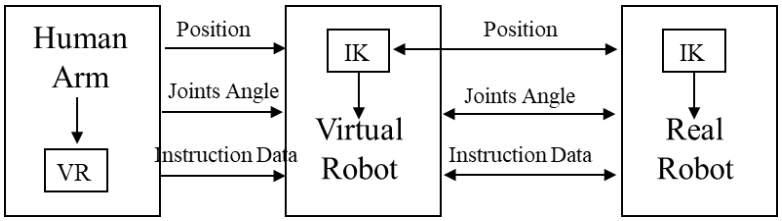
This system needs to control the virtual robot arm and the real robot arm, including three control modes: pose, joint angle and command control, as shown in the figure below. Since the pose and joint angle control of the real manipulator can be completed through the manipulator controller, this section mainly introduces the control mode of the virtual manipulator and the command control of the manipulator.



(1) On the control of the virtual manipulator, first specify the rotation axis of each joint, and achieve the effect that the end of the virtual manipulator rotates with the rotation of the front joint by setting the parent-child relationship of the joint. That is, the front joint acts as the parent node of the rear joint, and when the current joint moves, it will drive the motion of the rear joint, eventually leading to the follow-up of the end, and completing the motion control of the virtual manipulator. In addition, in order to synchronize the motion of the virtual robot arm with the rendering frequency, we strictly control the time of each frame by setting the *FixedUpdate* speed in Unity3D, and assign each joint a corresponding angle in each frame, so as to achieve precise control in time. See the actual code for details DT-Alita/REI_DT/DT_unity

(2) On the command control of the robot arm, we can realize remote control of the robot arm through the official library *Python urx*, which includes two ways: one is to directly control the robot arm through the joint or pose mapping, which can be realized by directly calling the corresponding functions in *Python urx*; The second is to control the robot arm through the atomic action. At this time, URScript script needs to be written according to the atomic action characteristics and passed to the robot arm to execution. Because Robotiq's wrist camera has the advantage of accurately identifying and tracking objects, and can complete tasks such as grasping by writing URScript scripts compatible with UR robot arm and Robotiq gripper, the platform uses a wrist camera at the end of the physical robot arm to identify and grab target objects and make path planning. The physical picture is shown in the following figure.

The following figure shows the semantic instructions represented by URScript scripts. First, the wrist camera is used to locate the target object, and the path close to the target object is planned based on MoveJ to complete the approach operation; Then define the grab point of the object based on MoveL, and control the opening

and closing distance of the gripper to complete the grab operation. Similarly, move together and place operations can also be realized through these instructions; The hold operation can be expressed as keeping the grasping action until it changes; The operation of pour into and stir can be completed through joint angle 4 and joint angle 6. It should be noted that the target object can be replaced with any other calibrated object, as shown in Figure (b). Moreover, the above grab point takes the target object itself as the reference point and does not change with the change of object position. Therefore, when the object position changes, the gripper will not be affected to grab. Therefore, it is possible to control various atomic action commands of the robot arm by writing URScript scripts with different objects in the robot server.