2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Bi-Hap: a Bi-directional Learning-Based Control and Momentum-based Haptic Feedback System for Dexterous In-hand Telemanipulation

Haoyang Wang[1], Haoran Guo[2], Zhengxiong Li[3], *Member, IEEE*,
and Lingfeng Tao[4], *Member, IEEE*

***Abstract*— Dexterous in-hand telemanipulation demands precise control and realistic haptic feedback to achieve stable and intuitive human–robot interaction. Existing systems often emphasize isolated control policies or unidirectional force feedback, limiting performance in tasks that require coordinated bidirectional information flow. In this work, we introduce Bi-Hap, a bi-directional learning-based control and momentum-based haptic feedback system for real-time, in-hand telemanipulation. On the control side, Bi-Hap leverages an inertial measurement unit to capture operator motion and drives a deep reinforcement learning policy that enables robust and adaptive manipulation of objects with fine rotational dexterity. On the feedback side, a compact, palm-sized momentum-actuated mechanism delivers torque and vibration cues directly to the operator, augmented by an error-adaptive strategy that modulates feedback intensity based on task states. When integrated, this closed-loop design establishes an immersive bidirectional control–feedback framework. Experimental results show that Bi-Hap achieves low feedback latency (<0.03s), high torque fidelity (RMSE <0.01Nm), and significantly improved telemanipulation performance by elevating manipulation accuracy, responsiveness, and operator situational awareness in diverse task settings.**

I. Introduction

Dexterous in-hand telemanipulation has emerged as a key capability in domains such as human–robot collaboration [1], advanced telerehabilitation [2], and imitation learning [3], where operators must precisely and intuitively control an object's pose within a robotic hand. Unlike conventional teleoperation with simple grippers, in-hand manipulation requires fine skills such as rotation, re-grasping, and dynamic adjustment under uncertain contact conditions [4], [5]. Achieving this remotely requires not only accurate motion control but also realistic haptic cues that convey both force and torque. Although many model-based and kinematic-mapping approaches are valued for their structured design and ease of implementation [6], they place the burden of adjustment on the operator, resulting in inefficiency and a considerable cognitive load. At the same time,

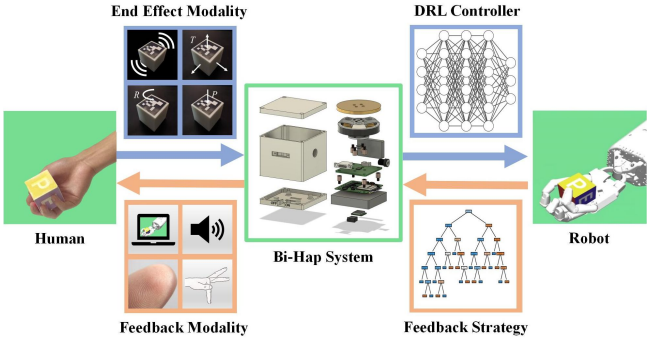


Fig. 1: The end-effect modality captures task-level interaction data and provides it to a DRL controller that drives the robot to reproduce these interactions. Torque and vibration feedback are adaptively modulated, forming a closed-loop telemanipulation framework for precise and intuitive haptic control. A palm-sized cubic device is designed to demonstrate the Bi-Hap system, though the framework is not limited to this shape.

most existing haptic systems focus primarily on force feedback while leaving torque feedback underexplored [7], [8], thereby limiting the operator's ability to perceive rotational dynamics, which is essential information for manipulation tasks. These limitations leave a significant gap between manipulation dexterity and the bidirectional interaction required for immersive teleoperation.

Recent advances in deep reinforcement learning (DRL) serve as a strong methodological foundation for tackling the control aspect of this challenge. DRL has demonstrated the ability to learn complex dexterous skills such as object reorientation, finger gaiting, and dynamic stabilization in high-dimensional environments where analytical models struggle [9], [10]. By representing telemanipulation as a human–robot Markov game, DRL policies can incorporate human intent as partial observations and translate them into robust low-level robot actions [11]. However, most prior DRL-based telemanipulation frameworks only allow the operator to perceive on the robot side visually. Such frameworks largely conceptualize the human operator as an open-loop output conduit and seldom incorporate the operator's haptic feedback into the closed control loop, overlooking its essential role in intuitive telemanipulation.

Existing haptic devices have significantly advanced tactile and kinesthetic feedback in teleoperation. However, their effectiveness in in-hand tasks such as object reorientation remains limited. For example, DoGlove [7] offers multi-finger force feedback but lacks rotational cues, whereas DextrES [8] reproduces grasping force through electrostatic brakes, but does not provide torque, which limits the operator's perception and control of rotational

*This work is partially supported by the NVIDIA Academic Grant Program Award, KSU MOVE center, and US NSF Awards #2426469, #2426470, #2436428, and #2602731.

[1]H. Wang is with the University of Alberta, Department of Mechanical Engineering, Donadeo Innovation Centre for Engineering, Edmonton, AB T6G 1H9, Canada (e-mail: haoyan25@ualberta.ca).

[2]H. Guo is with the Oklahoma State University, 63 Engineering North, Stillwater, OK, 74078, USA (e-mail: haoran.guo@oksate.edu).

[3]Z. Li is with the University of Colorado Denver, Department of Computer Science and Engineering, 1380 Lawrence St. Center, LW-834, Denver. CO 80217, USA (e-mail: zhengxiong.li@ucdenver.edu).

[4]L. Tao is with the Kennesaw State University, Department of Robotics and Mechatronics Engineering, 840 Polytechnic Lane, Marietta, GA 30060, USA (e-mail: ltao2@kennesaw.edu).

dynamics in complex telemanipulation tasks. Some existing prototypes attempt to provide torque feedback but rely on large, complex mechanical structures [12] that are unsuitable for in-hand applications. These limitations highlight the absence of compact, effective solutions for delivering torque feedback in in-hand telemanipulation.

Recently, momentum-based haptic devices have shown strong potential for torque and vibration feedback [13]. Unlike conventional haptic devices, which are anchored to the ground with motors or actuators, momentum-based systems generate inertial forces through spinning masses. This enables a compact and portable design that delivers rich haptic feedback, including torque, impact simulation, and both low-frequency and high-amplitude vibrations. Despite these advantages, momentum-based haptic devices have not yet been explored in the context of dexterous in-hand telemanipulation. One reason is that most existing devices require full-palm grasping [14], [15], and their irregular shapes make it difficult for fingers to manipulate them dexterously within the hand. Additionally, the complexity of controlling such systems due to the inherent inertia of the spinning masses leads to feedback latency, which is critical in dynamic telemanipulation tasks.

In this work, we present Bi-Hap, a Bi-directional Learning-Based Control and Momentum-Based Haptic Feedback System for real-time dexterous in-hand telemanipulation, as illustrated in Fig. 1. Bi-Hap integrates two complementary components into a unified closed-loop framework. On the control side, an inertial measurement unit (IMU) captures the operator's end-effect motion, which is mapped into an end-effect-oriented DRL controller for robust and adaptive telemanipulation under contact-rich dynamics. On the feedback side, a lightweight ($<$200g), palm-sized momentum-actuated mechanism delivers perceivable torque and vibration cues directly to the operator's hand without external grounding. We also introduce an error-adaptive strategy to modulate haptic feedback intensity according to manipulation states, improving operator perception, error correction, and decision-making. Extensive experiments demonstrate that Bi-Hap can significantly enhance control precision and situational awareness in in-hand object rotation tasks while maintaining low latency.

The contributions of this work are summarized as follows:

1). We design Bi-Hap as the first integrated system that couples learning-based control with momentum-based ungrounded haptic feedback for dexterous telemanipulation.

2). We develop a DRL control policy that mimics a human operator's interaction with the object and generates robust in-hand manipulation commands, bridging human partial observations with robot execution.

3). We design a palm-sized torque and vibration feedback device and introduce an error-adaptive strategy to adjust feedback in response to manipulation states dynamically.

4). We evaluate Bi-Hap in simulation and real-world telemanipulation experiments, demonstrating low-latency control–feedback integration and measurable improvements in manipulation accuracy and responsiveness.

II. Related Work

A. DRL-based Manipulation Control System

DRL has emerged as a powerful tool for robotic manipulation, particularly in high-dimensional, contact-rich environments where classical model-based controllers struggle to handle complex interactions [16]. Prior work has applied DRL to dexterous hands control, enabling complex skills such as object reorientation, finger gaiting, and coordinated in-hand manipulation [9], [10]. Policy representations combining multilayer perceptrons with recurrent modules have demonstrated the ability to capture temporal dependencies in manipulation tasks [17], while domain randomization and sim-to-real transfer strategies further improve robustness under real-world uncertainties [18]. Beyond standard DRL, recent work such as DEFT leverages human video demonstrations to build affordance priors for grasp and contact prediction, and then employs online residual reinforcement learning to fine-tune policies directly in the real world, enabling sample-efficient dexterous manipulation of both rigid and deformable objects with anthropomorphic soft hands [19].

Despite these progresses, challenges still remain in achieving reliable deployment for telemanipulation, as most DRL frameworks assume fully autonomous settings and thereby neglect critical issues such as human-in-the-loop integration. In particular, current methods rarely account for operator intent and feedback, which further limits their effectiveness in real-world teleoperation.

B. Telemanipulation with Haptic Feedback

Telemanipulation bridges human expertise and robotic dexterity, enabling real-time control of high-DoF hands. Recent systems such as Tilde demonstrate that kinematic twin interfaces can achieve precise, low-latency in-hand control [20]. Within this context, haptic feedback is a vital complement to telemanipulation [21], particularly for tasks requiring fine dexterity and interaction with complex objects. By delivering real-time tactile and kinesthetic cues, it enhances perception of contact forces, stiffness, and dynamic interaction, leading to more intuitive and stable control [7]. Studies further show that force feedback reduces task time and errors in delicate operations such as needle insertion [22], object alignment [23], and vehicle assembly [24].

Traditional haptic interfaces, such as grounded force-feedback arms or exoskeleton gloves, have demonstrated the benefits for immersive and precise interaction. However, there remains a significant gap in providing effective torque feedback, which is crucial for tasks involving object rotation within the hand, such as turning a screwdriver, twisting a bottle cap, or adjusting a knob [25]. While some high-DOF haptic gloves attempt to emulate torque sensations by applying 3D fingertip forces [12], this remains indirect and is further constrained by the limited workspace of the hand. Moreover, such systems typically rely on bulky, grounded actuation, limiting their use in in-hand telemanipulation where lightweight, integrated feedback is essential.

C. Momentum-based Haptic Feedback Device

Momentum-based haptic devices offer a promising solution for ungrounded torque rendering. Leveraging rotating flywheels or control moment gyroscopes (CMGs), these systems generate perceptible torques without requiring a physical ground [13], making them suitable for in-hand or mobile contexts. Several studies have demonstrated the feasibility of such designs for handheld applications. For example, Walker et al. [26] developed a compact dual-CMG device delivering clear directional torque cues by rapidly reorienting counter-rotating flywheels with asymmetric pulses. Similarly, Berna Moya et al. [14] proposed HapticWhirl, a handheld controller using a flywheel and dual-axis gimbal to deliver multimodal feedback including torque, vibration, and inertial cues. Notably, Hashimoto et al. [15] designed MetamorphX, an ungrounded 3-DoF moment display with four CMGs, which enables low-latency rotational impedance feedback and offers realistic inertia and viscosity changes for VR object manipulation. More recently, Yang et al. [27] introduced TorqueCapsules, fully encapsulated flywheel modules that emphasize modularity, safety, and rapid prototyping of kinesthetic haptics, thereby broadening the design space for momentum-based interactions.

These designs offer promising avenues for wearable or handheld torque feedback in virtual reality (VR) systems. However, they are not specifically tailored for dexterous in-hand telemanipulation. The devices are typically grasped by the whole hand [14], [15], [28], with torque cues primarily perceived through wrist rotation rather than finger-level interaction. Moreover, their overly heavy mechanical structures and complex geometric shapes make it difficult to perform fine in-hand movements such as translation and reorientation using the fingers alone. This limits their applicability in scenarios requiring precise object manipulation within the operator's palm.

III. Bi-Directional Haptic Feedback and Control System

A. DRL-based Modeling and End-Effect-Oriented Control

DRL methods have shown strong capability in dexterous in-hand tasks such as pen rotation [29] and Rubik's cube solving [17]. To better understand and generalize such capabilities, we start from a mathematical formulation. A single-agent DRL problem is typically formulated as a Markov Decision Process (MDP), defined by a tuple $\{S, A, R, \gamma\}$, where S is the state space, A the action set, $R : S \times A \to \mathbb{R}$ the reward, and $\gamma \in [0, 1]$ the discount factor. The goal is to maximize cumulative reward. Unlike single-agent cases, telemanipulation involves two agents: **a human operator** and **a robot**. Thus, we model the telemanipulation problem as a Markov Game, an extension of MDPs, in which each agent maintains its own policy $\pi_i : S \to A_i$ while sharing the same environment dynamics.

In practice, the human mind is a black box, making the human MDP a Partially Observable MDP (POMDP) [30], where internal transitions, actions, and rewards cannot

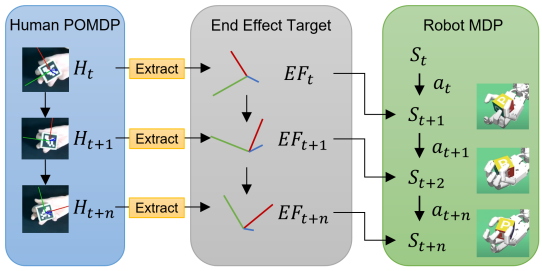


Fig. 2: We model the telemanipulation tasks as a Markov Game that involves two agents: a human operator and a robot. The robot's MDP depends on the operation command from the state of the human's POMDP because the human's mind is a black box and only the end effects that the human applies to the object can be observed by the robot.

be directly accessed, and only physical interactions are observable. This motivates our end-effect-oriented approach, which separates the operator's hidden decision process from observable outcomes and converts partial observations into robot interpretable goals. The operator follows a latent process H_t, inaccessible to the robot, only end-effect targets EF_t, representing object-level effects of human actions, are observable. The robot's state S_t then evolves under its policy π_r, conditioned on its dynamics and the sequence of EF_t, generating actions a_t that update the state to S_{t+1} and enable coordinated telemanipulation (Fig. 2). To implement this framework, a 6-DoF IMU on the Bi-Hap hardware captures real-time end-effect motion EF_t, which is mapped into the robot's observation space. These compact signals provide object-level goals that guide π_r despite partial human observability. The control policy is trained in simulation with Deep Deterministic Policy Gradient (DDPG) and Hindsight Experience Replay (HER). The learned policy integrates joint states, object poses, and EF_t to achieve robust manipulation under contact-rich dynamics, and is later deployed on the host computer for real-time telemanipulation.

B. Error-Adaptive Feedback Strategy

We formulate error-adaptive feedback as a decision-making problem under partial observability. The system state is $s_t = (x_t, z_t)$, where $x_t = (e_t, \dot{e}_t, \tau_t^{\text{fb}}, \alpha_t^{\text{fb}}, \sigma_t)$ collects observable variables: orientation error e_t, its rate $\dot{e}_t$, delivered torque τ_t^{fb}, vibration amplitude α_v, and success indicator σ_t. The latent component z_t captures human-related factors such as stress or fatigue. At each step, the controller chooses an action $f_t \in \mathcal{A}_f = \{\text{Haptic}(\tau_t^{\text{fb}}), \text{Vibrotactile}(\alpha_t^{\text{fb}})\}$, where τ_t^{fb} encodes continuous haptic torque and α_v encodes vibrotactile intensity. The Bi-Hap device thus combines continuous torque for interaction forces with high-frequency vibration to signal overshoot, abrupt changes, or failure states. Visual cues are always available as baseline feedback and excluded from the adaptive action space.

Since z_t is unobservable, the controller constructs a belief summary to approximate the operator workload [31]. We define the summary vector as $\phi_t = [|e_t|, |\dot{e}_t|, |\tau_t^{\text{fb}}|, |\alpha_t^{\text{fb}}|, |\sigma_t|]$, which aggregates error, dynamics, feedback effort, vibration activity, and task success. The workload estimate is then computed as:

$$\bar{\ell}_t = \sigma\big(w^\top\big(\lambda\phi_{t-1} + (1-\lambda)\phi_t\big)\big) \tag{1}$$

where $w \geq 0$ is a weight vector, $\lambda \in [0, 1)$ a temporal smoothing factor, and $\sigma(\cdot)$ a logistic function ensuring $\bar{\ell}_t \in (0, 1)$. This representation integrates multimodal feedback signals into a single workload indicator. Importantly, $\bar{\ell}_t$ does not directly produce torque or vibration, but instead modulates the decision boundaries of the rule-based policy. Higher workload estimates lower the tolerance thresholds and increase feedback sensitivity, while lower values relax the thresholds to minimize unnecessary intervention.

To operationalize this framework, we parameterize the rule-based policy using interpretable thresholds and impedance control gains. A success threshold θ_s determines task completion, while a near-target threshold θ_n specifies a region where further feedback is unnecessary. For torque rendering, the impedance control law employs a stiffness gain K_e and a damping gain B_e to generate consistent, physically plausible torques. Thus, $\bar{\ell}_t$ serves as a supervisory signal that guides the selection of thresholds (θ_s, θ_n) and gains (K_e, B_e), ensuring that feedback parameters are chosen in accordance with both task progress and operator workload. Building on these parameter choices, the following rule-based conditions specify how feedback is triggered under different task states.

Target Reached: If $|e_t| < \theta_s$, that is, when $\sigma_t = 1$, no feedback is delivered in order to minimize unnecessary interference and reduce the operator's cognitive load.

Near Target (Small Error): If $|e_t| < \theta_n$, no additional feedback is applied to avoid unnecessary intervention.

Far From Target: If $|e_t| \geq \theta_n$, torque feedback is applied via the impedance control law (Eq. 2), scaling with the error.

Overshoot: If $|e_{t-\Delta}| \approx 0$ and $e_t \cdot \dot{e}_t > 0$, the system is considered to have overshot the target, meaning it has crossed the goal and the error starts to increase again. In this case, torque feedback is applied to counteract the reversal.

Failure Movements: If $|e_t| \geq \theta_s$ and $\dot{e}_t = 0$, the object is considered to have exited the permissible manipulation workspace on the robot hand, either due to jamming or detachment. In such cases, a vibrotactile alert is delivered to promptly notify the operator of the failure state.

To implement the above rule-based conditions in practice, the feedback torque is generated through an impedance control law that models the operator–device interaction as a virtual spring–damper system. Specifically, the torque command is computed as

$$\tau_t^{\text{fb}} = K_e \cdot e_t + B_e \cdot \dot{e}_t \tag{2}$$

where e_t is the orientation error and $\dot{e}_t$ its rate. This ensures physically consistent feedback and stabilizes telemanipulation, while simultaneously offering intuitive operator perception and safe, bounded force delivery.

C. Feedback Hardware and PID Design

The Bi-Hap hardware system adopts a modular architecture that supports different motors and sensors for various tasks and can be extended to additional degrees of freedom due to its scalability. Fig. 3 illustrates the overall system architecture. The main modules are as follows:

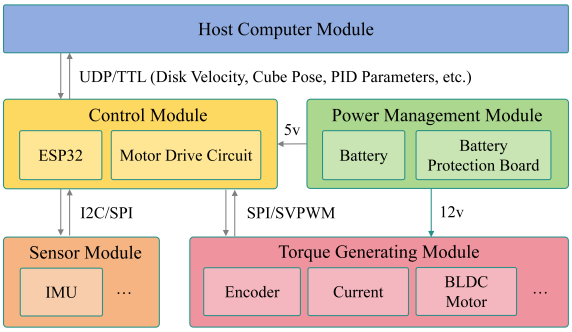


Fig. 3: System architecture of the Bi-Hap hardware, illustrating key modules for control, sensing, power, and torque feedback.

Host Computer Module: The host computer executes the robot control program and implements the error-adaptive feedback strategy. It communicates wirelessly with the controller module via Wi-Fi, receiving the object pose as the control target and transmitting the computed feedback commands for further processing.

Control Module: The ESP32 microcontroller manages communication and low-level sensor and actuator control. It executes the Field Oriented Control (FOC) algorithm [32] to regulate the motor velocity, enabling stable and precise torque generation even under dynamic conditions.

Torque Generating Module: A compact BLDC motor drives a 55 mm flywheel designed to balance weight (128 g) and rotational inertia (4.8×10^{-5} kg·m²). This configuration enables efficient momentum-based haptic feedback, while a magnetic encoder provides precise angular measurements.

Power Management Module: A 12V battery with a protection board supplies power to the motor and converts 12V to 5V for control and sensors. A power switch and DC interface allow convenient charging or direct input, ensuring safe and stable distribution.

Sensor Module: A GY-95T IMU operating at 200 Hz provides real-time motion tracking. Expansion via SPI or I2C allows integration of additional sensors, such as temperature or pressure, for customized perception.

An exploded view of the Bi-Hap hardware is shown in Fig. 4. The device weighs 320 g with the counterweight and is enclosed in a 60 mm three-layer cube. This structure protects internal components, facilitates intuitive finger manipulation, and simplifies part arrangement and replacement. The removable upper layer allows counterweight exchange, while the lower layer provides access to the battery and sensors. Such a modular design enables easy disassembly, maintenance, and future upgrades, thereby improving overall usability and serviceability.

According to the angular momentum theorem, torque generation can be directly linked to the flywheel's angular acceleration. Through this principle, torque is proportional to angular acceleration, and velocity control is adopted to regulate feedback intensity. Specifically, a PID controller adjusts the FOC phase voltage to drive the motor to the target speed:

$$u_t = K_p \cdot \varepsilon_t^{\omega} + K_i \cdot \mathcal{I}(\varepsilon_t^{\omega}) + K_d \cdot \dot{\varepsilon}_t^{\omega} \tag{3}$$

where u_t is the control input, ε_t^{ω} denotes the angular velocity

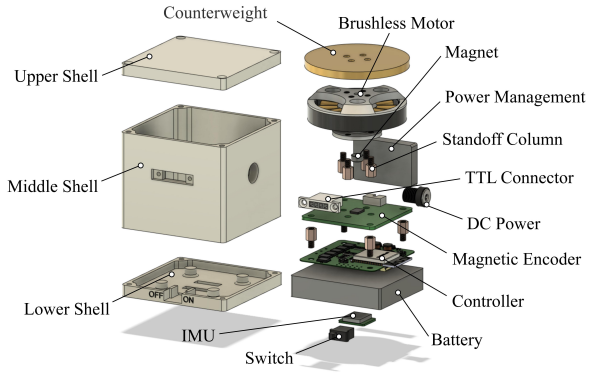
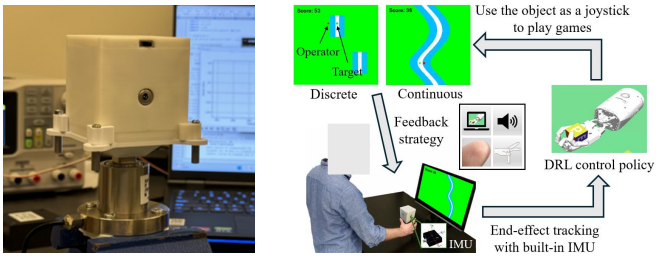


Fig. 4: Exploded view of the Bi-Hap hardware showing the three-layer cubic structure (side length = 60 mm), designed for ease of maintenance, modular component replacement, and intuitive in-hand telemanipulation.



(a) Feedback Hardware Tests (b) Telemanipulation Task Tests

Fig. 5: (a) shows the hardware testing platform, equipped with a torque sensor and a 3D printed mount. (b) shows the telemanipulation task, where the operator uses the Bi-Hap hardware as a joystick while a DRL controller drives the robotic hand to rotate a cube. The cube's orientation is mapped to the red dot on the screen, with white, blue, and green regions denoting *Target Reached*, *Near Target*, and *Far From Target*, respectively.

error (target minus actual), $\mathcal{I}(\varepsilon_t^\omega)$ is the accumulated error over time, and $\dot{\varepsilon}_t^\omega$ is the error derivative. K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively. The gains are set to $K_p = 0.2$, $K_i = 20$, and $K_d = 0$, tuned via the Ziegler–Nichols method with manual refinement to ensure fast response and steady-state accuracy.

IV. Experiments Design and Evaluation Metrics

A. Experimental Setup and Task Design

To comprehensively evaluate the performance of the Bi-Hap system, we divide the testing into three categories:

DRL Control Policy Test: The DRL control policy will be evaluated in a simulation environment to facilitate training and testing. Specifically, we adopt the Shadow hand environments from the OpenAI GYM Robotics platform, which runs on the MuJoCo physics simulator [33]. This control policy is designed to manipulate a cube placed on the palm with a random initial pose, rotating it around the Z-axis to achieve the goal pose like Fig. 2. The policy is trained with a dense reward defined as:

$$R = -2 \cdot \arccos\left(|\langle q_a, q_g \rangle|\right) + r_b \tag{4}$$

where q_a represents the object orientation and q_g represents the goal, both in quaternion form. The $\arccos$ function calculates the minimal angular distance between the current and goal orientations. This reward function provides a continuous reward signal during the manipulation process. Task success is defined as an angular distance less than 0.05 rad, with $r_b \in \{0, 1\}$ serving as a binary bonus: $r_b = 1$ when the task succeeds and $r_b = 0$ otherwise.

In the simulation environment, agents operate with a time step of 0.04 s. Based on this configuration, the policy network is implemented as a 5-layer fully connected neural network. The input vector comprises a total of 68 elements: 24 joint angular positions and 24 angular velocities of the robot, 13 object states including Cartesian position, quaternion-based orientation, as well as linear and angular velocities, and finally the end-effect goal defined by its Cartesian position and quaternion-based orientation. The output is 20 joint actuators of the hand. Most hyperparameters are from [34], but with changes to the total epoch to 200. The PC hardware for training includes an Intel 12900K, an Nvidia RTX3080ti, and 64GB of RAM.

For evaluation, two reference signals are defined, with the object's desired Z-axis rotation θ given as:

1) Sine wave goal:

$$\theta = A \sin(\omega t) \tag{5}$$

where $A = (0.25, 0.5)$ is the amplitude, $\omega = (0.1, 0.05)$ is the frequency and t is the time variable. The sinusoidal goal evaluates the policy's ability to follow smoothly varying dynamic references.

2) Square wave goal:

$$\theta = A \operatorname{sgn}\big(\sin(\omega t)\big) \tag{6}$$

where $A = (0.25, 0.5)$ is the amplitude, $\omega = 0.1$ is the frequency and t is the time variable. The square wave goal evaluates the policy's ability to follow discrete and abrupt changes in the target.

Feedback Hardware Tests: This section focuses on evaluating the torque output performance of the feedback hardware. A dedicated testing platform was constructed, equipped with a ±2 Nm ZNNT-series torque sensor and a custom 3D-printed mounting structure for secure installation and measurement (Fig. 5(a)). Operating at a 500 Hz sampling rate, the sensor accurately captures signal frequencies up to 250 Hz, as dictated by the Nyquist sampling theorem. This bandwidth encompasses the low-frequency (0–10 Hz) and mid-frequency (10–100 Hz) force components most perceptible to human touch [35]. The torque sensor provides an accuracy of 0.1%, which is well below the 1% force variation threshold detectable by human perception [36]. The torque output tests of the feedback hardware employ the same sinusoidal and square wave reference signals as those used in the DRL Control Policy Test; therefore, the equations are not repeated here. The difference lies in the parameter settings: for the sine wave signal, the amplitude is set to $A = (0.015,\ 0.030)$ and the angular frequency to $\omega = (8,\ 16)$; for the square wave signal, the amplitude is set to $A = (0.01,\ 0.02)$ and the angular frequency to $\omega = 5$.

Telemanipulation Task Tests: The specific task is conducted in a gaming environment, where the human operator holds the Bi-Hap hardware as a joystick, and a

DRL controller drives the robotic hand to rotate a cube to match the orientation of the Bi-Hap hardware. The cube's orientation in the robotic hand is simultaneously mapped to the movement of a red dot on the screen. The operator must rotate the Bi-Hap hardware to guide the robotic hand in aligning the cube with the target orientation, represented by the white region shown in the image (Fig. 5(b)). At each simulation step, one point is awarded when the red dot enters the white region, and zero points are given otherwise.

This experiment evaluates the Bi-Hap system in real human interaction through discrete and continuous scenarios. For ablation, operators also used a device of identical size and weight but without torque feedback. The controller employed stiffness $K_e = 2.5$, damping $B_e = 1$, and thresholds $\theta_s = 0.028$ rad and $\theta_n = 0.056$ rad. Five participants with prior telemanipulation experience were recruited under university IRB approval, each completing five trials across four conditions: discrete vs. continuous tasks, with and without torque feedback.

B. Evaluation Metrics

The following metrics are designed for the sine wave goal:
RMSE: The root mean square error is calculated as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(v_{m,i} - v_{d,i})^2} \tag{7}$$

where n is the number of timesteps, $v_{m,i}$ is the measured value at the i-th sampling, and $v_{d,i}$ is the desire value at the i-th sampling. RMSE evaluates the overall torque feedback accuracy during the testing process.
Average Latency: The average latency is calculated as:

$$L = \frac{1}{n}\sum_{i=1}^{n}\frac{\varphi_m - \varphi_d}{2\pi f} \tag{8}$$

where φ_m is the phase of the measured value, φ_d is the phase of the desired value. Both phases are obtained from the dominant frequency after the Fourier transformation. The average latency evaluates the system's real-time capability.

The following metrics are designed for the square wave:
RMSE: This metric is calculated the same as the sine wave.
Overshoot: We calculate the max-percent overshoot:

$$OS = \frac{p_m - p_d}{p_d} \times 100\% \tag{9}$$

where p_m denotes the measured peak value, and p_d is the desired peak value, equal to the square wave amplitude A. This metric reflects the aggressiveness of the system in response to abrupt torque changes.
Rise Time: The rise time t_r taken for the response to rise from 0% to 90% of its final value. It reflects how quickly the system follows target commands.

For the evaluation metrics, the test value corresponds to the object's rotation angle about the Z-axis in the DRL control policy tests, whereas it represents the torque about the Z-axis in the feedback hardware tests. In all cases, lower values indicate better performance.

The following metrics are designed for the telemanipulation task tests:
Fit Count: We record the number of times the target point is moved into the designated area in each experiment as a measure of overall performance. The total number of target completions is denoted as FC :

$$FC = \sum_{i=1}^{n} S_i \tag{10}$$

where n represents the total number of timesteps. The target angle must be reached within a $\pm 1.6^\circ$ range and held for 1.6 seconds. During each refresh, the system checks if the object's angle falls within the target range. If it does, $S_i = 1$; if not, $S_i = 0$, where i represent the refresh count.
Statistical Significance: We further assessed the statistical significance of performance differences between feedback and no-feedback conditions using a linear mixed-effects model. In this model, BiHap (On vs. Off) and Scene (Discrete vs. Continuous) were included as fixed effects, while Subject was modeled as a random intercept to account for repeated measures within each participant. Statistical significance was evaluated at $\alpha = 0.05$.

V. Results and Discussion

A. DRL Control Policy Performance

Table I shows that the DRL policy tracks sine waves accurately, with RMSE <0.04 rad and latency <0.04 s at low amplitudes and frequencies. Errors increase with amplitude and frequency, reflecting higher torque demands and phase lag, but overall tracking remains stable. Table II highlights the challenge of square wave inputs. Larger amplitudes yield higher RMSE (0.36 rad) and overshoot (9.3%), with longer rise times as instantaneous target jumps force repeated contact-point relocation within the limited finger workspace. Despite these effects, the policy remains stable, indicating robustness to discontinuous commands.

Fig. 6 further illustrates these behaviors by directly comparing the measurements with the reference trajectories. In sine wave tracking, the policy follows smooth variations with only minor deviations, arising when the restricted finger workspace necessitates frequent repositioning of contact points to sustain cube rotation. For square waves, overshoot is visible but corrected quickly, confirming effective adaptation to abrupt transitions. Overall, the end-effect-oriented DRL controller balances accuracy and stability, providing reliable real-time tracking even under high-complexity inputs.

B. Feedback Hardware Performance

Table IV shows the torque feedback performance under sine wave references. The device maintains high accuracy and low latency at low amplitude and frequency, while error and delay increase with larger torques and faster oscillations. Notably, latency does not increase uniformly with task difficulty, reflecting the combined effects of amplitude and frequency. Fig. 8(a) shows the Bi-Hap hardware torque

TABLE I: DRL Control Performance (Sine Wave, 20 Tests)

A (rad)	ω	RMSE (rad)	L (s)
0.25	0.1	0.0650	0.0629
0.25	0.05	0.0370	0.0373
0.50	0.1	0.1218	0.0905
0.50	0.05	0.0733	0.0698

TABLE II: DRL Control Performance (Square Wave, 20 Tests)

A (rad)	ω	RMSE (rad)	OS (%)	t_r (s)
0.25	0.1	0.1357	2.1047	0.2197
0.50	0.1	0.3651	9.2565	0.5421

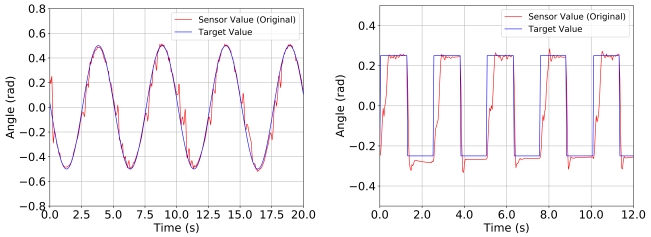


Fig. 6: DRL performance for (a) sine and (b) square wave goal.

closely tracks the sinusoid without overshoot, confirming real-time precision and responsiveness.

Table V reports performance under square wave inputs, which are more challenging due to discontinuous torque transitions. At small amplitude, the system achieves low error but shows pronounced overshoot, as seen in Fig. 8(b), where the red trajectory briefly exceeds the target after each step. This reflects the difficulty of sharp switching. As amplitude grows, overshoot is reduced while error and rise time increase, indicating the device favors stable convergence over aggressive tracking. The waveform shows rounded edges at transitions, suggesting the controller effectively limits instability. These results highlight a trade-off between responsiveness and stability under discontinuous inputs.

C. Telemanipulation Task Performance

Table III summarizes the telemanipulation task results, where A–E denotes the five subjects. Overall, the Bi-Hap system

TABLE III: Real-world Testing in a Tele-played Game (5 Tests)

Scene	Bi-Hap	A	B	C	D	E	Average
Discrete	On	573	597	593	569	602	587
	Off	577	619	558	494	584	566
Continuous	On	441	488	558	506	529	504
	Off	378	482	518	424	552	471

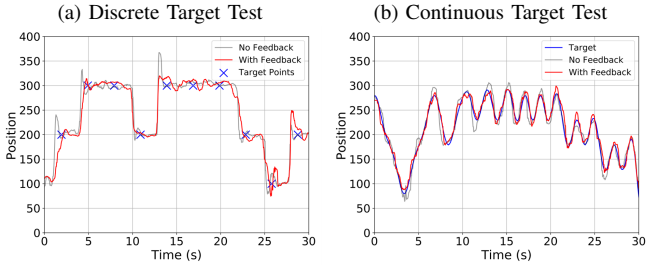


Fig. 7: Real-world telemanipulation test performance of (a) discrete and (b) continuous target.

TABLE IV: Feedback Hardware Performance (Sine Wave, 5 Tests)

A (N·m)	ω	RMSE (N·m)	L (s)
0.0150	8	0.0021	0.0126
0.0150	16	0.0058	0.0239
0.0300	8	0.0081	0.0253
0.0300	16	0.0095	0.0174

TABLE V: Feedback Hardware Performance (Square Wave, 5 Tests)

A (N·m)	ω	RMSE (N·m)	OS (%)	t_r (s)
0.0100	5	0.0043	27.38	0.0979
0.0200	5	0.0140	11.16	0.2262

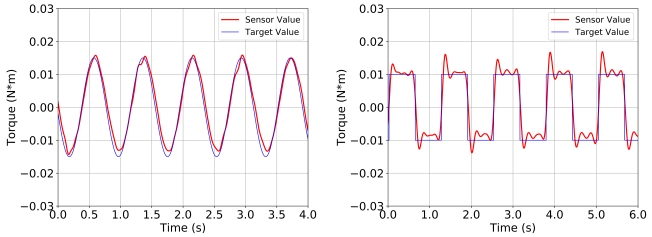


Fig. 8: Feedback performance of (a) sine and (b) square wave goal.

with torque feedback outperforms the condition of no torque feedback. In the discrete task, torque feedback improves average scores by 21 points, while in the continuous task, the improvement is 33 points. While individual gains vary, this variability stems from differences in tracking strategies and rotational habits. Nonetheless, the results indicate that torque feedback enhances task performance across most subjects and conditions, with only three exceptions out of ten cases. The mixed-effects model confirmed a significant effect of torque feedback (on vs. off) ($\beta = 26.98 \pm 12.46$, $p = 0.030$). To further demonstrate Bi-Hap's performance, we recorded a subject's tracking behavior in discrete and continuous angle tasks. As shown in Fig. 7, torque feedback (red) not only improved tracking accuracy but also reduced overshoot compared to the no-feedback condition (grey). Feedback provided reliable cues during large errors or sudden reversals, preventing premature or exaggerated movements. Consequently, the subject aligned more closely with the target (blue), corrected deviations steadily, and achieved smoother transitions.

In summary, the Bi-Hap system demonstrates its capability to enhance dexterous telemanipulation by extracting the human's interaction command and delivering low-latency, accurate torque and vibration feedback through a compact, momentum-based design. Experimental results confirm improvements in accuracy, responsiveness, and error correction across multiple task types and subjects. This work validates the feasibility of employing a palm-sized momentum-based torque device for immersive telemanipulation. Future work will focus on extending the system to support multi-DoF torque feedback, enabling the manipulation of asymmetric objects and more complex rotational motions. In addition, we plan to integrate the proposed framework with physical robotic hands and conduct larger-scale user studies to further validate its performance.

REFERENCES

[1] Y. Huang, D. Fan, D. Yan, W. Qi, G. Deng, Z. Shao, Y. Luo, D. Li, Z. Wang, Q. Liu *et al.*, "Human-robot collaborative tele-grasping in clutter with five-fingered robotic hands," *IEEE Robotics and Automation Letters*, 2025.

[2] R. Rätz, F. Conti, I. Thaler, R. M. Müri, and L. Marchal-Crespo, "Enhancing stroke rehabilitation with whole-hand haptic rendering: development and clinical usability evaluation of a novel upper-limb rehabilitation device," *Journal of NeuroEngineering and Rehabilitation*, vol. 21, no. 1, p. 172, 2024.

[3] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, "Dexcap: Scalable and portable mocap data collection system for dexterous manipulation," *arXiv preprint arXiv:2403.07788*, 2024.

[4] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[5] A. Bhatt, A. Sieler, S. Puhlmann, and O. Brock, "Surprisingly robust in-hand manipulation: An empirical study," *arXiv preprint arXiv:2201.11503*, 2022.

[6] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox, "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9164–9170.

[7] H. Zhang, S. Hu, Z. Yuan, and H. Xu, "Doglove: Dexterous manipulation with a low-cost open-source haptic force feedback glove," *arXiv preprint arXiv:2502.07730*, 2025.

[8] A. C. Abad, D. Reid, and A. Ranasinghe, "A novel untethered hand wearable with fine-grained cutaneous haptic feedback," *Sensors*, vol. 22, no. 5, p. 1924, 2022.

[9] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General in-hand object rotation with vision and touch," in *Conference on Robot Learning*. PMLR, 2023, pp. 2549–2564.

[10] J. Pitz, L. Röstel, L. Sievers, and B. Bäuml, "Dextrous tactile in-hand manipulation using a modular reinforcement learning architecture," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2023. [Online]. Available: http://dx.doi.org/10.1109/ICRA48891.2023.10160756

[11] H. Wang, H. Bai, X. Zhang, Y. Jung, M. Bowman, and L. Tao, "Real-time dexterous telemanipulation with an end-effect-oriented learning-based approach," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 164–12 169.

[12] L. Liu, S. Miyake, N. Maruyama, K. Akahane, and M. Sato, "Development of two-handed multi-finger haptic interface spidar-10," in *Haptics: Neuroscience, Devices, Modeling, and Applications: 9th International Conference, EuroHaptics 2014, Versailles, France, June 24-26, 2014, Proceedings, Part II 9*. Springer, 2014, pp. 176–183.

[13] C. Meijneke, B. Sterke, G. Hermans, W. Gregoor, H. Vallery, and D. Lemus, "Design and evaluation of pint-sized gyroscopic actuators," in *2021 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2021, pp. 454–461.

[14] J. L. Berna Moya, A. van Oosterhout, M. T. Marshall, and D. Martinez Plasencia, "Hapticwhirl, a flywheel-gimbal handheld haptic controller for exploring multimodal haptic feedback," *Sensors*, vol. 24, no. 3, p. 935, 2024.

[15] T. Hashimoto, S. Yoshida, and T. Narumi, "Metamorphx: An ungrounded 3-dof moment display that changes its physical properties through rotational impedance control," in *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, 2022, pp. 1–14.

[16] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[17] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas *et al.*, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

[18] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam *et al.*, "Dextreme: Transfer of agile in-hand manipulation from simulation to reality," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5977–5984.

[19] A. Kannan, K. Shaw, S. Bahl, P. Mannam, and D. Pathak, "Deft: Dexterous fine-tuning for real-world hand policies," *arXiv preprint arXiv:2310.19797*, 2023.

[20] Z. Si, K. L. Zhang, Z. Temel, and O. Kroemer, "Tilde: Teleoperation for dexterous in-hand manipulation learning with a deltahand," *arXiv preprint arXiv:2405.18804*, 2024.

[21] C. Pacchierotti and D. Prattichizzo, "Cutaneous/tactile haptic feedback in robotic teleoperation: Motivation, survey, and perspectives," *IEEE Transactions on Robotics*, vol. 40, pp. 978–998, 2023.

[22] M. Raitor, C. M. Nunez, P. J. Stolka, A. M. Okamura, and H. Culbertson, "Design and evaluation of haptic guidance in ultrasound-based needle-insertion procedures," *IEEE Transactions on Biomedical Engineering*, vol. 71, no. 1, pp. 26–35, 2023.

[23] Z. F. Quek, W. R. Provancher, and A. M. Okamura, "Evaluation of skin deformation tactile feedback for teleoperated surgical tasks," *IEEE transactions on haptics*, vol. 12, no. 2, pp. 102–113, 2018.

[24] M. Sagardia, T. Hulin, K. Hertkorn, P. Kremer, and S. Schätzle, "A platform for bimanual virtual assembly training with haptic feedback in large multi-object environments," in *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, 2016, pp. 153–162.

[25] K. Krieger, Y. De Pra, H. Ritter, and A. Moringen, "Motion analysis of upper limb and hand in a haptic rotation task," *arXiv preprint arXiv:2411.12765*, 2024.

[26] J. M. Walker, H. Culbertson, M. Raitor, and A. M. Okamura, "Haptic orientation guidance using two parallel double-gimbal control moment gyroscopes," *IEEE transactions on haptics*, vol. 11, no. 2, pp. 267–278, 2017.

[27] W. Y. Yang, Y. Zou, J. Huang, R. Abujaber, and K. Nakagaki, "Torquecapsules: Fully-encapsulated flywheel actuation modules for designing and prototyping movement-based and kinesthetic interaction," in *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, 2024, pp. 1–15.

[28] S. Verret, T. Laliberté, R. Cloutier, and C. Gosselin, "Synthesis, dynamic modeling, prototyping and control of a handheld rotational inertia generator," *IEEE Transactions on Haptics*, vol. 17, no. 4, pp. 591–603, 2024.

[29] J. Wang, Y. Yuan, H. Che, H. Qi, Y. Ma, J. Malik, and X. Wang, "Lessons from learning to spin "pens"," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 06–09 Nov 2025, pp. 3124–3138. [Online]. Available: https://proceedings.mlr.press/v270/wang25j.html

[30] X. Xiang and S. Foo, "Recent advances in deep reinforcement learning applications for solving partially observable markov decision processes (pomdp) problems: Part 1—fundamentals and applications in games, robotics and natural language processing," *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 554–581, 2021.

[31] M. Chen, S. Nikolaidis, H. Soh, D. Hsu, and S. Srinivasa, "Planning with trust for human-robot collaboration," in *Proceedings of the 2018 ACM/IEEE international conference on human-robot interaction*, 2018, pp. 307–315.

[32] A. Skuric, H. S. Bank, R. Unger, O. Williams, and D. González-Reyes, "Simplefoc: a field oriented control (foc) library for controlling brushless direct current (bldc) and stepper motors," *Journal of Open Source Software*, vol. 7, no. 74, p. 4232, 2022.

[33] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[34] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder *et al.*, "Multi-goal reinforcement learning: Challenging robotics environments and request for research," *arXiv preprint arXiv:1802.09464*, 2018.

[35] N. Bai, Y. Xue, S. Chen, L. Shi, J. Shi, Y. Zhang, X. Hou, Y. Cheng, K. Huang, W. Wang *et al.*, "A robotic sensory system with high spatiotemporal resolution for texture recognition," *Nature Communications*, vol. 14, no. 1, p. 7121, 2023.

[36] Y. Wang, H. Wu, T. Li, J. Wang, Z. Wei, and H. Wang, "Toward human-like touch sense via a bioinspired soft finger with self-decoupled bending and force sensing," *Cell Reports Physical Science*, vol. 5, no. 10, 2024.