Robotic Arm Teleoperation through Real-time Human Arm Motion Imitation

Presenter(s): Qin Su

Faculty Mentor: Biyun Xie

University of Kentucky

ABSTRACT

Human-robot teleoperation is crucial for simplifying robot programming and ensuring operator's safety in remote and dangerous environment. This study aims to develop two efficient robot teleoperation algorithms through real-time human arm motion imitation. In the first method, the robotic arm is only required to mimic human arm postures through computing joint angles based on the equivalent joint positions captured by a motion capture system. In the second method, the robotic arm is required to satisfy the same end effector position and orientation and also have a similar arm posture. To validate these developed methods, simulation and physical experiments are conducted on a Kinova Gen-3 robot arm. In physical experiments, the calculated joint angles are executed by angular speed control, with a filter designed for smooth robot movement. The results approve the effectiveness of the algorithms.

Methods

Method 1: Mimic human arm posture

Step 1: Compute joint positions based on motion capture data

$$R = R_y R_x R_z = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\gamma & -sin\gamma \\ 0 & sin\gamma & cos\gamma \end{bmatrix} \begin{bmatrix} cos\eta & -sin\eta & 0 \\ sin\eta & cos\eta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Step 2: Compute joint angles based on joint positions

To ensure that the Kinova robot accurately mimics these human arm postures, we apply geometric methods for inverse kinematics to compute the robot's joint angles. This joint mapping algorithm is a novel equivalent-angle method in which the corresponding robot joints are calculated using basic vector algebra:

$$\theta_1 = atan2(proj(h_e)_x, proj(h_e)_y)$$

$$\theta_2 = arccos(\frac{h_{bs} \cdot h_{se}}{|h_{bs}||h_{se}|})$$

$$\theta_3 = atan2(proj(h_w)_x, proj(h_w)_z)$$

$$\theta_4 = arccos(\frac{h_{se} \cdot h_{ew}}{|h_{se}||h_{ew}|})$$

$$\theta_5 = atan2(proj(h_p)_x, proj(h_p)_z)$$

$$\theta_6 = arccos(\frac{h_{ew} \cdot h_{wp}}{|h_{ew}||h_{wp}|})$$

$$\theta_7 = atan2(proj(h_n)_x, proj(h_n)_z)$$

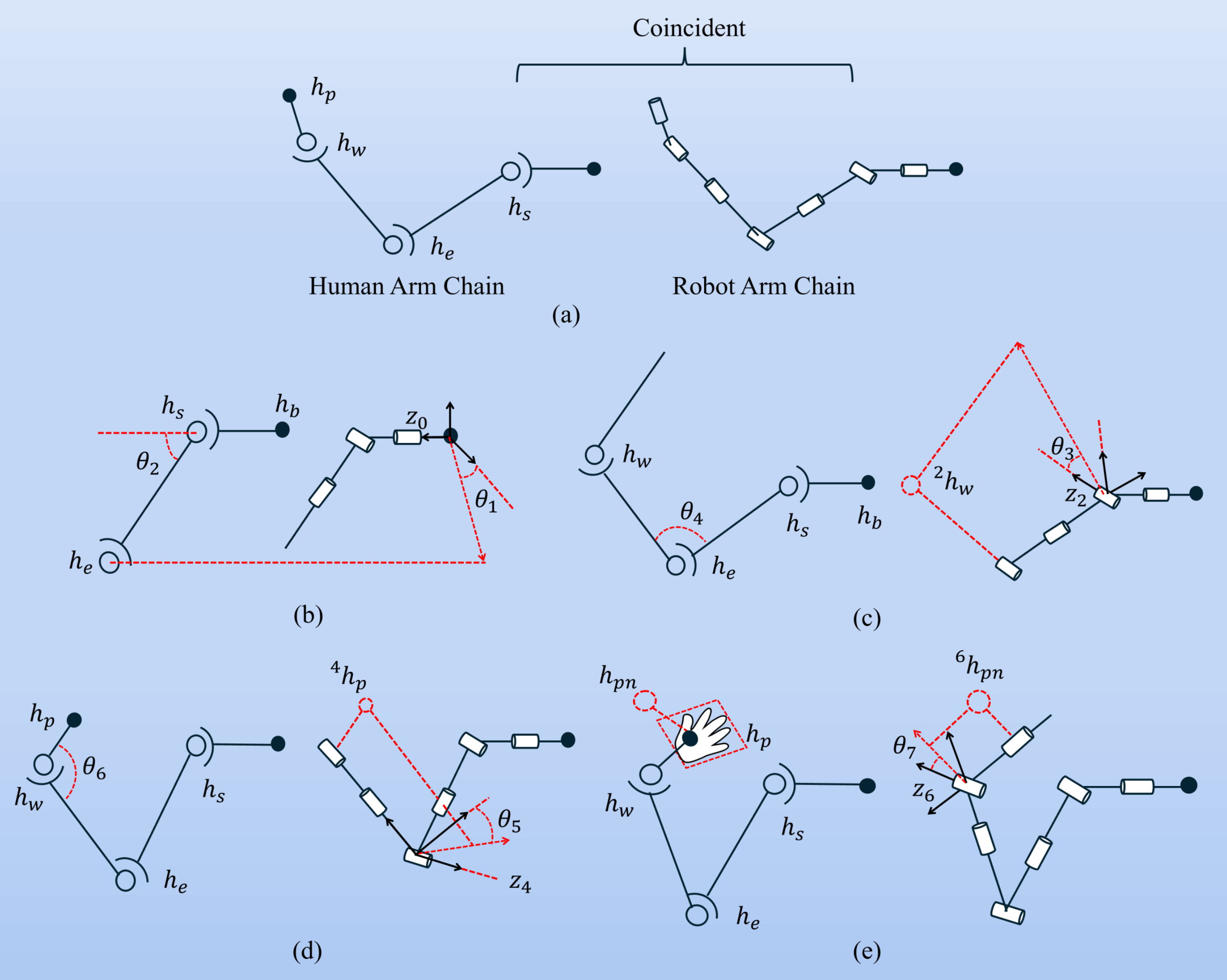

Fig. 1. The inverse kinematics to solve the human joint angles.

Methods (cont.)

Method 2: Satisfy end effector position and orientation and elbow swivel angle

The elbow swivel angle (ESA) , denoted as φ, is defined as the angle of rotation of the elbow about the axis that begins at the shoulder and points toward the wrist measured from a reference vector u. In Fig. 2, P_s, P_e, and P_w are the positions of the shoulder, elbow, and wrist respectively; n is the unit vector in the direction of P_s to P_w; and r is the unit vector along R, which lies on a plane perpendicular to n and intersects P_e. P_c is the center of rotation and is calculated as:

$$P_c = P_s + n * U * \cos(\Omega)$$

where U is upper arm length. u is the projection of a reference vector a onto the perpendicular plane, and $v = u \times n$. Finally, the ESA can be calculated using :

$$\varphi = atan2(v \cdot r, u \cdot r)$$

By computing the ESA of the human arm and ensuring the robotic arm maintains the same ESA as the human arm, the equivalent elbow position can be computed. Based on the equivalent elbow position and human palm position and orientation, the joint angles of robot arm can be computed by using the first method.

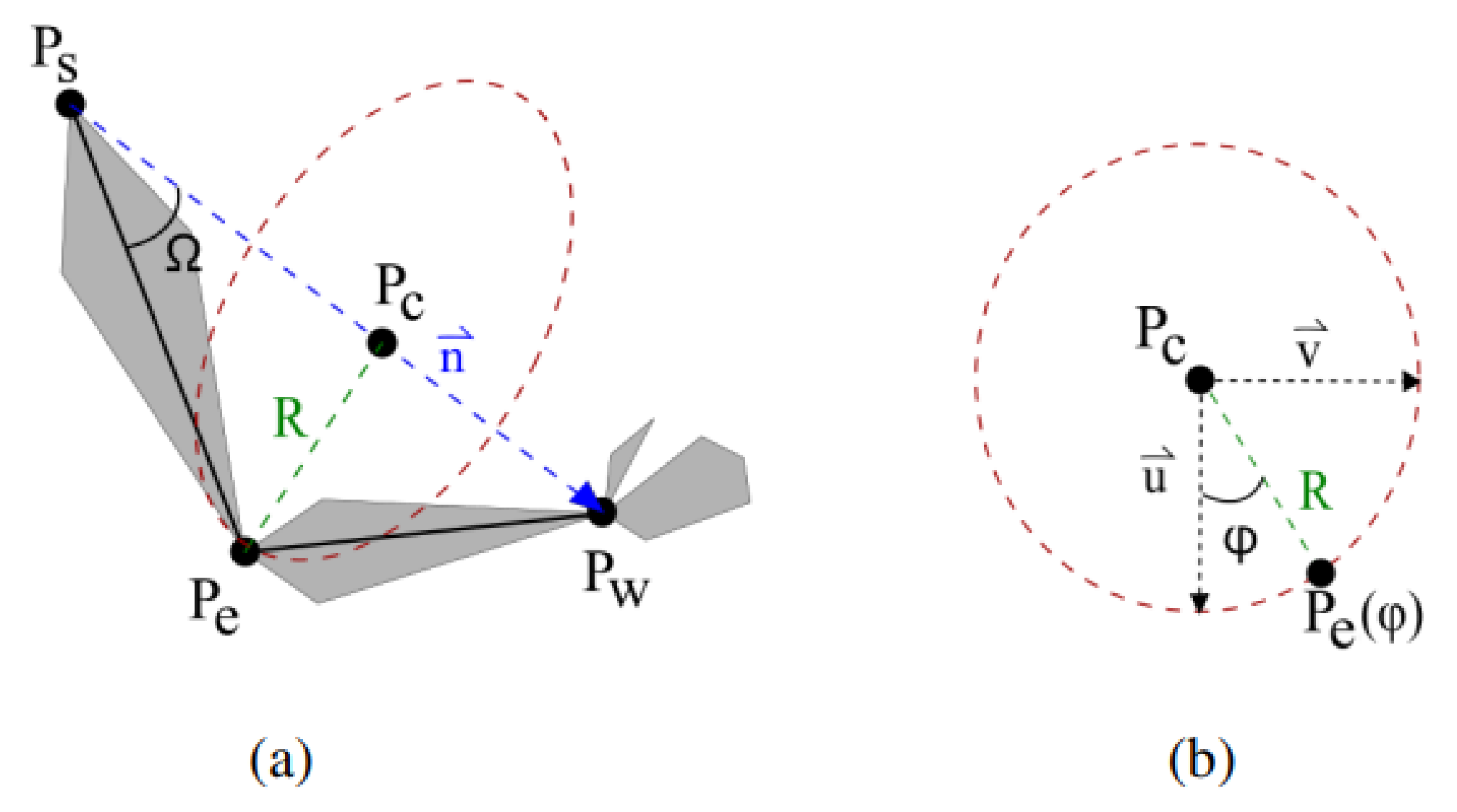

Fig. 2. Diagram of ESA calculation.

Results

Results 1: Mimic human arm posture

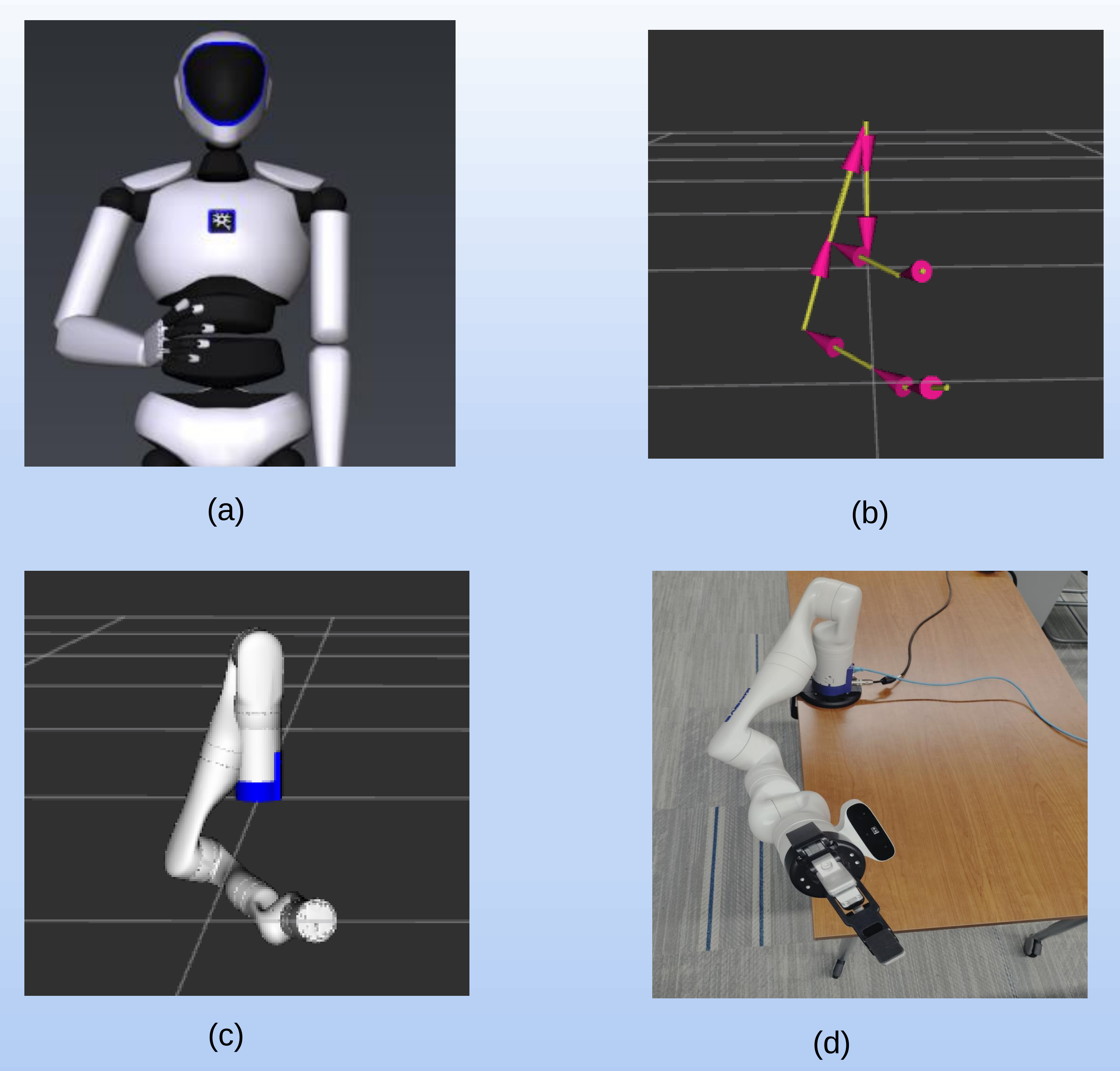

Fig. 3. An example of the "pick-shake" task is shown. (a) is the captured human arm posture. (b) is the comparison between the captured human arm posture and the calculated robot arm posture. (c) is the arm posture of simulated robot arm. (d) is the arm posture of the physical robot arm.

Results (cont.)

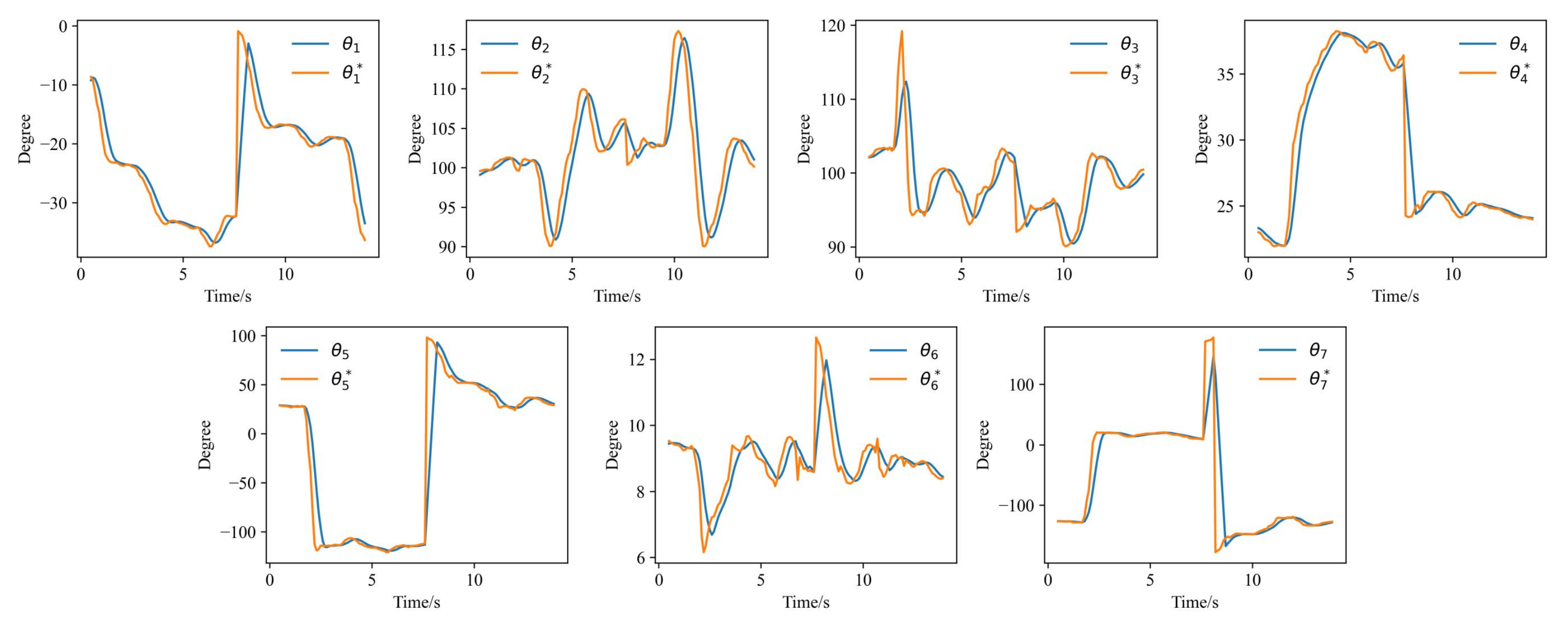

Fig. 4. Calculated joint angles θ^* and actual angles θ.

In experiments, the real-time transmission of motion data is facilitated through the TCP/IP protocol, establishing a reliable network communication channel between the Windows operating system and an Ubuntu virtual machine. Our methodology has undergone rigorous testing through simulations within the RViz environment, part of the ROS (Robot Operating System) framework, and has been validated through physical experiments on the robot, as illustrated in Fig. 3.

The joint angles are computed via inverse kinematics algorithms. A moving-average filter is then applied to these angles prior to their delivery to the robot controller, ensuring the smooth operation of the robotic arm. This filtering process is essential for the precise emulation of human arm dynamics. Additionally, we collect real joint angle data from the Kinova robot to verify the accuracy of the robotic arm's movements in mirroring human postures, thus validating the system's efficacy, as shown in Fig. 4.

Results 2: Satisfy end effector position and orientation and elbow swivel angle

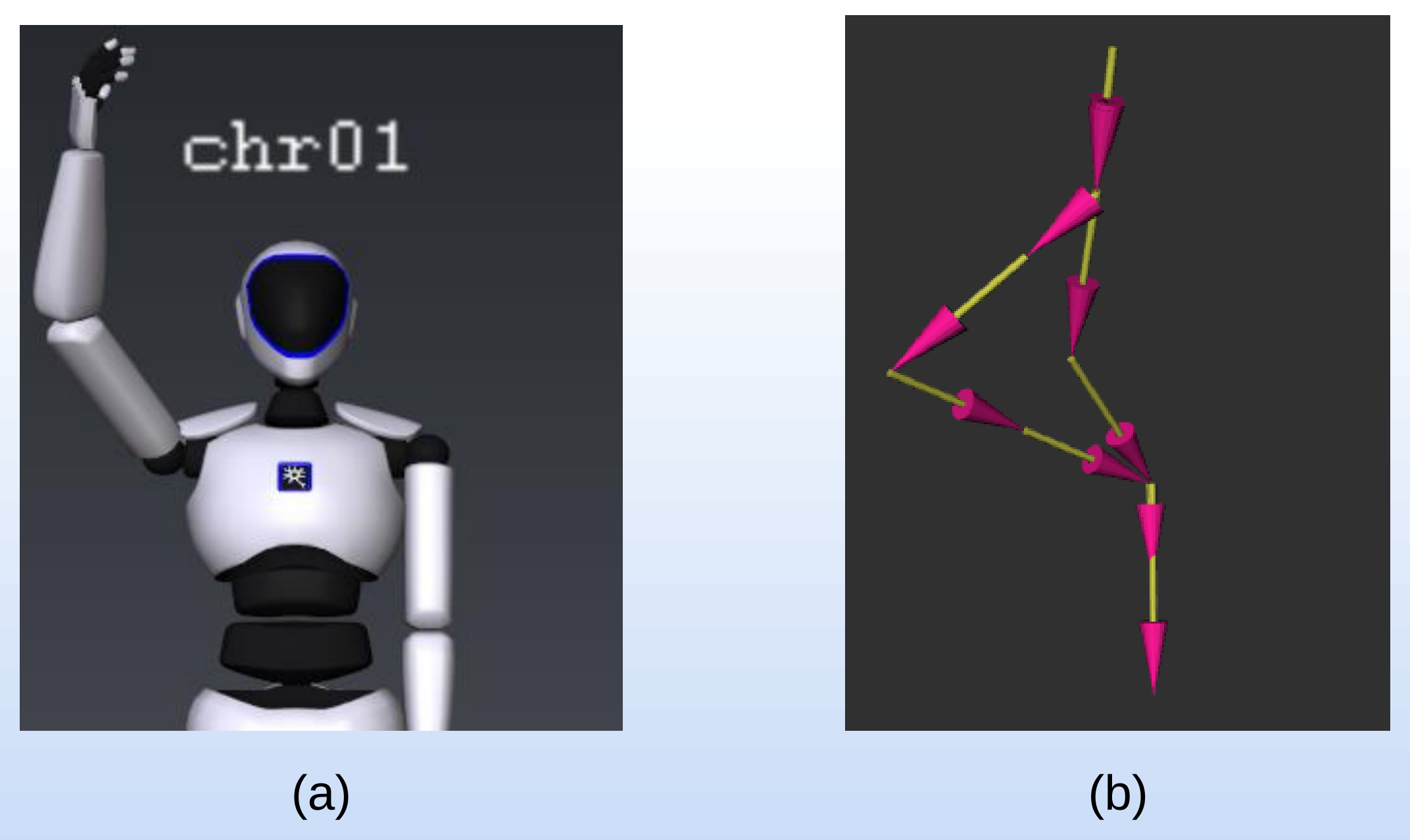

Fig. 5. An example of the " reaching high" task is shown. (a) is the captured human arm posture. (b) is the comparison between the captured human arm posture and the calculated robot arm posture.

CONCLUSION

Our research provides a significant advance in robot arm teleoperation through real-time human arm motion imitation. By developing two advanced inverse kinematics algorithms for different scenarios, we've achieved a high fidelity in mapping human arm movements to a robotic counterpart. This ensures that the robot arm mimics human arm posture and satisfy the human palm's position and orientation accurately. The smoothness of the robot arm movements have been enhanced by a specialized filter for angular speed control. Through rigorous testing in both simulation and physical experiments, we've demonstrated the platform's effectiveness in various scenarios, proving it to be a robust tool for simplifying robot programming, improving safety, and fostering human-robot collaboration.

In our future work, we will consider robot self-collision avoidance and surrounding obstacle avoidance to further improve the above developed teleoperation algorithms.