A End-Effector Control Mapping Details

With our different modes of operation, we can accurately map the user's real-world end effector (EE) center, which can comfortably and freely draw spheres in physical space, to the center of the corresponding task's workspace, while adjusting the control ratio.

Algorithm 1 Mapping the Workspace

Require: Robot workspace center robot_center, robot workspace radius robot_radius
Require: Human movement data max_x, max_y, max_z, min_x, min_y, min_z
Ensure: Mapped human workspace within robot workspace

$\texttt{human\_center} \leftarrow (\frac{\texttt{max\_x}+\texttt{min\_x}}{2}, \frac{\texttt{max\_y}+\texttt{min\_y}}{2}, \frac{\texttt{max\_z}+\texttt{min\_z}}{2})$

$\texttt{human\_radius} \leftarrow \min(\texttt{max\_x} - \texttt{human\_center.x}, \texttt{max\_y} - \texttt{human\_center.y}, \texttt{max\_z} - \texttt{human\_center.z})$

$\texttt{control\_rate} \leftarrow \frac{\texttt{robot\_radius}}{\texttt{human\_radius}}$

$\texttt{offset} \leftarrow \texttt{robot\_center} - \texttt{control\_rate} \times \texttt{human\_center}$

$\texttt{Mapped\_ee} \leftarrow \texttt{human\_ee} \times \texttt{control\_rate} + \texttt{offset}$

Ensure $\|\texttt{mapped\_ee} - \texttt{robot\_center}\| < \texttt{robot\_radius}$ for safety

Algorithm 2 Map Certain Task

Require: Robot workspace center robot_center, Robot workspace radius robot_radius
Require: Task center task_center, Task control rate task_control_rate
Require: Human movement data max_x, max_y, max_z, min_x, min_y, min_z
Ensure: Mapped human task within robot workspace

$\texttt{human\_center} \leftarrow (\frac{\texttt{max\_x}+\texttt{min\_x}}{2}, \frac{\texttt{max\_y}+\texttt{min\_y}}{2}, \frac{\texttt{max\_z}+\texttt{min\_z}}{2})$

$\texttt{human\_radius} \leftarrow \min(\texttt{max\_x} - \texttt{human\_center.x}, \texttt{max\_y} - \texttt{human\_center.y}, \texttt{max\_z} - \texttt{human\_center.z})$

$\texttt{control\_rate} \leftarrow \texttt{task\_control\_rate}$

$\texttt{offset} \leftarrow \texttt{task\_center} - \texttt{control\_rate} \times \texttt{human\_center}$

$\texttt{Mapped\_ee} \leftarrow \texttt{human\_ee} \times \texttt{control\_rate} + \texttt{offset}$

Ensure $\|\texttt{mapped\_ee} - \texttt{robot\_center}\| < \texttt{robot\_radius}$ for safety

Algorithm 3 Map Rotation

Require: Task rotation range task_roll_range, task_pitch_range, task_yaw_range
Require: Human movement data max_roll, max_pitch, max_yaw, min_roll, min_pitch, min_yaw

$\texttt{rate\_roll} \leftarrow \frac{(\texttt{max\_roll}-\texttt{min\_roll})}{\texttt{task\_roll\_range}}$

$\texttt{rate\_pitch} \leftarrow \frac{(\texttt{max\_pitch}-\texttt{min\_pitch})}{\texttt{task\_pitch\_range}}$

$\texttt{rate\_yaw} \leftarrow \frac{(\texttt{max\_yaw}-\texttt{min\_yaw})}{\texttt{task\_yaw\_range}}$

$\texttt{offset\_roll} \leftarrow -\texttt{rate\_roll} \times \frac{(\texttt{max\_roll}-\texttt{min\_roll})}{2}$

$\texttt{offset\_pitch} \leftarrow -\texttt{rate\_pitch} \times \frac{(\texttt{max\_pitch}-\texttt{min\_pitch})}{2}$

$\texttt{offset\_yaw} \leftarrow -\texttt{rate\_yaw} \times \frac{(\texttt{max\_yaw}-\texttt{min\_yaw})}{2}$

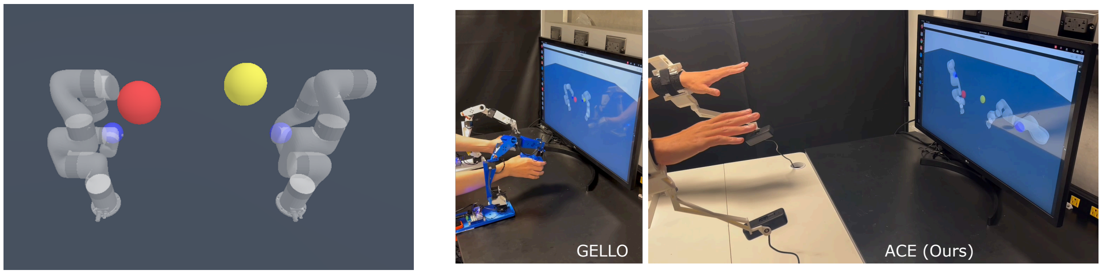
$\texttt{Mapped\_roll} \leftarrow \texttt{human\_roll} \times \texttt{rate\_roll} + \texttt{offset\_roll}$

$\texttt{Mapped\_pitch} \leftarrow \texttt{human\_pitch} \times \texttt{rate\_pitch} + \texttt{offset\_pitch}$

$\texttt{Mapped\_yaw} \leftarrow \texttt{human\_yaw} \times \texttt{rate\_yaw} + \texttt{offset\_yaw}$

B Target-Reaching Task Details

We design a target-reaching game for testing the key performance parameters of both systems in simulation. A screenshot of the designed game is shown in Fig. 6, where the only goal is to control the end-effectors (also represented as spheres) to reach the randomly generated goals. When the goal for both arms is reached and the end-effectors are kept inside the goals for an assigned keep time, it



(a) Game screenshot. (b) User play example..

Figure 6: Illustration of the designed target-reaching game, using XArm for example. The blue spheres represent the end-effector, the red and yellow spheres represent the target of the left and the right arm separately. We compare our ACE (b, right) with GELLO (b, left)

Table 5: The key parameters of the four variants. Note that the workspace, target, and end-effector are all represented as spheres so the only parameters are their radius.

Settings	Workspace Radius (cm)	Target Radius (cm)	End-Effector Sphere (cm)	Keep Time (secs)
Fine-grained	5	2	1	1
Board	60	12	6	0.05
General (a)	10	4	2	0.5
General (b)	25	8	4	0.1

is noted as a success and two new goals will be generated in the workspace. We take it as "inside" depending on the position of the end-effector and the target ball, using the following formula:

$$\mathbf{1}(\text{reached}) = \|p_{\text{ee}} - p_{\text{target}}\| <= r_{\text{target}} - r_{\text{ee}},$$

where p represents for position and r represents for radius, and $r_{\text{target}} > r_{\text{ee}}$ all the time. The game supports various types of robotics arms (with different kinetics), and during our test we chose XArm for evaluation.

B.1 Configurations

We construct four different game configurations by varying the workspace, the size of the end-effector/target balls, along with the keep time. The four variants are designed to assess the tele-operation system's performance under different conditions by simulating real-world robotic tasks. Specifically, these setups include:

- *Fine-Grained*, for tasks requiring precise operation within a small space;
- *Broad*, for tasks needing a larger workspace but less accuracy;
- *General*, balanced the two, with *General (a)* leaning towards fine-grained operation and *General (b)* favoring a larger workspace.

These variants differ in terms of workspace size, end-effector/target region size, and duration the end-effectors must remain in the goal regions. Their precise game parameters are shown in Tab. 5.

B.2 Participants

We invite 6 students as our volunteers, four of them are males and two of them are females. Their height ranges from 165 cm to 198 cm, and their weights vary from 55 cm to 90 kg.

B.3 Experiment Details

Playing process. Each participant first plays general range 2 for 5 times, each time lasting 30 seconds. We have two objectives. First, to allow users to familiarize themselves with the controller through five relatively simple task scenarios. Second, to record the results of these five attempts to observe the learnability of the system Then, participants will play general 1, fine, and board for once. We test only once in other scenarios because both GELLO (the joint copy system) and ACE are highly intuitive teleoperation systems. We aim to see if users can quickly adapt when encountering a new scenario for the first time. Additionally, we want to observe the performance of the joint copy system and the ACE system in task scenarios with varying precision requirements.

Grouping. To ensure the fairness of the experiment and the comparability of the results, targets appear in the same positions within the identical workspace across tests. Familiarity with the testing simulation could also affect the outcomes. Therefore, we divided the six testers into two groups. One group completed the full ACE test before the *GELLO* test, while the other group did the opposite.