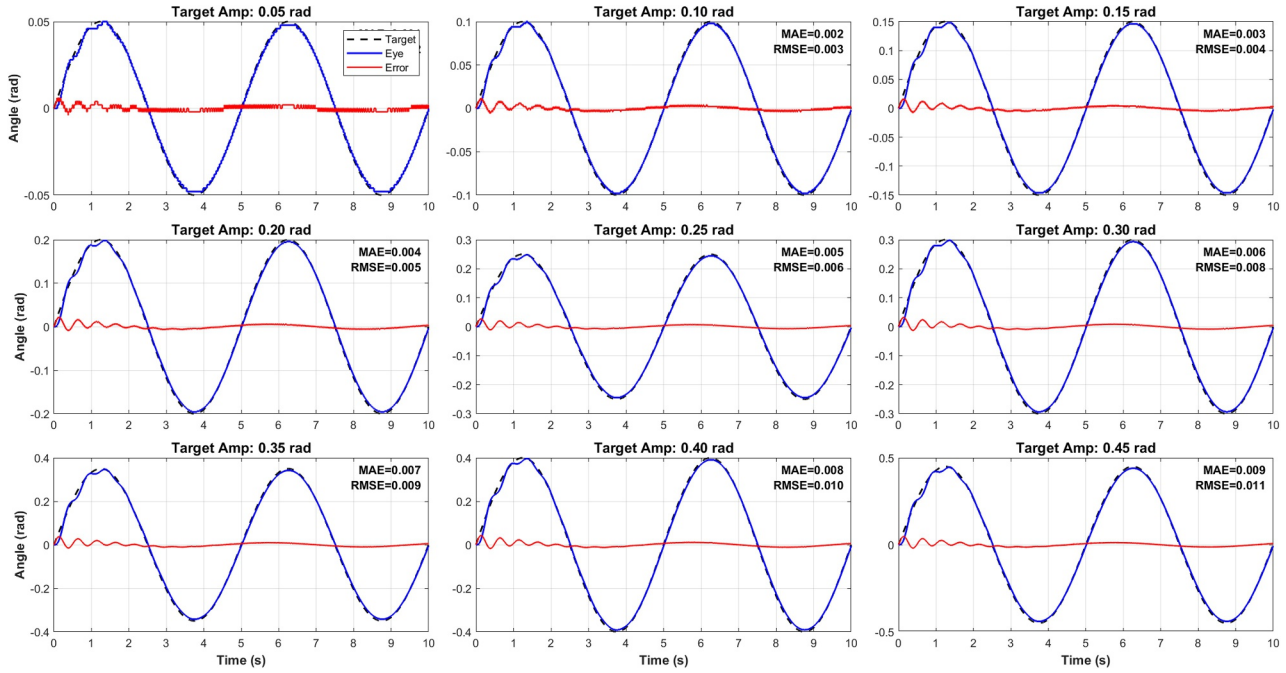


FIGURE 9 | Digital outputs representing the closed-loop tracking performance of the hardware-implemented oculomotor system for continuous sinusoidal target inputs (0.2 Hz) across nine target amplitudes.

confirm the system's ability to achieve accurate and stable tracking across multiple targets, illustrating both transient and steady-state performance of the hardware-implemented closed-loop control.

To further validate the real-time tracking performance of the proposed hardware architecture under dynamic conditions, the closed-loop system was evaluated using a continuous time-varying reference. Figure 9 presents the digital hardware outputs for a sinusoidal target input across the nine different amplitude levels. Unlike fixed step targets that evaluate transient response and settling time, the sinusoidal input rigorously tests the system's ability to maintain continuous error correction without introducing instability or severe phase lag. As observed in the subplots, the zero-DSP hardware architecture successfully tracks the moving target. These results confirm that the

resource-efficient implementation is fully capable of providing robust closed-loop control for both static fixations and dynamic smooth-pursuit eye movements.

In addition, Table 8 presents a compact view of the closed-loop eye-brain system performance across all input types. Mean Absolute Error (MAE) and root mean square error (RMSE) quantify the average and peak deviations of eye position from the target, respectively. Maximum Error (MaxError) indicates the largest instantaneous deviation observed. Settling percentage shows the fraction of simulation time the system remained within ± 5% of the target. Variance reflects the magnitude of output fluctuations around the target, highlighting stability under dynamic conditions such as sinusoidal and noisy inputs. This format allows easy comparison across multiple targets while conserving space in the manuscript.

It is worth noting that the variance of the eye position for noisy inputs (as reported in Table 8) is significantly lower than for other dynamic trajectories. This is not an anomaly but a direct manifestation of the oculomotor plant's biomechanics; the inertia and viscous damping act as a natural low-pass filter. Consequently, high-frequency neural noise is physically damped and smoothed out, preventing the eye from exhibiting erratic high-frequency oscillations and resulting in a highly stabilised gaze with minimal variance.

Furthermore, to validate the practical robustness of the proposed framework beyond synthetic control inputs, the hardware architecture was benchmarked against real biological data. Human gaze trajectories were obtained from a publicly available eye-tracking dataset (Kaggle Eye Movement Dataset), capturing horizontal eye movements during a continuous text-reading task. The raw biological data, characterised by rapid saccadic jumps and steady fixations, was interpolated to match the system's discrete time step (dt = 0.001 s) and fed into the FPGA hardware emulator.

Figure 10 illustrates the tracking performance of the system under these real-world conditions. The macro-view (top) and zoomed-in micro-view (middle) clearly demonstrate that the hardware's fixed-point digital output successfully emulates and tracks the biological saccades with minimal latency. Despite the severe irregularities inherent to biological eye movements, the system

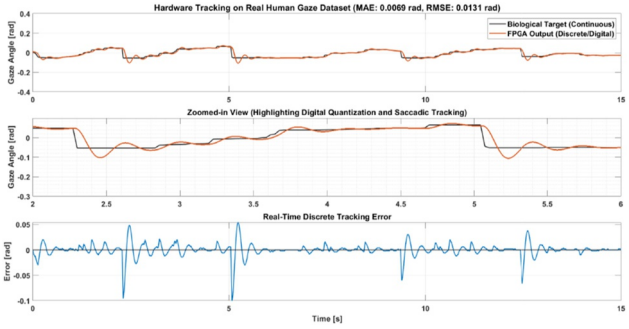


FIGURE 10 | Hardware tracking performance evaluated on a real biological human gaze dataset during a text-reading task. The figure displays the full tracking trajectory (top), a zoomed-in segment highlighting the hardware's digital fixed-point step response to biological saccades (middle), and the real-time discrete tracking error (bottom).

TABLE 8 | Compact performance metrics of the closed-loop eye–brain system for different input types.

Input	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45
MAE step	0.0316	0.0318	0.0326	0.0338	0.0350	0.0366	0.0380	0.0396	0.0414
MAE ramp	0.0313	0.0305	0.0296	0.0287	0.0279	0.0270	0.0261	0.0252	0.0244
MAE sin	0.0322	0.0322	0.0321	0.0321	0.0322	0.0322	0.0323	0.0324	0.0326
MAE noisy	0.0322	0.0322	0.0322	0.0323	0.0324	0.0324	0.0328	0.0334	0.0341
RMSE step	0.0333	0.0355	0.0392	0.0443	0.0499	0.0562	0.0628	0.0697	0.0768
RMSE ramp	0.0318	0.0309	0.0301	0.0292	0.0284	0.0276	0.0268	0.0260	0.0253
RMSE sin	0.0326	0.0327	0.0327	0.0329	0.0330	0.0333	0.0335	0.0339	0.0342
RMSE noisy	0.0327	0.0331	0.0335	0.0342	0.0350	0.0359	0.0372	0.0384	0.0400
MaxErr step	0.0930	0.1290	0.1650	0.2010	0.2500	0.3000	0.3500	0.4000	0.4500
MaxErr ramp	0.0568	0.0575	0.0573	0.0570	0.0568	0.0566	0.0567	0.0566	0.0565
MaxErr sin	0.0562	0.0554	0.0549	0.0549	0.0554	0.0561	0.0570	0.0583	0.0597
MaxErr noisy	0.0628	0.0670	0.0741	0.0774	0.0831	0.0947	0.0977	0.1143	0.1343
Settling step	0.73	1.68	2.53	4.05	5.87	7.34	11.78	65.12	71.51
Settling ramp	0.32	0.48	0.61	0.71	1.04	1.78	2.23	12.24	41.22
Settling sin	0.48	0.77	1.06	2.32	3.17	4.93	5.91	7.99	19.82
Settling noisy	0.25	0.44	0.85	2.64	6.91	13.76	21.52	27.30	32.33
Variance step	0.000167	0.000426	0.000806	0.001307	0.001926	0.002666	0.003525	0.004505	0.005605
Variance ramp	0.000224	0.000813	0.001791	0.003160	0.004920	0.007072	0.009612	0.012546	0.015866
Variance sin	0.001224	0.004811	0.010790	0.019159	0.029917	0.043065	0.058603	0.076531	0.096851
Variance noisy	0.000028	0.000029	0.000029	0.000028	0.000033	0.000028	0.000031	0.000042	0.000042

Note: Metrics include MAE, RMSE, MaxError, settling percentage, and variance for nine targets.

maintained tight closed-loop stability, yielding a highly accurate tracking response with a mean absolute error (MAE) of 0.0069 rad and a root mean square error (RMSE) of 0.0131 rad, as depicted in the error profile (bottom). This experiment explicitly confirms the system's viability for real-time neuromorphic emulation and clinical brain-machine interface (BMI) applications.

4.4 | Comparison With State-of-the-Art Implementations

To fully evaluate the hardware efficiency and performance of the proposed closed-loop brain-eye system, Table 9 presents a comprehensive quantitative comparison with state-of-the-art FPGA and ASIC implementations of eye-tracking, saccade classification, and oculomotor control systems, explicitly including the recent implementations highlighted in [24–27]. As shown in the table, previous approaches heavily rely on resource-intensive algorithms such as convolutional neural networks (CNNs) [27, 44], spiking neural networks (SNNs) [37, 39] and complex morphological or projection operations [24–26, 38, 42]. Consequently, these systems consume significant logic resources and essentially require hardware multipliers (DSP slices) or external memory interfaces. For instance, recent conventional FPGA-based vision trackers [24–26] operate at standard camera frame rates, inherently introducing 16–33 ms of processing latency. In contrast, our proposed reduced-order oculomotor model is explicitly optimised for a zero-DSP architecture. By replacing complex arithmetic with cost-effective shift-add logic, our design completely eliminates the need for DSP blocks and utilises merely 765 LUTs and 589 FFs on a Virtex-4 platform. Furthermore, the streamlined multiplier-less data path yields an ultra-low closed-loop latency of 90 ns, which is orders of magnitude faster than both conventional camera-based trackers [24–26, 40, 42] and recent event-based neuromorphic architectures [44, 45]. The power consumption is also maintained at a modest 280 mW. Overall, this comparison demonstrates that the proposed hardware-algorithm codesign offers a highly scalable and resource-efficient solution for real-time low-latency neuromorphic control systems.

4.5 | Limitations and Future Work

Although the proposed framework demonstrates significant hardware efficiency and real-time performance, certain limitations must be acknowledged. Regarding robustness, although the physical inertia of the model effectively dampens standard

TABLE 9 | Comprehensive comparison of state-of-the-art hardware-accelerated eye tracking and oculomotor control systems.

Ref.	Application	Algorithm	Platform	Type	Resources (L = LUT, F = FF, D = DSP, B = BRAM)	Freq.	Latency	Power
[24]	Eye tracking	Projection Alg.	Ti60F100	HW	N/A L, N/A F, N/A D, N/A B	N/A	16.6 ms	N/A
[25, 26]	Pupil tracking	Black Box Tech.	Zynq SoC	HW/SW	9056 L, 5892 F, > 0 D, N/A B	100 MHz	33.3 ms	N/A
[27]	Saccade class.	CNN	Zynq-7020	HW/SW	32.7 K L, 31 K F, 172 D, 111 B	N/A	79 µs	2.53 W
[37]	Motor control	SNN	Spartan-6 LX100	HW	57 K L, 22.5 K F, 160 D, 255 B	40 MHz	1 ms	0.6 W
[38]	Eye tracking	Morph. Ops	Cyclone-II	HW	8829 L, N/A F, 28 D, 52 Kb B	27 MHz	118.9 ms	375.3 mW
[39]	OKR adapt.	Large-scale SNN	6 × Stratix III	HW	N/A L, N/A F, 0 D, N/A B (3.5 M neurons)	50 MHz	200 ns	N/A
[40]	Pupil tracking	LP filter + ellipse	Kintex-7 + i7 CPU	HW/SW	N/A L, N/A F, N/A D, N/A B	100 MHz	0.6–2.1 ms	N/A
[41]	Blink/saccade	FIR + linear SVM	Zynq-7000	HW/SW	19,684 L, 4256 F, 119 D, 1 B	N/A	0.29 µs	0.8 W
[42]	Pupil tracking	Morph. Ops	Zynq-7020	HW	9.6 K L, 11.7 K F, 11 D, 40 B	100 MHz	1.8 ms	247 mW
[43]	Ellipse Estim.	RANSAC	Stratix III	HW	18.7 K L, 20.6 K F, 44 D, 2.75 Mb B	119.5 MHz	3.05 µs	3.35 W
[44]	Event-based ET	SCNN + GRU	Zynq US+	HW/SW	135.1 K L, 94 K F, 1880 D, 1001 B	250 MHz	0.61 ms	3.86 W
[45]	Event-based ET	ConvJANET	12-nm ASIC	HW	17.6 K params, 0.28 mm^2 Area, 175 KB SRAM	400 MHz	0.26 ms	37.7 mW
[46]	Eye detection	Tiny YOLO-v3	Zynq US+	HW/SW	66.7 K L, 94.8 K F, 335 D, 182 B	150 MHz	11.51 ms	N/A
[47]	Gaze tracking	POLONet (ViT + RNN)	22-nm ASIC	HW/SW	0.8 M params, 0.75 mm^2 Area, 256 KB SRAM	1 GHz	9.81 ms	150.4 mW
This work	Brain-eye control	Reduced-order model	Virtex-4	HW	765 L, 589 F, 0 D, 64 Kb B	100 MHz	90 ns	280 mW

physiological noise (as validated by our real-world dataset benchmarking), its reliance on a static linear proportional controller restricts its adaptability. Under severe unpredictable sensorimotor delays or highly complex visual stimuli, this static gain may exhibit limited robustness compared to adaptive or nonlinear control strategies. Regarding scalability, the current zero-DSP architecture is a deliberate design choice explicitly optimised for decoupled, low-order 1D/2D oculomotor tracking. Scaling the system to support fully coupled 3D ocular kinematics (e.g., Listing's law) or binocular coordination would reintroduce complex nonlinear cross-coupling terms. Implementing these nonlinearities in hardware would necessitate standard multipliers (DSPs), thereby trading the current ultra-low resource footprint and 90 ns latency for higher biological fidelity. Consequently, our future work will explore adaptive gain-scheduling mechanisms and multi-loop expansions, aiming to optimally balance hardware constraints with advanced 3D neuromorphic emulation.

Furthermore, to elevate the proposed framework from a purely reflexive motor controller to a higher-level cognitive visual tracking system, future expansions will draw inspiration from recent large-scale digital neuromorphic architectures. Integrating our highly optimised ocular plant model with multi-granular cognitive architectures, such as BiCoSS [48], or incorporating biophysically detailed multi-compartment spiking neural networks [49], could pave the way for a fully comprehensive brain-eye emulation platform. Such integration would allow the hardware to simulate complex visual pursuit tasks requiring memory and predictive tracking, albeit at the cost of increased hardware resource utilization.

5 | Conclusion

This paper presented a complete closed-loop brain–eye movement framework, covering simplified modelling, dynamic analysis and real-time FPGA implementation. The reduced-order oculomotor model preserved essential neural–mechanical feedback, enabling accurate and stable eye tracking across a broad range of parameter variations and target inputs. The FPGA realization employed a fully arithmetic-based design without multipliers, achieving low resource utilization, sub-microsecond latency and modest power consumption. Experimental validation using a 3×3 multi-target grid confirmed the system's ability to execute real-time closed-loop control under realistic conditions. Future work will focus on extending the framework to include nonlinear neural dynamics, adaptive control strategies, and sensory delays for increased biological fidelity. Moreover, scaling to multi-loop architectures and integrating advanced sensory feedback on modern FPGA or SoC platforms can further broaden applications in neuromorphic vision, brain–machine interfaces and real-time neurophysiological emulation.

Funding

This work was supported in part by Natural Science Foundation of Zhejiang Province (LQ24F030013). The authors acknowledge the Supercomputing Center of Hangzhou Dianzi University for providing computing resources.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

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

References

1. P. Bonazzi, S. Bian, G. Lippolis, Y. Li, S. Sheik, and M. Magno, "Retina: Low-Power Eye Tracking With Event Camera and Spiking Hardware," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2024), 5684–5692.

2. A. Tanwear, X. Liang, Y. Liu, et al., "Spintronic Sensors Based on Magnetic Tunnel Junctions for Wireless Eye Movement Gesture Control," *IEEE Transactions on Biomedical Circuits and Systems* 14, no. 6 (2020): 1299–1310, https://doi.org/10.1109/tbcas.2020.3027242.

3. T. Saquib and D. Terzopoulos, "Biomimetic Oculomotor Control With Spiking Neural Networks," *Machine Vision and Applications* 35, no. 1 (2024): 14, https://doi.org/10.1007/s00138-023-01494-z.

4. R. J. Alitappeh, A. John, B. Dias, A. J. van Opstal, and A. Bernardino, "Emergence of Human Oculomotor Behavior in a Cable-Driven Biomimetic Robotic Eye Using Optimal Control," *IEEE Transactions on Cognitive and Developmental Systems* 16, no. 4 (2024): 1546–1560, https://doi.org/10.1109/tcds.2024.3376072.

5. C. Wu, N. Chen, T. Sun, P. Tan, P. Wang, and G. Li, "Synergizing Deepseek's Artificial Intelligence Innovations With Brain-Computer Interfaces," *Brain-X* 3, no. 2 (2025): e70035, https://doi.org/10.1002/brx2.70035.

6. D. A. Robinson, "The Mechanics of Human Saccadic Eye Movement," *Journal of Physiology* 174, no. 2 (1964): 245–264, https://doi.org/10.1113/jphysiol.1964.sp007485.

7. D. A. Robinson, "Models of the Saccadic Eye Movement Control System," *Kybernetik* 14, no. 2 (1973): 71–83, https://doi.org/10.1007/bf00288906.

8. M. E. Broucke, "Model of the Oculomotor System Based on Adaptive Internal Models," *IFAC-PapersOnLine* 53, no. 2 (2020): 16430–16437, https://doi.org/10.1016/j.ifacol.2020.12.729.

9. M. T. Islam, F. Hazzazi, A. Hoque, S. Haghiri, M. A. Chaudhary, and M. Ghanbarpour, "FPGA Implementation of Nerve Cell Using Izhikevich Neuronal Model as Spike Generator (SG)," *IEEE Access* 12 (2023): 2303–2312, https://doi.org/10.1109/ACCESS.2023.3343156.

10. M. A. Chaudhary, F. Hazzazi, and M. Ghanbarpour, "Digital System Implementation and Large-Scale Approach in Neuronal Modeling Using Adex Biological Neuron," *IEEE Transactions on Circuits and Systems II: Express Briefs* 71, no. 5 (2024): 2814–2818, https://doi.org/10.1109/TCSII.2024.3353333.

11. M. Ghanbarpour, A. Naderi, B. Ghanbari, S. Haghiri, and A. Ahmadi, "Digital Hardware Implementation of Morris-Lecar, Izhikevich, and Hodgkin-Huxley Neuron Models With High Accuracy and Low Resources," *IEEE Transactions on Circuits and Systems I: Regular Papers* 70, no. 11 (2023): 4447–4455, https://doi.org/10.1109/TCSI.2023.3303941.

12. G. Ghanbarpour and M. Ghanbarpour, "Low-Error, High-Speed, and Large-Scale Hardware Implementation of Retinal Photoreceptor Cells: Cone and Rod Cells," *AEU-International Journal of Electronics and Communications* 185 (2024): 155456, https://doi.org/10.1016/j.aeue.2024.155456.

13. M. Ghanbarpour, S. Haghiri, F. Hazzazi, M. Assaad, M. A. Chaudhary, and A. Ahmadi, "Investigation on Vision System: Digital FPGA Implementation in Case of Retina Rod Cells," *IEEE Transactions*

on Biomedical Circuits and Systems 18, no. 2 (2023): 299–307, https://doi.org/10.1109/TBCAS.2023.3323324.

14. M. Ghanbarpour, A. Naderi, S. Haghiri, and A. Ahmadi, "An Efficient Digital Realization of Retinal Light Adaptation in Cone Photoreceptors," *IEEE Transactions on Circuits and Systems I: Regular Papers* 65, no. 12 (December 2021): 5072–5080, https://doi.org/10.1109/tcsi.2021.3111202.

15. A. J. Leigh, M. Heidarpur, and M. Mirhassani, "A Resource-Efficient and High-Accuracy CORDIC-Based Digital Implementation of the Hodgkin–Huxley Neuron," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 31, no. 9 (2023): 1377–1388, https://doi.org/10.1109/tvlsi.2023.3296057.

16. J. Lu, F. Min, L. Gan, and S. Yang, "Dynamic Analysis of Coupled Hindmarsh-Rose Neurons With Enhanced FPGA Implementation," *Chaos, Solitons & Fractals* 191 (2025): 115889, https://doi.org/10.1016/j.chaos.2024.115889.

17. S. Nazari and S. Jamshidi, "Efficient Digital Design of the Nonlinear Behavior of Hindmarsh–Rose Neuron Model in Large-Scale Neural Population," *Scientific Reports* 13, no. 1 (2024): 3833, https://doi.org/10.1038/s41598-024-54525-8.

18. Y. Chen, X. Wu, H. Ba, et al., "Digital Approach in Case of FPGA Realization of Quartic Neuron Model (QNM) Using Cost-Effective Mathematical Modifications," *IEEE Transactions on Circuits and Systems I: Regular Papers* 71, no. 12 (2024): 6327–6335, https://doi.org/10.1109/tcsi.2024.3400399.

19. J. Sun, H. Chen, X. Ji, G. Zhang, C. Wang, and X. Miao, "An Investigation on the Three-Dimensional Memristive Morris–Lecar Model With Magnetic Induction Effects: Simulation of Biological Behaviors and Cost-Effective Digital Circuit Implementation," *IEEE Transactions on Circuits and Systems I: Regular Papers* 72, no. 6 (2025): 2690–2700, https://doi.org/10.1109/TCSI.2025.3534329.

20. S. Liu, B. Ni, X. Wang, et al., "Efficient Low-Cost Design for High-Frequency, Accurate Hardware Implementation of Cardiac Purkinje Fiber Cells for Biomedical Engineering Approaches," *Engineering Science and Technology, an International Journal* 60 (2024): 101888, https://doi.org/10.1016/j.jestch.2024.101888.

21. R. Chen, S. Xiang, D. Yu, et al., "An Investigation on Inner Ear Operation in Sense of Hearing: High-Speed FPGA-based Circuits for Digital Realization of 2D Cochlea Modeling," *Alexandria Engineering Journal* 118 (2025): 371–383, https://doi.org/10.1016/j.aej.2025.01.054.

22. S. Lin, S. Xiang, R. Chen, L. Li, Y. Ge, and X. Gao, "Digital Low-Cost FPGA Implementation of Two-Coupled and Grid-Based Network of 2D Artificial Cochlea Using the Hopf Resonator Approach," *IEEE Transactions on Circuits and Systems I: Regular Papers* 72, no. 5 (2024): 2178–2188, https://doi.org/10.1109/TCSI.2024.3494268.

23. V. Lopez, C. Montalvo, A. Varela, O. Cham, and E. Galvan, "A Review of Design Methodologies and Evaluation Techniques for FPGA-Based Visual Object Tracking Systems," *International Journal of Combinatorial Optimization Problems and Informatics* 15, no. 5 (2024), https://doi.org/10.61467/2007.1558.2024.v15i5.571.

24. P. Zhang, K. Jia, Y. Liu, Y. He, L. Chen, and C. Liu, "RTL Design and FPGA Implementation of a Real-Time Eye Tracking System Using a Novel Projection Algorithm," in *Fifth International Conference on Signal Image Processing and Communication (ICSIPC 2025)* (2025), 366–373.

25. N. Subramani, A. Varshini, A. Anil, and C. Thanusha, "A Real-Time Human Eye Pupil Tracking System Using FPGA," *AIP Conference Proceedings* (2025), https://doi.org/10.1063/5.0247030.

26. G. K. Kumar and B. Narayanam, "FPGA Implementation of Eye Movement Detection Algorithm," *Microprocessors and Microsystems* (2021): 104364, https://doi.org/10.1016/j.micpro.2021.104364.

27. C. Domingo, R. Stoean, and G. Caparros, "FPGA Design and Hardware Implementation of a Convolutional Neural Network for Classification of Saccadic Eye Movements," in *Annals of the University of Craiova-Mathematics and Computer Science Series* (2018), 263–274.

28. L. Manfredi, E. S. Maini, P. Dario, et al., "Implementation of a Neurophysiological Model of Saccadic Eye Movements on an Anthropomorphic Robotic Head," in *2006 6th IEEE-RAS International Conference on Humanoid Robots* (2006), 438–443.

29. A. Al-Rahayfeh and M. Faezipour, "Eye Tracking and Head Movement Detection: A State-of-Art Survey," *IEEE Journal of Translational Engineering in Health and Medicine* (2013), https://doi.org/10.1109/JTEHM.2013.2289879.

30. F. Wadehn, T. Weber, D. J. Mack, T. Heldt, and H. A. Loeliger, "Model-Based Separation, Detection, and Classification of Eye Movements," *IEEE Transactions on Biomedical Engineering* 67, no. 2 (2020): 588–600, https://doi.org/10.1109/tbme.2019.2918986.

31. D. A. Robinson, "The Use of Control Systems Analysis in the Neurophysiology of Eye Movements," *Annual Review of Neuroscience* 4, no. 1 (1981): 463–503, https://doi.org/10.1146/annurev.ne.04.030181.002335.

32. P. A. Sylvestre and K. E. Cullen, "Quantitative Analysis of Abducens Neuron Discharge Dynamics During Saccadic and Slow Eye Movements," *Journal of Neurophysiology* 82, no. 5 (1999): 2612–2632, https://doi.org/10.1152/jn.1999.82.5.2612.

33. V. Radisavljevic-Gajic, "Dynamics of Eye Movements Under Time Varying Stimuli," *Journal of Eye Movement Research* 11, no. 1 (2018): 6, https://doi.org/10.16910/jemr.11.1.6.

34. M. Heidarpur, A. Ahmadi, M. Ahmadi, and M. Rahimi Azghadi, "CORDIC-SNN: On-Fpga STDP Learning With Izhikevich Neurons," in *IEEE Transactions on Circuits and Systems-I* (2019), 2651–2661.

35. M. Heidarpur, P. Khosravifar, A. Ahmadi, and M. Ahmadi, "CORDIC-Astrocyte: Tripartite Glutamate-IP3-Ca2+ Interaction Dynamics on FPGA," *IEEE Transactions on Biomedical Circuits and Systems* 14, no. 1 (February 2020): 36–47, https://doi.org/10.1109/tbcas.2019.2953631.

36. H. Soleimani and E. M. Drakakis, "An Efficient and Reconfigurable Synchronous Neuron Model," *IEEE Transactions on Circuits and Systems II: Express Briefs* (2017), https://doi.org/10.1109/TCSII.2017.2697826.

37. Y. Shinji, H. Okuno, and Y. Hirata, "Artificial Cerebellum on FPGA: Realistic Real-Time Cerebellar Spiking Neural Network Model Capable of Real-World Adaptive Motor Control," *Frontiers in Neuroscience* (2024), https://doi.org/10.1109/ICASSP.2012.6288193.

38. T. Ando, V. G. Moshnyaga, and K. Hashimoto, "A Low-Power FPGA Implementation of Eye Tracking," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), 1573–1576.

39. S. Yang, J. Wang, N. Zhang, B. Deng, Y. Pang, and M. R. Azghadi, "CerebelluMorphic: Large-Scale Neuromorphic Model and Architecture for Supervised Motor Learning," *IEEE Transactions on Neural Networks and Learning Systems* 33, no. 9 (2022): 4398–4412, https://doi.org/10.1109/TNNLS.2021.3057070.

40. B. Kowalski, X. Huang, S. Steven, and A. Dubra, "Hybrid FPGA-CPU Pupil Tracker," *Biomedical Optics Express* 12, no. 10 (2021): 6496–6513, https://doi.org/10.1364/BOE.433766.

41. D. Das, M. H. Chowdhury, A. Chowdhury, K. Hasan, Q. D. Hossain, and R. C. C. Cheung, "Application Specific Reconfigurable Processor for Eyeblink Detection From Dual-Channel EOG Signal," *Journal of Low Power Electronics and Applications* 13, no. 4 (2023): 61, https://doi.org/10.3390/jlpea13040061.

42. A. M. Z. Khaki, S. Yang, H. Kim, A. Singh, and B.-G. Lee, "A Resource-Efficient Multi-Function Embedded Eye Tracker System Implemented on FPGA," *IEEE Access* 11 (2023): 142931–142939, https://doi.org/10.1109/ACCESS.2023.3342839.

43. K. Dohi, Y. Hatanaka, K. Negi, Y. Shibata, and K. Oguri, "Deep-Pipelined FPGA Implementation of Ellipse Estimation for Eye Tracking," in *22nd International Conference on Field Programmable Logic and Applications (FPL)* (2012), 458–463.

44. B. Zhang, Y. Gao, J. Li, and H. K.-H. So, "Co-Designing a Sub-Millisecond Latency Event-Based Eye Tracking System With Submanifold Sparse CNN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2024), 5771–5779.

45. T. Han, A. Li, Q. Chen, and C. Gao, "JaneEye: A 12-nm 2K-FPS 18.9-μJ/Frame Event-Based Eye Tracking Accelerator," in *2026 31st Asia and South Pacific Design Automation Conference (ASP-DAC)* (IEEE, 2026), 170–176.

46. C. A. Ruiz-Beltrán, A. Romero-Garcés, M. González-García, R. Marfil, and A. Bandera, "FPGA-Based CNN for Eye Detection in an Iris Recognition at a Distance System," *Electronics* 12, no. 22 (2023): 4713, https://doi.org/10.3390/electronics12224713.

47. H. Wang, W. Liu, and Q. Sun, "Process Only Where You Look: Hardware and Algorithm Co-Optimization for Efficient Gaze-Tracked Foveated Rendering in Virtual Reality," preprint, arXiv (2024).

48. S. Yang, J. Wang, X. Hao, et al., "BiCoSS: Toward Large-Scale Cognition Brain With Multigranular Neuromorphic Architecture," *IEEE Transactions on Neural Networks and Learning Systems* 33, no. 7 (July 2022): 2801–2815, https://doi.org/10.1109/tnnls.2020.3045492.

49. S. Yang, B. Deng, J. Wang, et al., "Scalable Digital Neuromorphic Architecture for Large-Scale Biophysically Meaningful Neural Network With Multi-Compartment Neurons," *IEEE Transactions on Neural Networks and Learning Systems* 31, no. 1 (January 2020): 148–162, https://doi.org/10.1109/tnnls.2019.2899936.