

A VR-based Robotic Teleoperation System with Haptic Feedback and Adaptive Collision Avoidance

Fan Wu, Ziyuan Jiao, *Member, IEEE*, Wanlin Li, *Member, IEEE*, Zeyu Zhang, *Member, IEEE*,
Hang Li, Jiahao Wu, Baoxiong Jia, Shaopeng Dong

Abstract—Robotic teleoperation systems enable humans to control robots remotely. Recent advancements in VR have transformed teleoperation into immersive, intuitive platforms that improve human-machine synergy. However, existing VR-based teleoperation systems face challenges such as under-informed shared control, limited tactile feedback, and computational inefficiencies, which hinder their effectiveness in complex, cluttered scenarios. This paper introduces a novel VR-based teleoperation system incorporating haptic feedback and adaptive collision avoidance. The system tracks human hand trajectories via an VR-based handheld device and provides tactile feedback from a robotic gripper. Through the proposed tele-MPPI method, the system anticipates robot motions, adaptively adjusts trajectories to avoid obstacles, and maintains computational efficiency, enabling real-time operation at 20 Hz. Simulations and experiments demonstrate the system’s ability to grasp delicate objects without damage and to navigate cluttered environments by mitigating collision risks.

Index Terms—Teleoperation, virtual reality, human-machine interface, inverse kinematics.

I. INTRODUCTION

TELEOPERATION systems are a crucial form of Human-Machine Interaction (HMI), facilitating communication and control between humans and remote machines or virtual entities. Over the years, robotic teleoperation has evolved from basic remote control to advanced, immersive systems where human operators interact with intelligent robots and virtual agents in complex, simulated environments. This evolution has expanded the scope of teleoperation systems, blending human decision-making with machine precision for improved synergy: they function as interactive platforms for cognitive engagement, training, and skill transfer between humans and intelligent agents [1–4]. These systems have become indispensable across diverse fields, including industrial automation, healthcare, and remote exploration, by allowing precise and efficient control over physical or digital entities from a distance.

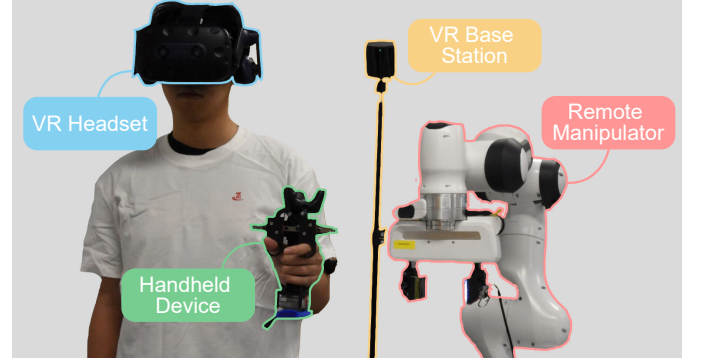
The advantage of Virtual Reality (VR)-based teleoperation systems lies in their ability to provide a fully immersive

This work was supported in part by the National Natural Science Foundation of China (Grant No.52305007), and by the State Key Laboratory of Mechanical System and Vibration (Grant No. MSV202519).

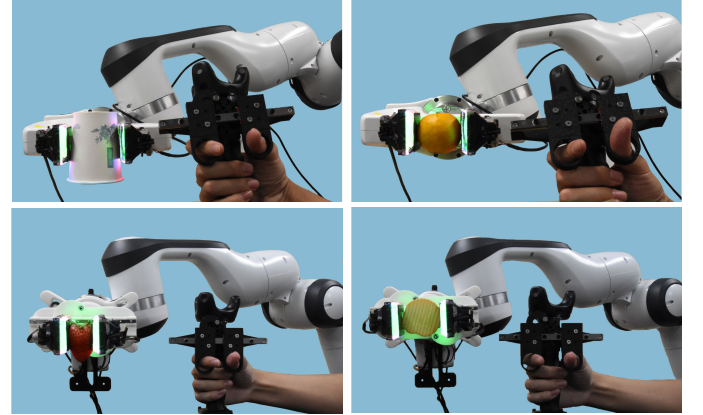
Fan Wu, Ziyuan Jiao, Wanlin Li, and Zeyu Zhang contributed equally to this work. *Corresponding authors: Ziyuan Jiao and Wanlin Li (emails: jiaoziyuan@bigai.ai; liwanlin@bigai.ai).*

Fan Wu and Shaopeng Dong are with the Department of Automation, Beihang University, Beijing 100191, China. This work was done during Fan Wu’s internship at Beijing Institute for General Artificial Intelligence (BIGAI).

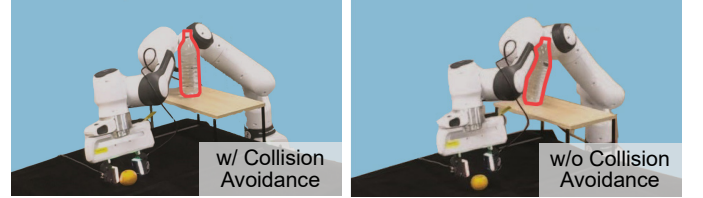
Ziyuan Jiao, Wanlin Li, Zeyu Zhang, Hang Li, Jiahao Wu, and Baoxiong Jia are with State Key Laboratory of General Artificial Intelligence, BIGAI, Beijing 100080, China.



(a) VR-based teleoperation system.



(b) Grasping with haptic feedback.



(c) Adaptive collision avoidance.

Fig. 1: A VR-based robotic teleoperation system with haptic feedback and adaptive collision avoidance.

virtual environment, allowing operators to interact intuitively with avatars, robots, or virtual tools, creating a natural and responsive experience. By integrating real-time data transmission and motion control, VR systems offer a heightened sense of presence and precision in manipulating virtual or remote objects. Moreover, VR environments can be adapted to accommodate various scenarios, such as different robot configurations or environmental setups, offering improved flexibility and adaptability. Combined with handheld tracking devices, VR-based teleoperation systems further extend the operator’s

interactive awareness by incorporating sensory feedback, such as vibrations [5, 6]. Compared to exoskeleton-based systems, VR-based teleoperation offers greater flexibility, immersion, and reduced physical strain. Additionally, it overcomes the limitations of vision-based teleoperation, including sensitivity to lighting variations or potential occlusions, which can lead to recognition loss and reduced operational precision.

Nonetheless, VR-based robotic teleoperation faces several challenges. First, VR handheld devices inherently introduce an under-informed shared control problem [7], where multiple possible Inverse Kinematics (IK) solutions exist for achieving the same end-effector pose. Some solutions may adversely impact subsequent operations, leading to singularities or collisions with the surrounding environment. And exhaustively searching through all possible robot configurations would significantly increase computational complexity. Additionally, current VR handheld devices provide limited perceptive feedback, particularly lacking haptic feedback compatible with robotic systems—such as the ability to convey grasping forces from a parallel gripper—which restricts the operator’s tactile perception. This shortcoming makes grasping delicate objects particularly challenging. In summary, these limitations reduce the effectiveness of existing teleoperation systems, particularly in successfully performing sophisticated robotic tasks in cluttered or dynamic environments.

This work presents a novel real-time VR-based teleoperation system with haptic feedback and adaptive collision avoidance, as is shown in Fig. 1. The system controls the robot’s end-effector by tracking human hand trajectories captured through a VR-based handheld device, while simultaneously providing real-time feedback of grasping forces measured from the tactile sensor mounted on the robotic parallel gripper to the handheld device. By integrating Model Predictive Path Integral (MPPI) control and IK generative model, the system anticipates future motion during trajectory selection, resulting in a highly efficient teleoperation system that can adaptively avoid obstacles. In a series of simulations and experiments, we demonstrate the effectiveness of our teleoperation system. The operator successfully grasps delicate objects such as paper cup, strawberry, and potato chip without damage, showcasing the system’s precision and haptic feedback capabilities. Additionally, within cluttered environments, we show that even when the captured human hand trajectory leads to potential collisions with environmental obstacles, the proposed method actively adjusts the robot’s trajectory to avoid collisions while maintaining accurate tracking.

The main contributions of this work are three-fold:

- We present an integrated haptic feedback system that enhances the perception of grasping forces, enabling operators to grasp delicate objects more effectively.
- We propose a tele-MPPI framework that combines MPPI control with a generative IK model to predict motion trajectories and dynamically adjust robot paths, ensuring collision-free operation in cluttered environments.
- We develop a comprehensive VR-based teleoperation system that integrates haptic feedback and real-time trajectory tracking of a robotic gripper for precise and safe manipulation in complex environments.

II. RELATED WORK

A. Teleoperation Systems

The applications of robotic teleoperation have evolved significantly, expanding from their early use in handling radioactive materials [8] to a diverse range of modern domains [9]. Current teleoperation methods are broadly categorized into exoskeleton-based, vision-based, and VR-based approaches, depending on the motion capture devices.

Exoskeleton-based Teleoperation Systems, rooted in traditional teleoperation principles, have a long history of development [9]. Renowned for their high precision and real-time performance, they are particularly favored in medical applications. These systems directly capture motion signals from meticulously designed devices, enabling straightforward mapping to robotic systems [10, 11]. However, their design is often expensive, bulky, complex to maintain, and task-specific, limiting versatility and adaptability across different robots.

Vision-based Teleoperation Systems have emerged as a mainstream method, driven by significant advancements in computer vision [12]. By eliminating the need for wearable equipment, this approach significantly reduces operator burden while enhancing flexibility and convenience. However, it is highly sensitive to lighting conditions, susceptible to occlusions, and lacks somatosensory feedback, which undermines motion recognition and operational accuracy.

VR-based Teleoperation Systems have reshaped HMI, introducing immersive experiences and improved interactivity essential for teleoperation. Most VR-based systems utilize commercial VR headsets, combining real-time data acquisition, ease of use, and reliable implementation [13]. By leveraging technologies such as vision, infrared, and IMUs, VR-based systems address many of the limitations of methods solely based on vision while avoiding the unwieldy devices of exoskeleton-based systems. Building upon these VR interfaces, we propose a teleoperation system that enhances further their capabilities with integrated haptic feedback and real-time motion re-targeting for precise manipulation with automatic obstacle avoidance in complex environments.

B. Teleoperation Interface

The user interface in VR-based teleoperation is crucial for delivering sensory feedback to operators, such as visual, auditory, and tactile cues. Visual feedback provides immersive real-time graphics, enabling users to monitor interactions between the robot and its environment while enhancing spatial awareness and decision-making efficiency [14, 15]. Auditory feedback, such as ambient sounds and collision alerts, helps operators interpret changes and operational states through audio signals [16]. Tactile feedback, however, remains the most challenging to implement. Most VR controllers rely on basic vibration feedback [6] and are not optimized for robot-specific tasks like replicating gripper actions. While haptic gloves [4] provide more sophisticated feedback mechanisms, their high cost and limited accessibility hinder widespread adoption. To address these limitations, our work introduces a specialized handheld device designed to deliver effective haptic feedback during object teleoperation.

C. Robot Motion Retargeting

Robot motion retargeting involves translating human motion signals into corresponding robotic commands, aiming to minimize discrepancies between human intent and the robot's movements while adhering to constraints such as physical limitations, environmental obstacles, and the robot's structural characteristics. The simplest method involves directly mapping human joints to robot joints [17]. While straightforward, this approach lacks flexibility and requires complex hand-tuning for different robot configurations. Model-based optimization [18, 19] can theoretically achieve optimal retargeting results but often involves lengthy iterations and the risk of converging to local optima due to poor initial conditions [20]. Learning-based approaches train networks to meet specific retargeting requirements, offering adaptability and flexibility [21, 22]. However, they are data-dependent and lack transferability to new robots. IK approaches [23–25], on the other hand, are highly versatile across different robotic arms and can generate real-time joint configurations that precisely follow end-effector trajectories. In practice, robotic arms often feature redundant Degree of Freedoms (DoFs). Selecting an inappropriate robot configuration can result in unexpected outcomes, including collisions with the environment. In this article, we propose a tele-MPPI framework that combines MPPI control with a generative IK solver to predict motion trajectories and dynamically select optimal configurations in a real-time manner, ensuring collision-free teleoperations.

III. THE TELEOPERATION DEVICE

This section outlines the hardware design of our handheld interface with haptic feedback, which was developed for compatibility with standard robotic systems. Our goal is to develop a portable handheld teleoperation interface, see Fig. 2, that facilitates immersive interaction with robotic systems. The design is guided by the following criteria: *i) Lightweight and Portable*: A compact and lightweight design features mobility and ease of use, providing greater flexibility while minimizing user fatigue during operations. *ii) User-Friendly Haptic Feedback*: The ergonomically designed interface provides tactile feedback, enabling precise manipulation, particularly in teleoperation tasks involving delicate objects, thereby improving efficiency and responsiveness in complex operations. *iii) Easy-to-Fabricate*: The design emphasizes simplicity in manufacturing and assembly, reducing setup and deployment time. This ensures quick and seamless integration into different teleoperation systems.

A. Hardware

The internal structure of the handheld teleoperation interface is illustrated in Fig. 2. The device features a VR tracker mounted at the top, which wirelessly transmits hand position data to a computer at a maximum frequency of 86.9 Hz. Embedded within the handle is a servo motor, coupled with a rotor and slider mechanism, to measure the distance between the user's thumb and index finger via servo position values. A microcontroller unit, integrated into the handle, communicates with the computer via USB at a maximum baud rate of

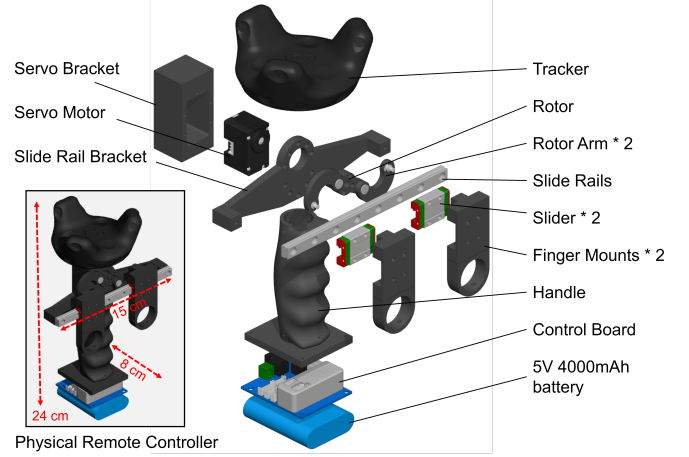


Fig. 2: Hardware design of the handheld device. The exploded view illustrates each component and its installation.

6 Mbps. Power is provided by a 5V 4000mAh lithium battery housed at the base, enabling up to 7.5 hours of continuous operation. The device measures 21 cm in length, weighs 400 g, and balances functionality with portability for extended use.

B. Haptic Feedback

The handheld device integrates bidirectional control and haptic feedback to facilitate intuitive teleoperation with collision avoidance capabilities. On the handheld device side, operators can actively control the robotic system by moving the handle and slider. The tracker captures the device's 6D pose information, while the motor encoder provides positional data for the robotic gripper's opening and closing movements. These inputs are transmitted to the robotic system, enabling precise control of the end-effector's 6D pose and gripper operation. The 6D pose data is also utilized for robot motion retargeting and collision avoidance, as discussed in Sec. IV-A. On the feedback side, a Vision-based Tactile Sensor (VBTS) [26, 27], is integrated into the robotic system to deliver tactile feedback. The VBTS consists of a soft gel layer that adapts to surface details upon contact and a camera system that records the gel's deformations. Image data is processed using photometric stereo techniques to derive force information, which is transmitted back to the user. The handheld device uses the motor under current control mode to generate resistance on the slider by reversing its rotation in response to the sensed forces, delivering haptic feedback as:

$$\frac{I_{rev}}{I_{max}} = k \frac{F}{F_{max}}. \quad (1)$$

where I_{rev} , I_{max} , F , F_{max} represent the goal current, maximum current of the motor, sensed force, maximum force of the sensor, and k is the scaling factor. To avoid unnecessary influence on the operator caused by sensor noise when there is no actual contact with an object, a feedback threshold is set. When the calculated current ratio falls below this threshold (set to 0.05 by default), the output is suppressed to zero. This integration improves the realism of interaction, effectively bridging the user's actions with the remote robotic system.

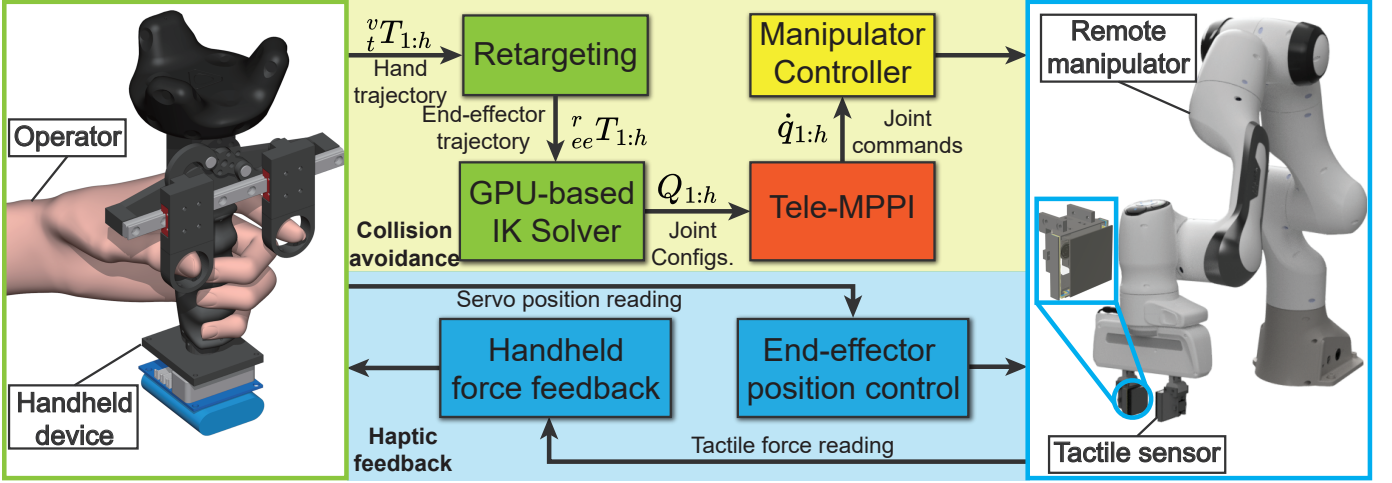


Fig. 3: The overview of the proposed VR-based robotic teleoperation system with haptic feedback and adaptive collision avoidance.

IV. THE TELEOPERATION SYSTEM

The overview of the proposed teleoperation system is illustrated in Fig. 3. Building upon the tactile feedback handheld device, we devise a collision avoidance module that processes human hand trajectories through three key stages to generate collision-free robot motions. First, a calibrated retargeting system maps the handheld device to the robot coordinate system, enabling intuitive motion transfer. Then, the retargeted poses feed into a GPU-accelerated IK solver, which efficiently generates diverse inverse kinematic solutions while maintaining precision through post-optimization refinement. Finally, our tele-MPPI framework processes these solutions in parallel to generate smooth, collision-free trajectories by optimizing joint accelerations with respect to end-effector tracking accuracy and constraint satisfaction. This integrated approach enables simultaneous real-time force feedback through the VBTS and safe robot end-effector trajectories, allowing users to achieve precise control with both tactile awareness and collision avoidance during operation.

A. Retargeting

To capture the motion of the handheld device, we employ two VR base stations and a tracker mounted on the device. Prior to using the teleoperation system, calibration is performed to estimate the handheld device's workspace for motion retargeting. During calibration, the operator holds the device, raises their arm to a front-raised position, and maintains the pose for 5 seconds. While the posture may vary depending on whether the operator is seated, standing, or performing specific tasks, the procedure remains consistent.

The position readings x_{cali}^v , y_{cali}^v , and z_{cali}^v from the tracker are used to map the handheld device's pose in the VR coordinate system to the robot manipulator's coordinate system:

$${}^v x_r = x_{\text{cali}}^v - \frac{x_{\text{ref}}^r}{a}, \quad (2)$$

$${}^v y_r = y_{\text{cali}}^v - \frac{y_{\text{ref}}^r}{b}, \quad (3)$$

$${}^v z_r = z_{\text{cali}}^v - \frac{z_{\text{ref}}^r}{c}. \quad (4)$$

where x_{ref}^r , y_{ref}^r , z_{ref}^r represent the robotic arm's end-effector home position, and a , b , and c are scaling parameters to account for workspace differences between the VR tracker and the robotic manipulator. This transformation ensures effective alignment between the handheld device and the robot's workspaces, enabling intuitive motion retargeting.

Once calibrated, motion retargeting translates tracker positions into corresponding end-effector positions. Given the tracker's positional readings (x_t^v , y_t^v , z_t^v) in the VR frame, the retargeting transformation is expressed as:

$$x_{ee}^r = a \cdot (x_t^v - {}^v x_r), \quad (5)$$

$$y_{ee}^r = b \cdot (y_t^v - {}^v y_r), \quad (6)$$

$$z_{ee}^r = c \cdot (z_t^v - {}^v z_r). \quad (7)$$

where the scaling parameters a , b , and c adaptively scale and translate human motion to align with the robotic arm's workspace. Additionally, the handheld device's orientation is directly mapped to the robot end effector's orientation, as they share the same during initialization. This ensures that the operator's gestures intuitively translate into precise robotic movements. Overall, this setup facilitates seamless interaction between the operator and the robotic system.

B. GPU-based IK Solver

Real-time teleoperation requires efficient generation of feasible robot trajectories that accurately track human motion. A key challenge in this process is obtaining diverse IK solutions to determine appropriate robot configurations that avoid environment collisions while tracking desired end-effector poses. Traditional numerical IK solvers, despite their precision, prove computationally prohibitive for real-time applications, particularly when diverse solutions are needed at scale.

To address this computational bottleneck, we adopt a generative model based on conditional normalizing flows [28]. This model transforms the IK solution process into a sampling problem from a learned conditional distribution of the solution space, parameterized by the desired end-effector pose. Thanks to the parallel nature of the neural network, our approach can generate approximately a batch of 2000 diverse feasible joint configurations in a single forward pass within 10ms.

While computationally efficient, the sampled solutions exhibit accuracy limitations with position and orientation deviations of 12 mm and 3 degrees respectively. We resolve this through a hybrid approach that combines the generative model with a GPU-accelerated optimizer [29] to refine the sampled solutions. By using the sampled solutions as seeds for the optimizer, we achieve remarkable precision with positional accuracy within 0.01mm and rotational accuracy within 0.05 degrees. This hybrid methodology processes 2000 high-precision solutions within 40ms on average, successfully balancing computational efficiency with the accuracy demands of real-time trajectory generation.

C. Tele-MPPI

Traditional optimization methods face significant challenges in converting IK solutions into feasible robot trajectories, particularly in three key areas: time efficiency, trajectory smoothness, and obstacle avoidance. Real-time teleoperation requires rapid processing of numerous IK solutions while maintaining smooth trajectories to prevent jerky motions that could compromise control stability. Additionally, generating collision-free paths further compounds these challenges.

To address these limitations, we propose tele-MPPI which leverages the MPPI control in our teleoperation system. Tele-MPPI introduces noise into joint acceleration, causing joint trajectories to diffuse over time. Through importance sampling in the joint space, the algorithm identifies optimal trajectories by minimizing a predefined cost function. This approach effectively combines filtering infeasible IKs solutions, generating collision-free paths, and smoothing whole trajectories.

The complete tele-MPPI algorithm is presented in Alg. 1. Given a sequence of tracker poses ${}^vT_{1:1+h}$ over an observation horizon h , these poses are retargeted to the robotic end-effector poses ${}^r_{ee}T_{1:1+h}$. A GPU-based IK solver computes a set of possible robot joint states $Q_{1:1+h}^r$, which serve as input for tele-MPPI. Next, the *GreedySearch*(\cdot) function selects M candidate trajectories from $Q_{1:1+h}^r$, prioritizing trajectories with minimal consecutive time-step distances. This ensures the selection of feasible and efficient trajectories in real-time.

In the standard MPPI algorithm, directly adding noise to control inputs can cause trajectory discontinuities. To mitigate this issue, our proposed tele-MPPI method instead applies noise to joint accelerations, while maintaining joint positions and velocities as state variables. This approach ensures smoother trajectories while sampling initial joint accelerations from previously generated paths to incorporate diverse IK solutions. The perturbed control input is given by $\tilde{u} = \ddot{q} + \epsilon$ where \tilde{u} represents the control input with noise, and the noise is sampled from a normal distribution $\mathcal{N}(0, \sigma^2)$.

The tele-MPPI algorithm employs a comprehensive cost function to generate smooth, collision-free trajectories. The total cost for each rollout is defined as:

$$C_{\text{total}} = C_{\text{smooth}} + \sum_{t=1}^h C_{\text{run}} + \gamma C_{\text{pert}}, \quad (8)$$

where C_{smooth} ensures trajectory continuity by penalizing large joint movements, C_{run} combines end-effector tracking accu-

Algorithm 1: Tele-MPPI

Input : $Q_{1:h}^r$: Robot joint configurations over the observation horizon h
F: System transition model
Output : $\dot{q}_{1:c}^*$: Joint velocity commands
Params : M : No. of candidate trajectories
 E : No. of control noise vectors
 σ : Noise standard deviation
 Σ : Covariance matrix between added noises

```

1 // Get current robot state
2  $q_0, \dot{q}_0, \ddot{q}_0 \leftarrow \text{robot.getState}()$ 
3 // Greedy search for  $M$  trajectories
4  $\{q_{1:h}^m\}_M \leftarrow \text{GreedySearch}(Q_{1:h}^r)$ 
5  $U \leftarrow \emptyset$ 
6 // Iterate over  $M$  trajectories
7 for  $m = 1, 2, \dots, M$  do
8    $\dot{q}_{1:h}^m, \ddot{q}_{1:h}^m \leftarrow \text{diff}(q_{1:h}^m)$ 
9   // Sample  $E$  control noise vectors
10   $\{\epsilon_0, \epsilon_1, \dots, \epsilon_{E-1}\} \sim \mathcal{N}(0, \sigma^2)$ 
11  for  $e = 0, 1, \dots, E-1$  do
12     $x_0, u_0, C(e) \leftarrow \{q_0, \dot{q}_0\}, \{\ddot{q}_0\}, 0$ 
13    // Add noise to the system
14    for  $t = 1, 2, \dots, h$  do
15       $x_t \leftarrow F(x_{t-1}, u_{t-1} + \epsilon_e(t))$ 
16    end
17    // Evaluate trajectory effort
18     $C(e) \leftarrow$ 
19       $C_{\text{smooth}}(x_h) + \sum_{t=1}^h C_{\text{run}}(x_t) + \gamma u_{t-1}^\top \Sigma^{-1} \epsilon_{t-1}$ 
20  end
21  // Compute control input according to the cost
22   $\rho \leftarrow \min C$ 
23   $\eta \leftarrow \sum_{e=1}^E \exp(\frac{1}{\lambda}(C(e) - \rho))$ 
24   $u^m \leftarrow \dot{q}_{1:h}^m + \sum_{e=1}^E \frac{1}{\eta} \exp(\frac{1}{\lambda}(C(e) - \rho)) \epsilon_e$ 
25   $U \leftarrow U \cup u^m$ 
26 end
27 // Compute joint positions and velocities
28  $Q^m, \dot{Q}^m \leftarrow \text{integrate}(U)$ 
29  $\dot{q}_{1:c}^* \leftarrow \text{selectBest}(Q^m, \dot{Q}^m, U)$ 

```

racy and constraint violation penalties over the horizon h , and C_{pert} discourages noisy trajectories weighted by γ .

The running cost C_{run} evaluates both end-effector tracking accuracy and constraint violations:

$$C_{\text{run}} = C_e + C_c \quad (9)$$

$$C_e = \|f(q) - {}^r_{ee}T\| \quad (10)$$

$$C_c = w_c \cdot (\mathbb{I}(\text{collision}) + \mathbb{I}(\text{limit})), \quad (11)$$

where $f(\cdot)$ represents the forward kinematics given the joint configuration q , ${}^r_{ee}T$ denotes the target pose, and $\mathbb{I}(\cdot)$ is an indicator function that suggests collision or joint limit violations with weight w_c .

The smoothness cost C_{smooth} ensures trajectory continuity:

$$C_{\text{smooth}} = \sum_{t=1}^h \left(\|q_t - q_{t-1}\| + \max_{1 \leq k \leq n_{\text{dof}}} |(q_t - q_{t-1})_k| \right), \quad (12)$$

where $\|q_t - q_{t-1}\|$ penalizes total joint travel distance, and $\max_{1 \leq k \leq n_{\text{dof}}} |(q_t - q_{t-1})_k|$ penalizes maximum individual joint displacement between consecutive waypoints. These terms work together to generate smooth trajectories while preventing jerky or unstable joint movements.

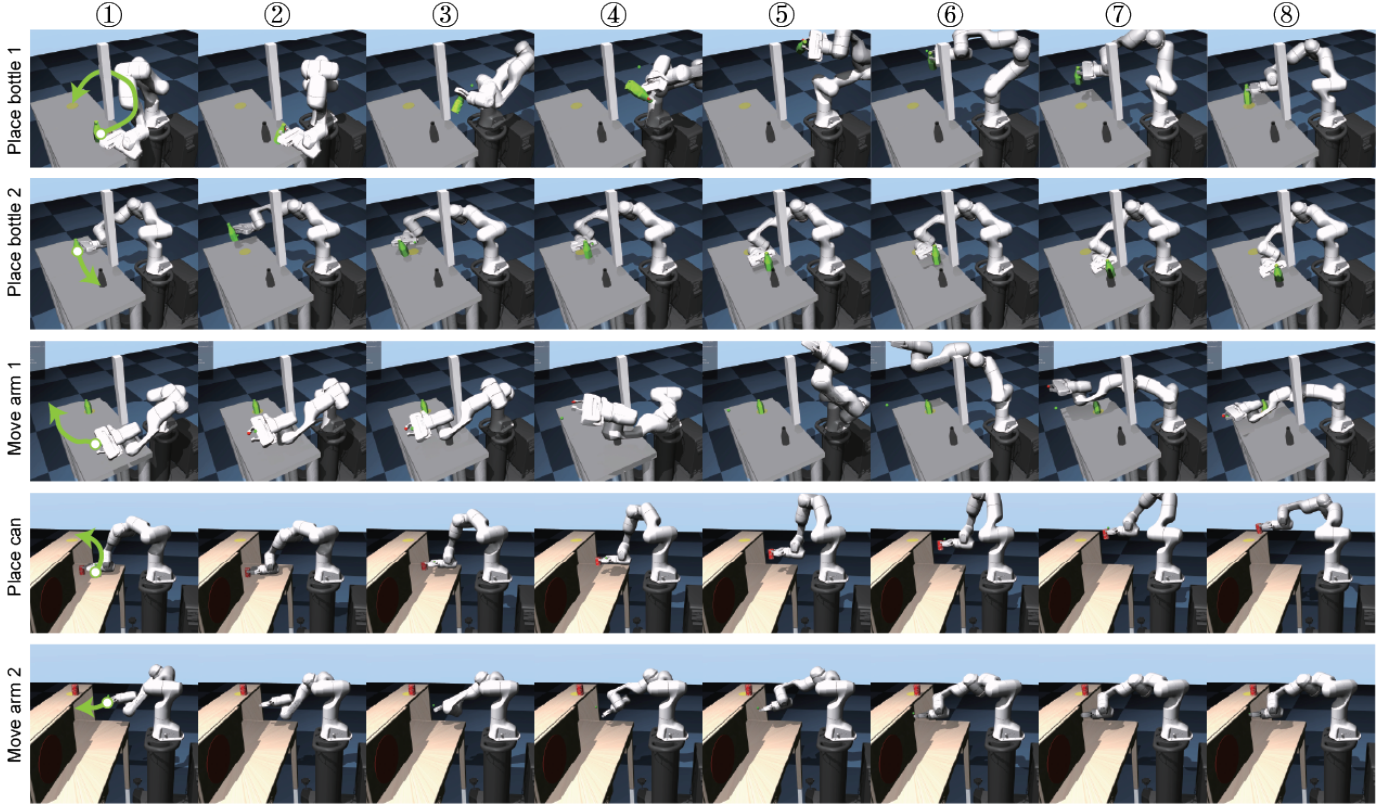


Fig. 4: Snapshots from five trials in the simulated environment demonstrating the proposed tele-MPPI method. The robot successfully avoids obstacles while following green trajectories controlled by the human operator.

Finally, the perturbation cost C_{pert} introduces a noise penalty over the control input:

$$C_{\text{pert}} = \tilde{u} \cdot \Sigma^{-1} \epsilon, \quad (13)$$

where \tilde{u} is the perturbed control input, Σ is the control covariance matrix, and ϵ is the noise vector. This term penalizes noisy trajectories in the optimization process, promoting smoother and more stable joint trajectory solutions.

These cost terms collectively promote stable and smooth trajectories while maintaining computational efficiency for real-time control. The weight for each rollout is computed according to their total cost as follows:

$$\omega = \frac{1}{\eta} \exp \left(-\frac{1}{\lambda} (C - \rho) \right), \quad (14)$$

where ρ is the minimum cost among all rollouts, and $\eta = \sum \exp \left(-\frac{1}{\lambda} (C - \rho) \right)$ serves as the normalizing factor

The final control command is computed as a weighted sum of perturbed accelerations:

$$u = \sum^E \omega \cdot \tilde{u}. \quad (15)$$

To determine the optimal solution, the system generates M trajectories through parallel execution of the aforementioned procedure. The system performs temporal integration on joint accelerations to obtain positions and velocities, then the optimal solution is found by evaluating the cost function over all

M trajectories as follows:

$$u^* = \arg \min_u (C_{\text{smooth}}(u) + C_{\text{run}}(u)). \quad (16)$$

Since u represents a weighted sum across all rollouts (see (15)), the weighting factors mitigate the effects of noise in \tilde{u} . Thereby, the perturbation cost C_{pert} is omitted from (16). Prior to execution, the optimal trajectory undergoes quintic interpolation to ensure smooth and continuous motion of the robotic manipulator.

V. SIMULATION AND EXPERIMENT

The proposed system was validated through two distinct sets of experiments: (1) trajectory tracking and obstacle avoidance in simulated and real-world environments, and (2) qualitative evaluation of the haptic feedback module via grasping tasks on fragile/deformable objects. The system was implemented in Python, with the MPPI algorithm accelerated via GPU for parallel computation. The hardware included a 7-DoF robotic arm equipped with two VBTs on its end-effector, a VR base station, and a custom handheld device. Simulations were conducted using a physics-based simulator, while real-world experiments utilized a real-time kernel-based lower-level controller. The system operates at 20 Hz, processing data from observation horizon $h = 10$ time steps per cycle. A GPU-based inverse kinematics solver generates $r = 100$ candidate solutions per time step, with a greedy search selecting $M = 20$ optimal trajectories. The MPPI algorithm is configured with $\lambda = 100$, $\gamma = 20$, and noise covariance matrix $\Sigma = \text{diag}(5, 5, 5, 5, 5, 5, 5)$. For each trajectory, $E = 300$

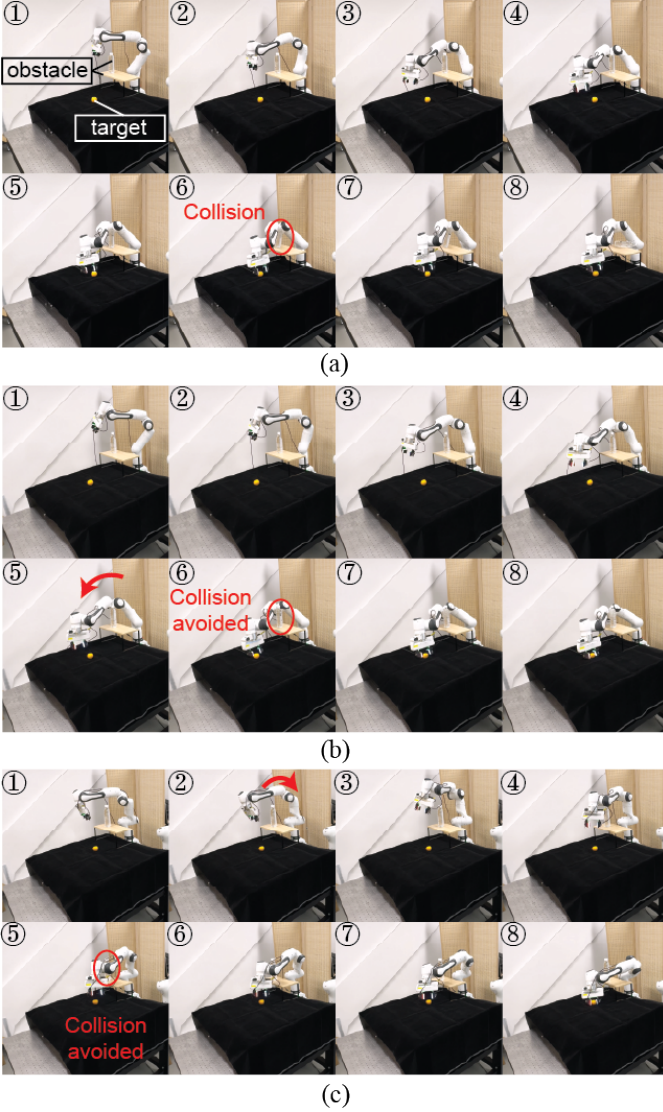


Fig. 5: Real-world experiment comparing the proposed teleoperation system with the tele-MPPI method to a baseline system without tele-MPPI. (a) Without the tele-MPPI module, the elbow collided with and knocked over the water bottle. (b) With the tele-MPPI module, the arm made fine adjustments to its elbow joint to avoid collisions. (c) The arm performed alternative pose adjustments to navigate around the obstacle and successfully retrieve the orange.

noise-perturbed trajectories are generated in parallel, with the first $c = 2$ points used for the arm control. Quintic interpolation is applied at the end to smooth the trajectory.

A. Haptic Feedback Module Validation

This experiment validates the effectiveness of the haptic feedback gripper module. The robotic arm's gripper was controlled via the handheld device to grasp delicate objects. System performance was evaluated by monitoring object deformation and feedback from the handheld device.

Grasp fragile objects: Experiments were conducted on fragile objects, including strawberries, oranges, chips, hand cream containers, and paper cups. As shown in Fig. 1(b) and the supplementary video, the system effectively controlled gripper force to prevent deformities or damage to the objects

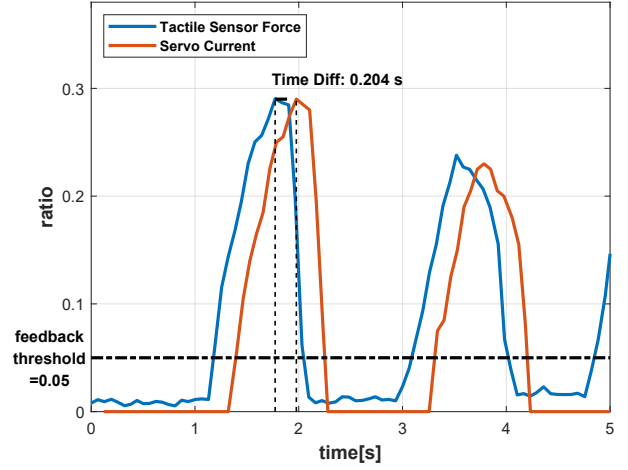


Fig. 6: Haptic feedback latency performance of the proposed system.

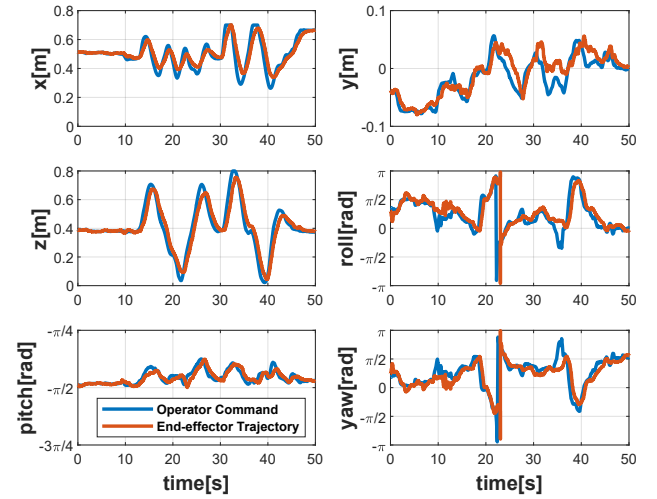


Fig. 7: Trajectory tracking performance of the proposed system.

while ensuring stable grasping. The gripper dynamically adjusted its force during manipulation to maintain stability.

Feedback frequency and latency: As shown in Fig. 6, the latency between the tactile sensor and the servo was measured to be approximately 200ms, with a frequency of 15Hz.

B. Trajectory Tracking and Obstacle Avoidance

This set evaluated the system's ability to track trajectories and avoid collisions in structured environments.

Teleoperation with collision avoidance: As shown in Fig. 4, simulated tasks included placing a bottle, moving the arm around a pillar obstacle, placing a can, and navigating around a bookshelf. In the real-world setting in Fig. 5, the human operator controlled the robotic arm to retrieve an orange behind a water bottle. The robotic arm dynamically adjusted its pose to navigate constrained spaces.

In the *Place Bottle 1* task, the arm moved the bottle from its initial to the target position, dynamically adjusting its pose to avoid obstacles. As shown in Fig. 4 ③ to ⑤, the gripper rotated the bottle to bypass the obstacle and quickly recovered for subsequent placement. In the *Place Bottle 2* task, the arm returned the bottle to its original position, adopting a

hugging posture to avoid the obstacle and making significant pose adjustments for larger movements. In the *Move Arm 1* task, the robot initially could not reach the target due to obstruction by the pillar obstacle. However, the tele-MPPI algorithm successfully identified an alternative path, guiding the arm around the obstacle from above to reach the target.

In the second simulation task, *Place Can* and *Move Arm 2*, the arm retrieved a can from a shelf and placed it above the shelf while avoiding collisions. When the operator attempted to guide the arm into the shelf at different angles, the system automatically adjusted the arm's posture or halted movement if no collision-free path was available.

In real-world experiments, as shown in Fig. 5, initial trials without the tele-MPPI module resulted in the elbow colliding with and knocking over the water bottle. When the tele-MPPI module was enabled, the arm made fine adjustments to its elbow joint to avoid collisions. In a third trial, with the orange positioned closer to the obstacle's left side, the arm performed significant pose adjustments to navigate around the obstacle and successfully retrieve the orange.

Trajectory tracking: The robotic arm traced an "8" shape in free space to evaluate teleoperation tracking performance. As shown in Fig. 7, the end-effector trajectory closely follows the command trajectory from the handheld device, with minimal deviations at rest poses. A slight time lag of approximately 0.5 seconds occurs during dynamic motions, caused by the tele-MPPI algorithm's requirement for sequential operator commands ($h = 10$ at 20 Hz) as input. Despite this latency, the system demonstrates robust steady-state tracking performance, confirming its effectiveness for teleoperated tasks. Additionally, between 30–40 seconds, tracking error increases as the robot prioritizes self-collision avoidance, slightly compromising end-effector tracking accuracy.

VI. DISCUSSION

The latency in the collision avoidance and trajectory tracking experiments mainly stems from the MPPI algorithm itself, which inherently introduces delays due to the need to compute a sequence of control actions over a prediction horizon. One potential approach to reducing this latency is to incorporate action prediction algorithms that can proactively generate control sequences, rather than relying on sufficient real-time data collection before execution. However, such predictive methods often come at the cost of reduced accuracy, which can compromise system performance in complex or dynamic environments. The tracker introduces only minimal intrinsic error; however, the stability of its output data is somewhat limited, resulting in occasional jitter in the robotic arm during teleoperation. This can be mitigated through appropriate signal filtering techniques to enhance overall control stability. The Haptic Feedback system experiences latency from multiple sources, including the visual-tactile sensor's camera (due to image acquisition and processing), system-level data transmission and computation delays. Given current hardware and system constraints, these latencies are relatively difficult to eliminate in the short term.

VII. CONCLUSION

This work presented a teleoperation system integrating the newly proposed tele-MPPI algorithm for real-time obstacle avoidance and trajectory tracking, combined with a haptic feedback handheld device for delicate object manipulation. The system's performance was validated through simulation and real-world experiments. Key results demonstrated that the tele-MPPI module enables dynamic obstacle avoidance in cluttered environments (e.g., navigating around pillars and shelves) while maintaining trajectory fidelity. The haptic feedback gripper module proved effective in preventing damage to fragile objects through adaptive force control, with real-time adjustments informed by haptic and visual feedback. The system's computational efficiency, operating at 20 Hz with GPU-accelerated parallelization, underscores its suitability for real-world applications.

REFERENCES

- [1] Y. Lee and W. Woo, "Interactive edutainment system with enhanced personalized user interface framework," *Transactions on Consumer Electronics (TCE)*, vol. 53, no. 2, pp. 424–432, 2007.
- [2] Y. Han, "A low-cost visual motion data glove as an input device to interpret human hand gestures," *Transactions on Consumer Electronics (TCE)*, vol. 56, no. 2, pp. 501–509, 2010.
- [3] D. Arseneault and A. D. Whitehead, "Gesture recognition using markov systems and wearable wireless inertial sensors," *Transactions on Consumer Electronics (TCE)*, vol. 61, no. 4, pp. 429–437, 2015.
- [4] H. Liu, Z. Zhang, Z. Jiao, Z. Zhang, M. Li, C. Jiang, Y. Zhu, and S.-C. Zhu, "A reconfigurable data glove for reconstructing physical and virtual grasps," *Engineering*, vol. 32, pp. 202–216, 2024.
- [5] S. Kim, G. Park, S. Yim, S. Choi, and S. Choi, "Gesture-recognizing hand-held interface with vibrotactile feedback for 3d interaction," *Transactions on Consumer Electronics (TCE)*, vol. 55, no. 3, pp. 1169–1177, 2009.
- [6] H. Liu, Z. Zhang, X. Xie, Y. Zhu, Y. Liu, Y. Wang, and S.-C. Zhu, "High-fidelity grasping in virtual reality using a glove-based system," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 5180–5186, IEEE, 2019.
- [7] G. Niemeyer, C. Preusche, S. Stramigioli, and D. Lee, "Telerobotics," *Springer handbook of robotics*, pp. 1085–1108, 2016.
- [8] R. C. Goertz, "Fundamentals of general-purpose remote manipulators," *Nucleonics*, pp. 36–42, 1952.
- [9] K. Darvish, L. Penco, J. Ramos, R. Cisneros, J. Pratt, E. Yoshida, S. Ivaldi, and D. Pucci, "Teleoperation of humanoid robots: A survey," *Transactions on Robotics (T-RO)*, vol. 39, no. 3, pp. 1706–1727, 2023.
- [10] C.-H. Lee, J. Choi, H. Lee, J. Kim, K.-m. Lee, and Y.-b. Bang, "Exoskeletal master device for dual arm robot teaching," *Mechatronics*, vol. 43, pp. 76–85, 2017.
- [11] Y. Ishiguro, T. Makabe, Y. Nagamatsu, Y. Kojio, K. Kojima, F. Sugai, Y. Kakiuchi, K. Okada, and M. Inaba, "Bilateral humanoid teleoperation system using whole-body exoskeleton cockpit tablis," *Robotics and Automation Letters (RA-L)*, vol. 5, no. 4, pp. 6419–6426, 2020.
- [12] C. Hu, M. Q. Meng, P. X. Liu, and X. Wang, "Visual gesture recognition for human-machine interface of robot teleoperation," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1560–1565, IEEE, 2003.
- [13] S. Bertrand, L. Penco, D. Anderson, D. Calvert, V. Roy, S. McCrory, K. Mohammed, S. Sanchez, W. Griffith, S. Morfe, et al., "High-speed and impact resilient teleoperation of humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 189–196, IEEE, 2024.
- [14] S. J. Siddiqi, M. A. Jan, A. M. Basalamah, and M. Tariq, "Secure teleoperated vehicles in augmented reality of things: A multichain and digital twin approach," *Transactions on Consumer Electronics (TCE)*, vol. 70, no. 1, pp. 956–965, 2023.
- [15] S. Katsigiannis, R. Willis, and N. Ramzan, "A qoe and simulator sickness evaluation of a smart-exercise-bike virtual reality system via user feedback and physiological signals," *Transactions on Consumer Electronics (TCE)*, vol. 65, no. 1, pp. 119–127, 2018.
- [16] S.-N. Yao, "Headphone-based immersive audio for virtual reality headsets," *Transactions on Consumer Electronics (TCE)*, vol. 63, no. 3, pp. 300–308, 2017.

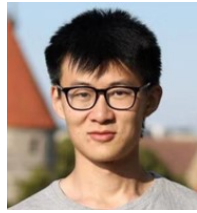
- [17] L. Penco, B. Clément, V. Modugno, E. M. Hoffman, G. Nava, D. Pucci, N. G. Tsagarakis, J.-B. Mouret, and S. Ivaldi, "Robust real-time whole-body motion retargeting from human to humanoid," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 425–432, IEEE, 2018.
- [18] K. Ayusawa and E. Yoshida, "Motion retargeting for humanoid robots based on simultaneous morphing parameter identification and motion optimization," *Transactions on Robotics (T-RO)*, vol. 33, no. 6, pp. 1343–1357, 2017.
- [19] Y.-P. Su, X.-Q. Chen, C. Zhou, L. H. Pearson, C. G. Pretty, and J. G. Chase, "Integrating virtual, mixed, and augmented reality into remote robotic applications: A brief review of extended reality-enhanced robotic systems for intuitive telemanipulation and telemanufacturing tasks in hazardous conditions," *Applied Sciences*, vol. 13, no. 22, p. 12129, 2023.
- [20] Y. Liang, W. Li, Y. Wang, R. Xiong, Y. Mao, and J. Zhang, "Dynamic movement primitive based motion retargeting for dual-arm sign language motions," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 8195–8201, IEEE, 2021.
- [21] T. Kim and J.-H. Lee, "C-3po: Cyclic-three-phase optimization for human-robot motion retargeting based on reinforcement learning," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 8425–8432, IEEE, 2020.
- [22] S. Choi, M. J. Song, H. Ahn, and J. Kim, "Self-supervised motion retargeting with safety guarantee," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 8097–8103, IEEE, 2021.
- [23] T. Asfour and R. Dillmann, "Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, pp. 1407–1412, IEEE, 2003.
- [24] B. Dariush, M. Gienger, A. Arumbakkam, C. Goerick, Y. Zhu, and K. Fujimura, "Online and markerless motion retargeting with kinematic constraints," in *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, pp. 191–198, IEEE, 2008.
- [25] G. Bin Hammam, P. M. Wensing, B. Dariush, and D. E. Orin, "Kinodynamically consistent motion retargeting for humanoids," *International Journal of Humanoid Robotics (IJHR)*, vol. 12, no. 04, p. 1550017, 2015.
- [26] Z. Zhao, Y. Li, W. Li, Z. Qi, L. Ruan, Y. Zhu, and K. Althoefer, "Tacman: Tactile-informed prior-free manipulation of articulated objects," *Transactions on Robotics (T-RO)*, vol. 41, no. 2, pp. 538–557, 2024.
- [27] W. Li, M. Wang, J. Li, Y. Su, D. K. Jha, X. Qian, K. Althoefer, and H. Liu, "L3f-touch: A wireless gelsight with decoupled tactile and three-axis force sensing," *IEEE Robotics and Automation Letters*, vol. 8, no. 8, pp. 5148–5155, 2023.
- [28] B. Ames, J. Morgan, and G. Konidaris, "Ikflow: Generating diverse inverse kinematics solutions," *Robotics and Automation Letters (RA-L)*, vol. 7, no. 3, pp. 7177–7184, 2022.
- [29] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos, et al., "Curobo: Parallelized collision-free robot motion generation," in *Proceedings of International Conference on Robotics and Automation (ICRA)*, pp. 8112–8119, IEEE, 2023.



Fan Wu received his B.S. degree in Robotics Engineering (Automation) in 2023 and is currently pursuing an M.S. degree in Control Science and Engineering at the School of Automation Science and Electrical Engineering, Beihang University. He is also an intern at the Beijing Institute for General Artificial Intelligence (BIGAI). His research interests include teleoperation, robot control, and motion planning.



Ziyuan Jiao (Member, IEEE) received his Ph.D. degree in Mechanical Engineering from the University of California, Los Angeles (UCLA) in 2022. He is now a research scientist at the State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). Dr. Jiao's research focuses on robotic manipulation planning, task and motion planning, mobile manipulation, and optimization.



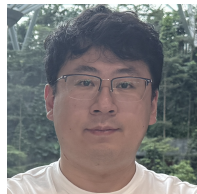
Wanlin Li (Member, IEEE) received the B.Eng. from the joint program of School of Electronic Engineering and Computer Science (EECS) at Queen Mary University of London (QMUL) and Beijing University of Posts and Telecommunications in 2016, and the Ph.D. degree from the School of EECS at QMUL in 2021. He is currently a research scientist at State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). His current research interests include robotic force and tactile sensing.



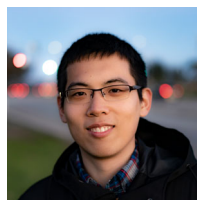
Zeyu Zhang (Member, IEEE) received his Ph.D. degree in Computer Science from the University of California, Los Angeles (UCLA) in 2023. He is a research scientist at State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). He received an M.S. degree in Computer Science from UCLA in 2019 and B.S. degree in Computer Science from Hunan University in 2017. His research interests focus on robot perception, learning, and cognitive robotics.



Hang Li received the B.S. and the M.S. degrees from the Central Academy of Fine Arts in 2018 and 2022. He is now a research engineer at State Key Laboratory of General Artificial Intelligence, Beijing Institute for General Artificial Intelligence (BIGAI). His research interests include robotics and robot design.



Jiahao Wu received his Bachelor's degree in Mechanical Engineering from Stevens Institute of Technology in May 2018 and his Master's degree in Mechanical Engineering from Columbia University in December 2021. He is currently a Senior Research Engineer at the Beijing Institute for General Artificial Intelligence (BIGAI). His research focuses on embodied intelligence in robotic systems and advanced studies in robot teleoperation.



Baoxiong Jia is currently a research scientist at Beijing Institute for General Artificial Intelligence. He obtained his Ph.D. degree from the Department of Computer Science at UCLA, advised by Prof. Song-Chun Zhu. Before that, He received his B.S. and M.S. degrees in computer science from Peking University and UCLA in 2018 and 2019, respectively. His research mainly focuses on spatial-temporal learning and reasoning, and its application to real-world understanding and planning tasks (including scene understanding, activity recognition, and prediction, task learning, inverse planning, etc.). He has related publications in ICML, NeurIPS, CVPR, ICLR, and TPAMI.



Shaopeng Dong obtained his B.S. degree at China Agricultural University in 2004, obtained his M.S. degree at Beihang University in 2007, and received his Ph.D. degree at Beihang University in 2018. Now he is an associate professor in School of Automation Science and Electrical Engineering, Beihang University. His research interests focus on structural health inspection and Embedded Systems.