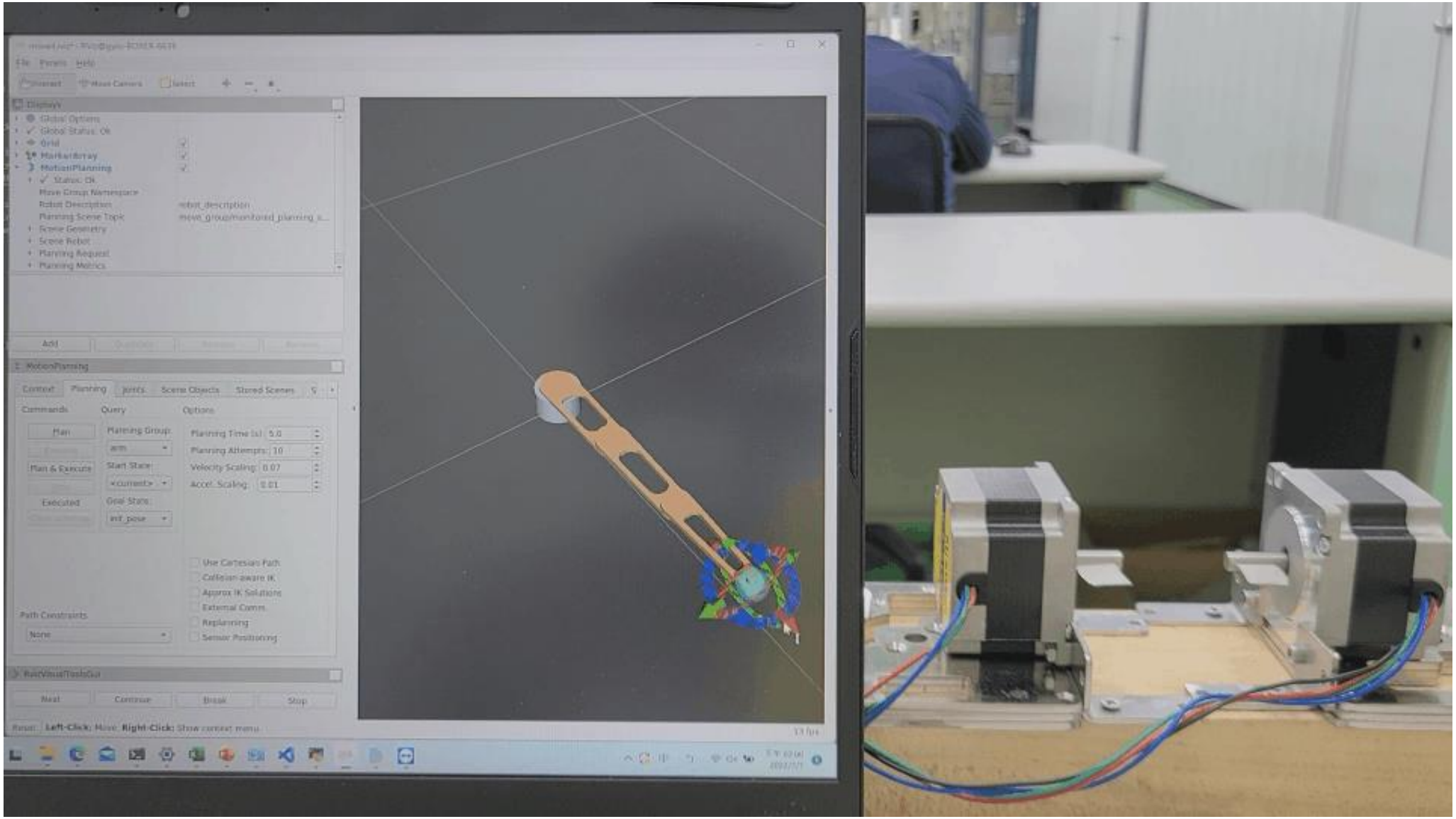
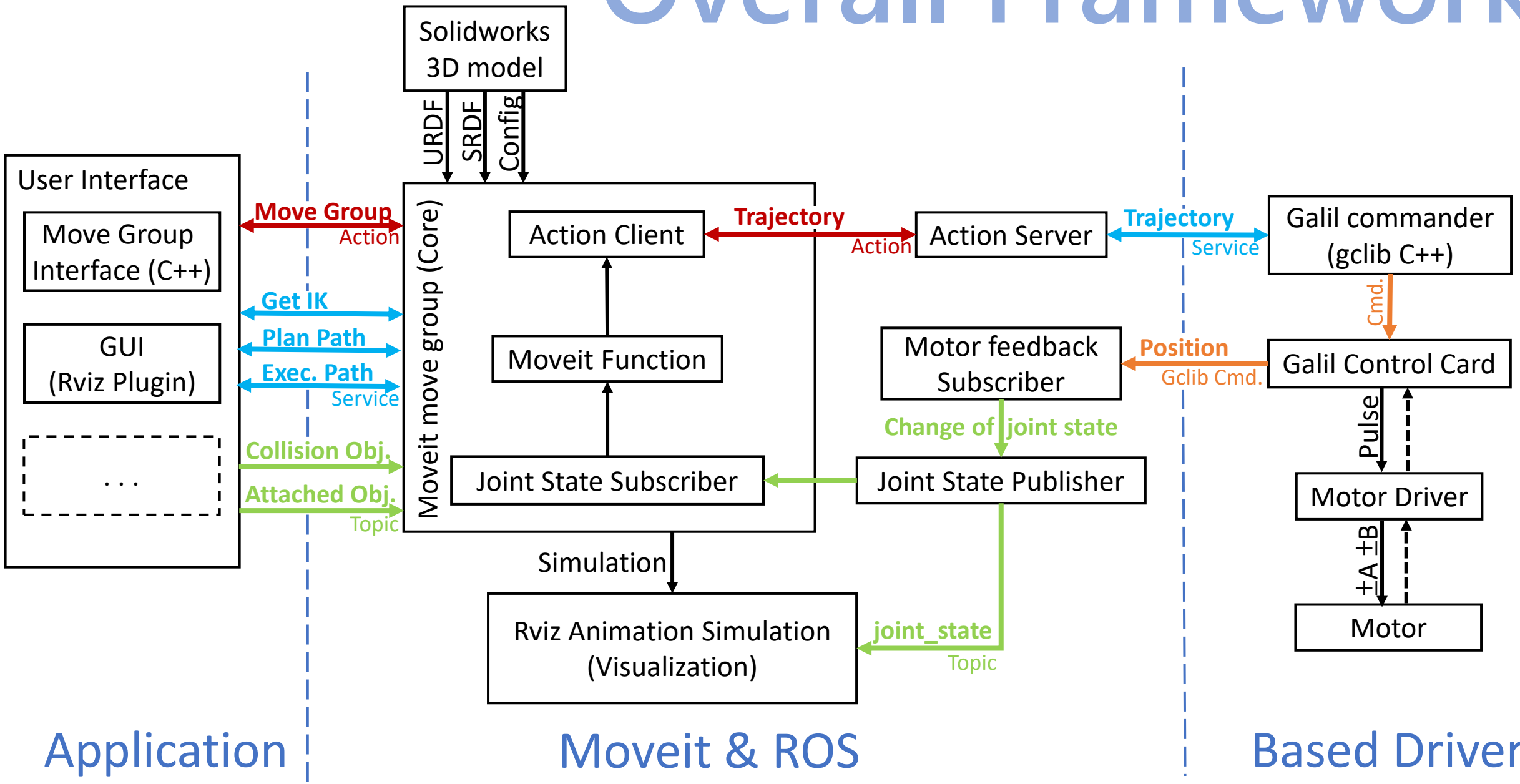
Robot Arm Development

With Moveit motion planner and Galil controller

Goal

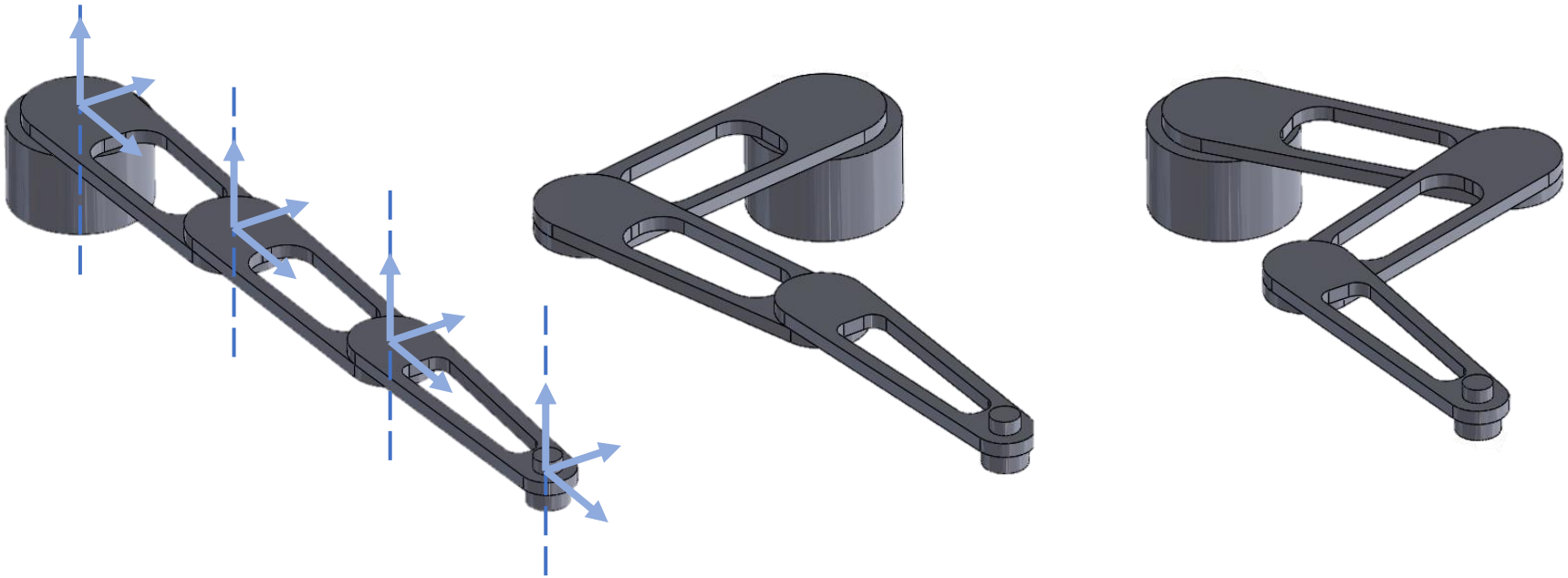

Displays
Global Options
Global Status: Ok
Grid
MarkerArray
MotionPlanning
Status: Ok
Move Group Namespace
Robot Description
robot_description
Planning Scene Topic
move_group/monitored_planning_s...
Scene Geometry
Scene Robot
Planning Request
Planning Metrics
Add
MotionPlanning
Context
Planning
Joints
Scene Objects
Stored Scenes
Commands
Query
Options
Plan
Planning Group:
arm
Plan & Execute
Start State:
<current>
Executed
Goal State:
init_pose
Planning Time (s): 5.0
Planning Attempts: 10
Velocity Scaling: 0.07
Accel. Scaling: 0.01
Use Cartesian Path
Collision-aware IK
Approx IK Solutions
External Comm.
Replanning
Sensor Positioning
Path Constraints
None
Next
Continue
Break
Stop
Left-Click: Move
Right-Click: Show context menu

Overall Framework



Solidworks Ass. to URDF

- Install **SolidWorks to URDF Exporter** from github.

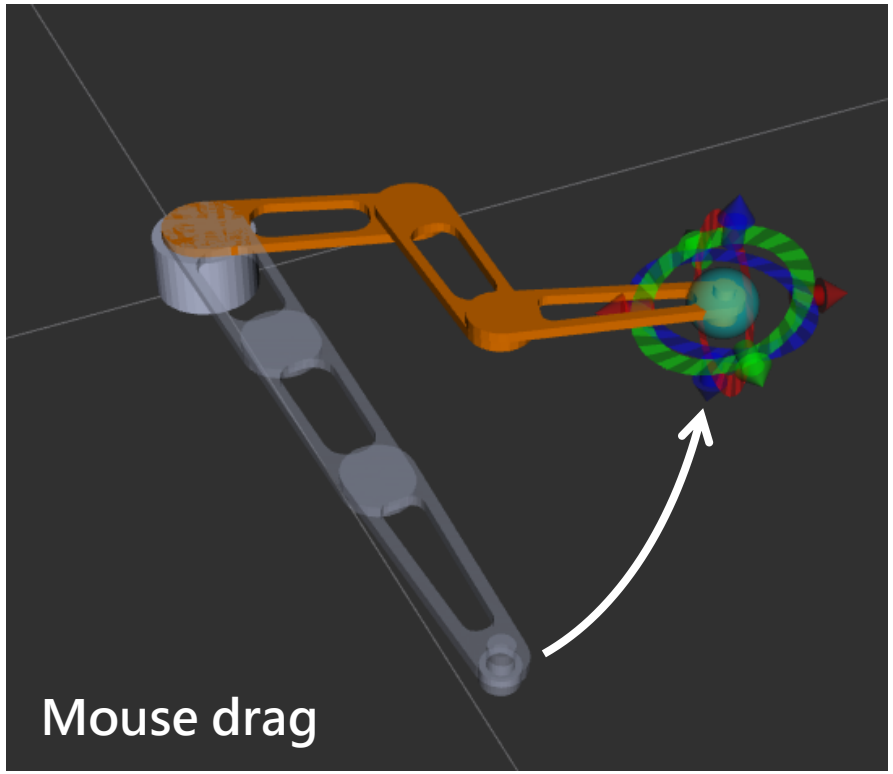


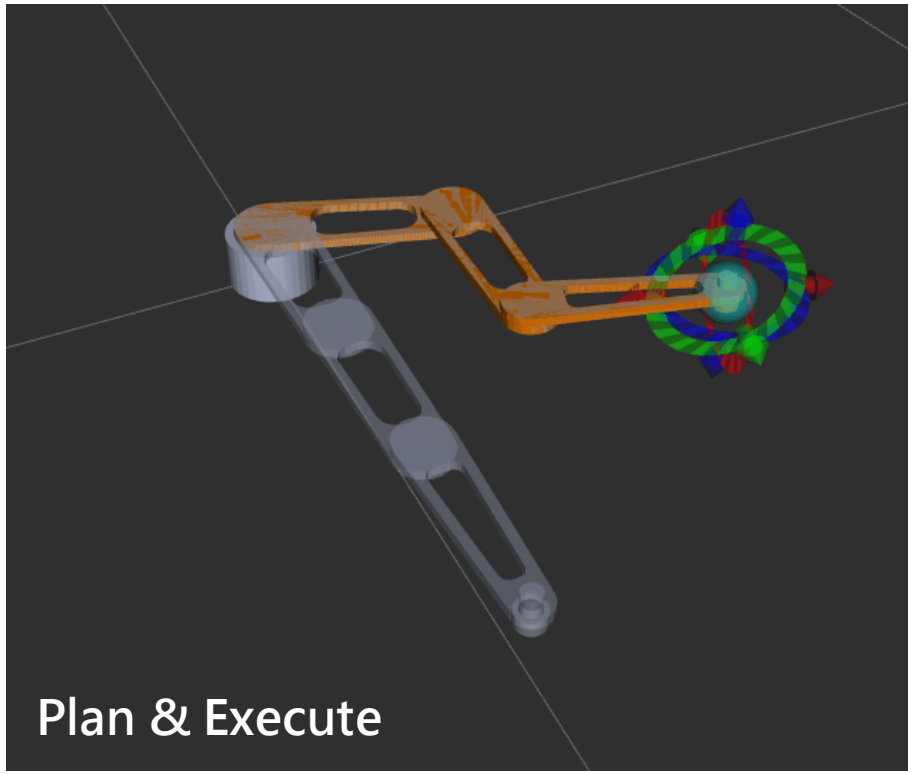
- Define Rotation **Axes**, **Coordinates** and **Limits** of each joints.
- Follow the sw2urdf GUI, and a ROS package with Rviz demo will be created automatically.

Solidworks to URDF tutorials

Inverse Kinematic Solver

- Use Moveit setup assistant to create a Moveit package and choose **KDL Kinematics Plugin** as the kinematics solver.
- Drag the end of the arm to the target position and click "Plan & Execute"

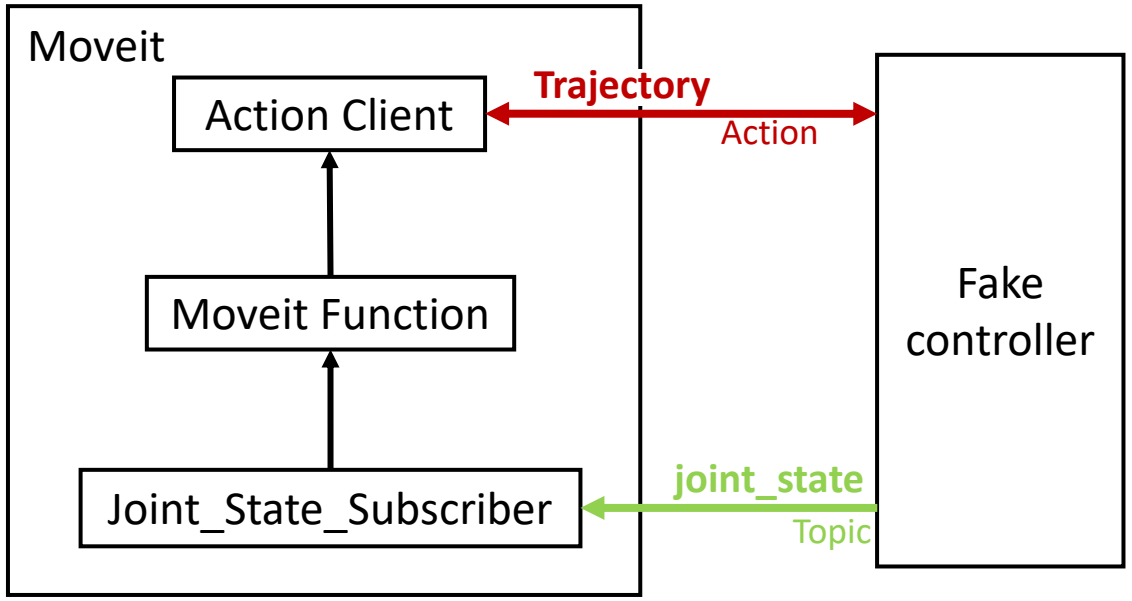




Moveit setup assistant tutorials

Moveit

- By default, Moveit run on a **fake controller** to simulate motion on a virtual robot
- Substitute the fake controller for our own robot controller to control the real-world robot
- Create an **action server** to process the trajectory message



Joint Trajectory

- Moveit generate a trajectory with specific positions, velocities and accelerations
- Moveit send planned joint trajectory via **action**
- Message type :

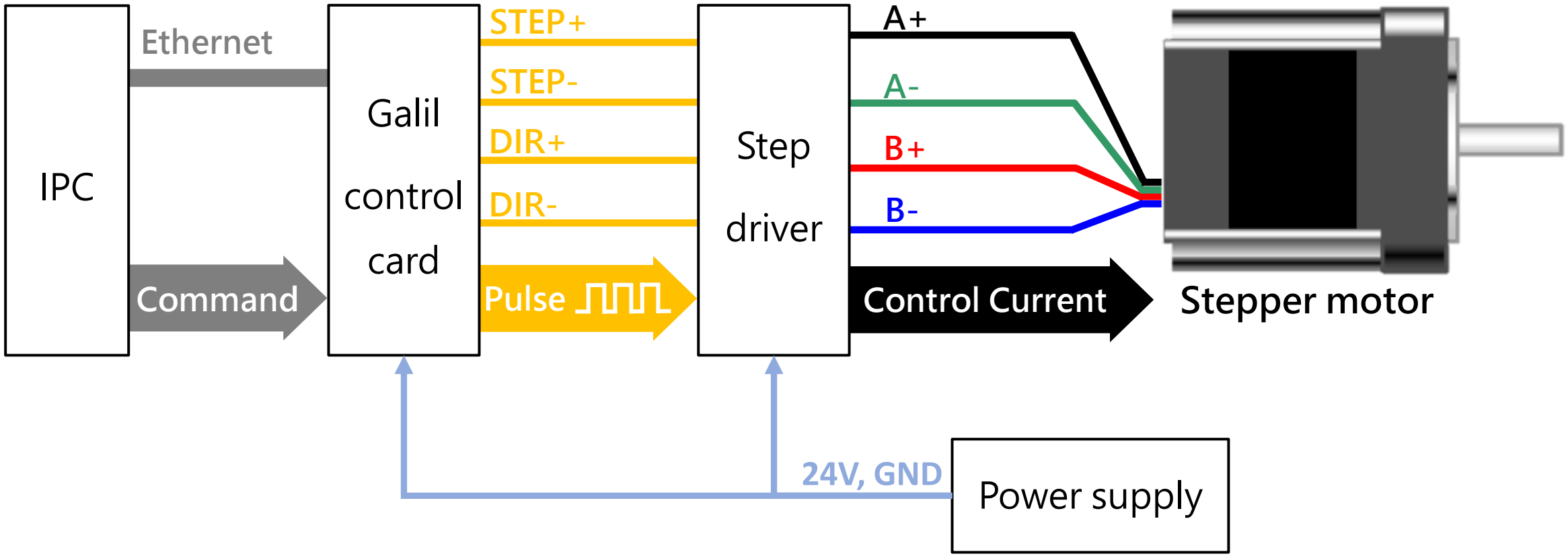
trajectory_msgs/JointTrajectoryPoint

float64[] positions
float64[] velocities
float64[] accelerations
float64[] effort
duration time_from_start

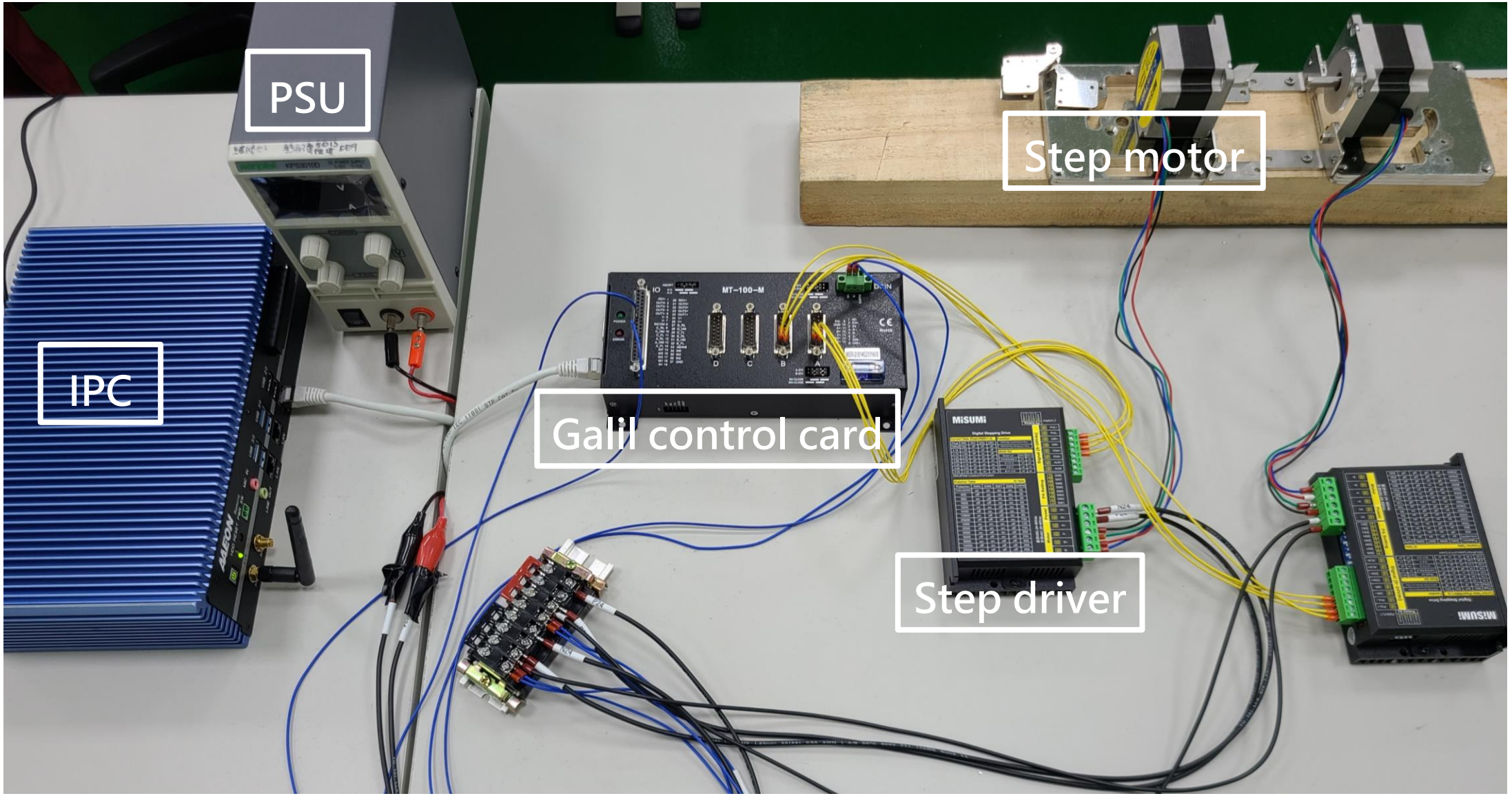
trajectory_msgs/JointTrajectoryPoint

Based Driver

- Hardware installation

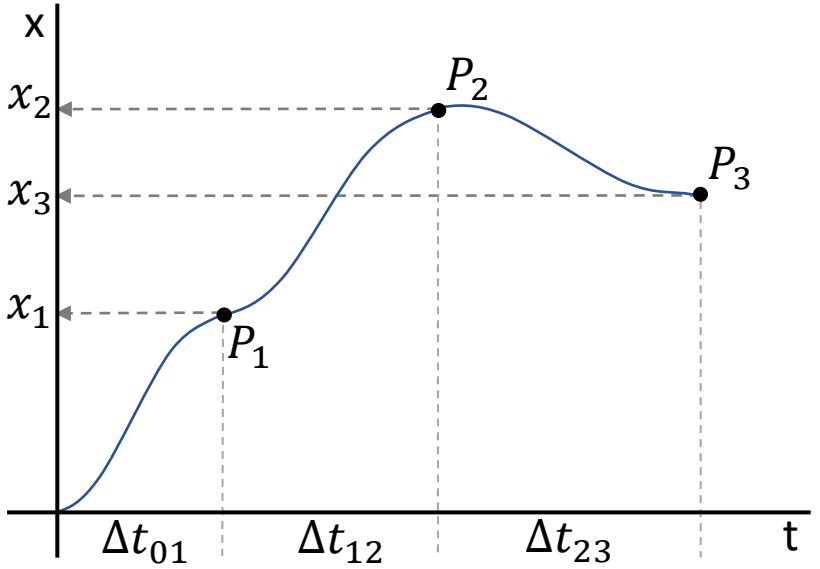


Based Driver



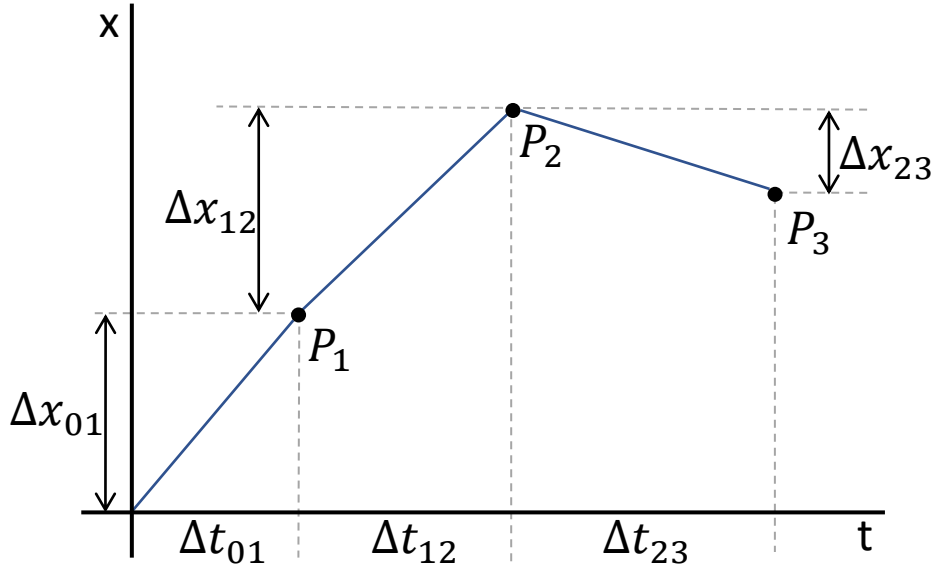
Galil Controller Solutions

- Position tracking




```
Cmd(g, "PT 1");//start pos. tracking mode
Cmd(g, "PT x1");//assign an absolute position
//motion start immediately after a PT command
Sleep(Δt01);//wait motion to complete
Cmd(g, "PT x2");//new target position
Sleep(Δt12);
Cmd(g, "PT x3");
Sleep(Δt23);
Cmd(g, "PT 0");//end of pos. tracking mode
```

- Contour motion

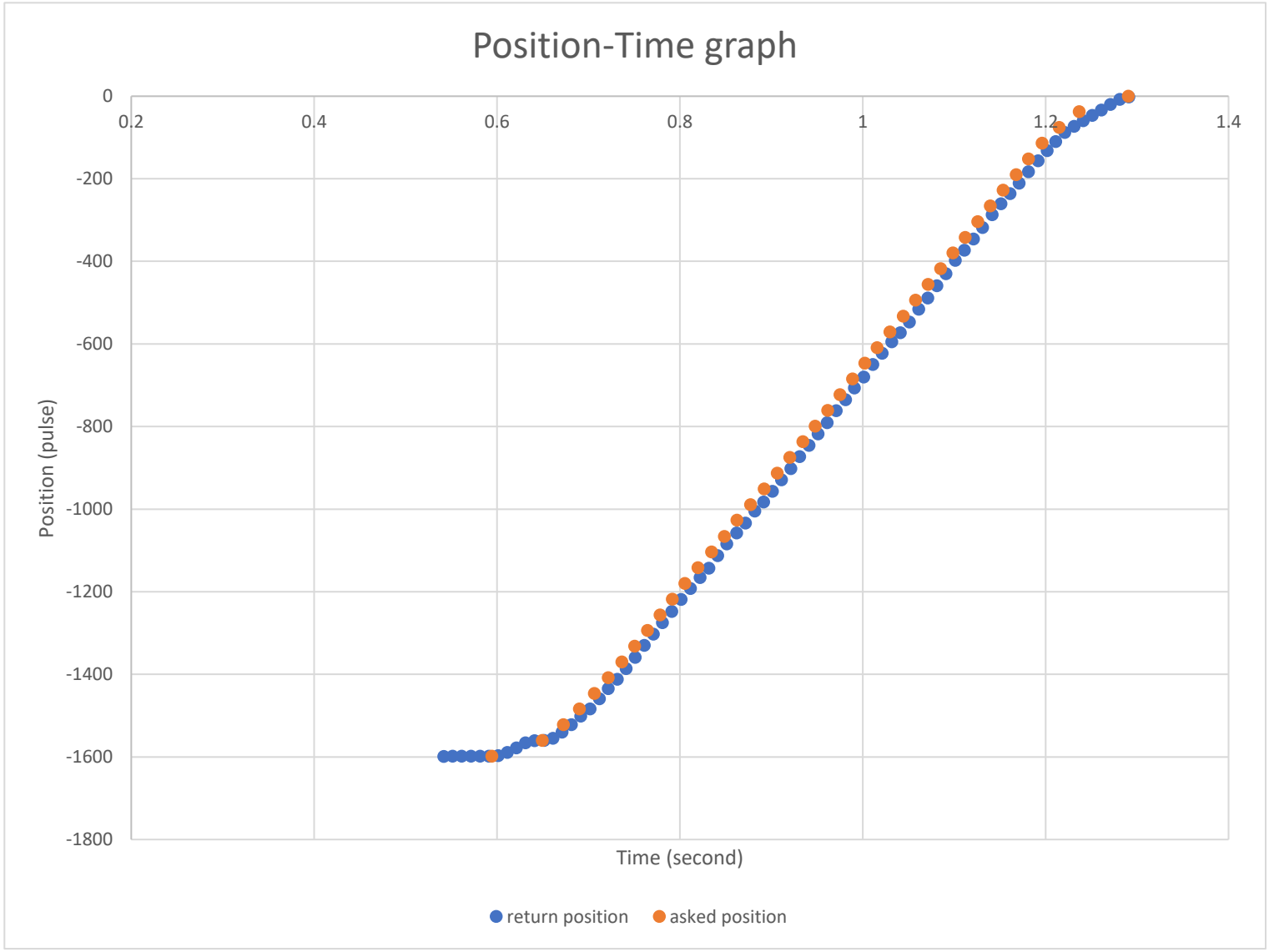



```
Cmd(g, "CM X");//start contour mode on X-axis
Cmd(g, "CD Δx01 = Δt01");
//write displacement and time step into the buffer
Cmd(g, "CD Δx12 = Δt12");
Cmd(g, "CD Δx23 = Δt23");
Cmd(g, "CD 0=0");//end of contour mode
//motion start after this command
```

Command Reference

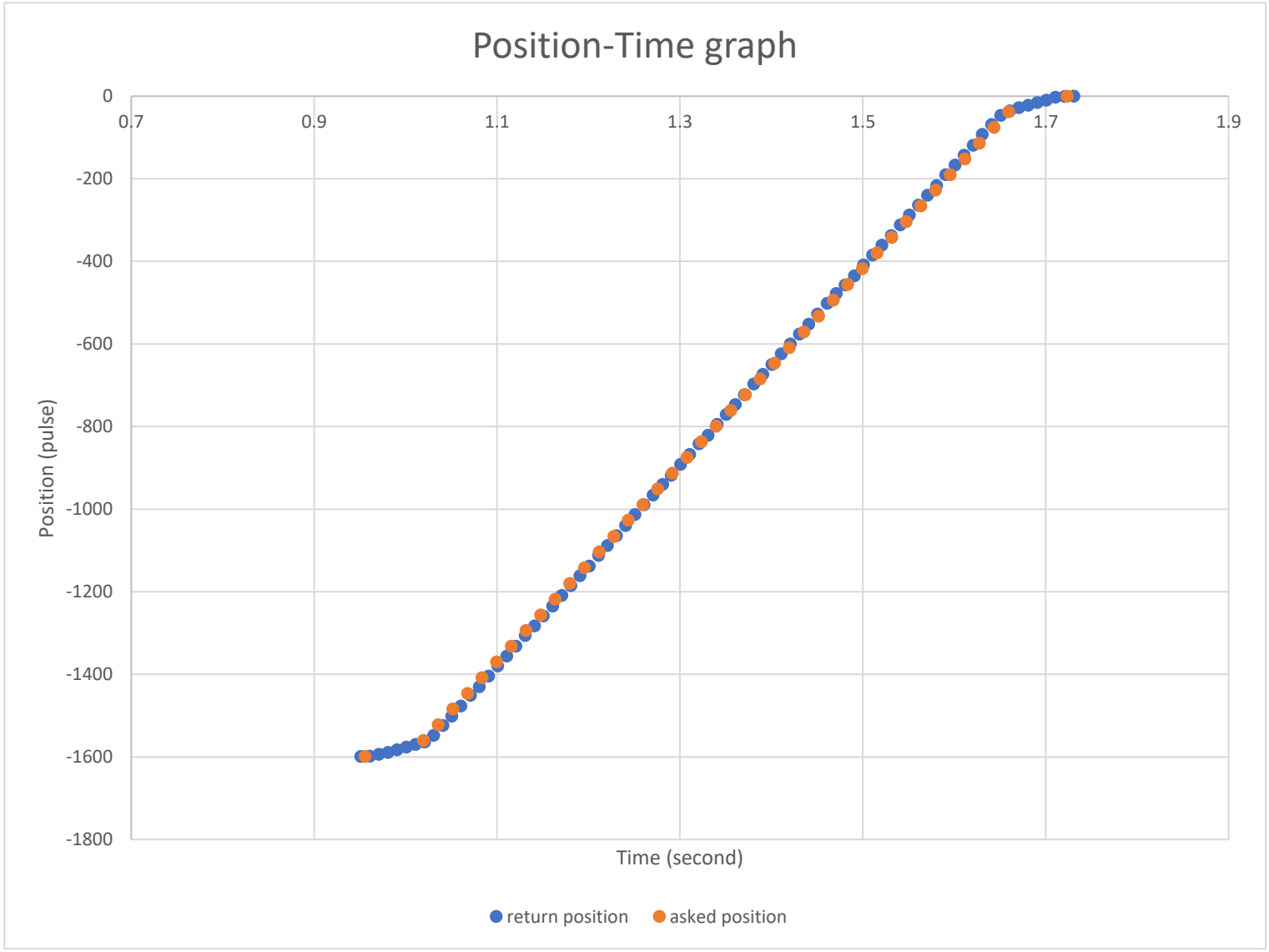
Error Analysis

- Position tracking



Error Analysis

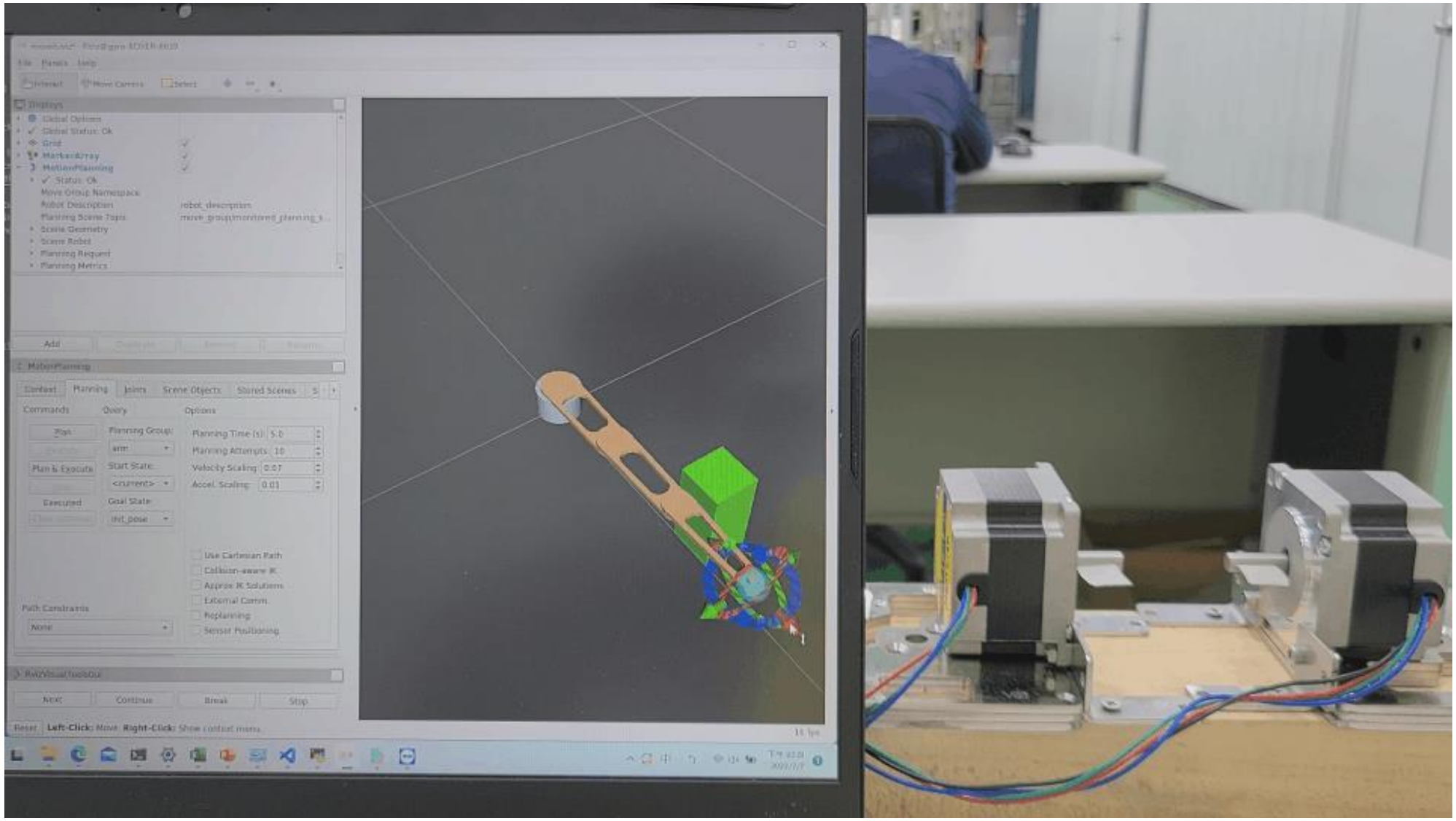
- Contour motion



Comparation

- Position tracking
  - Pros :
    - Control card planning, accurate and stable
    - Less error
  - Cons :
    - Only 32 points can be stored into the buffer
    - Time interval need to be a power of 2 (2, 4, 8, …, 256)
- Contour motion
  - Pros :
    - Command one point at a time, so there is no limit to the number of planning points
    - Can change goals during the motion
  - Cons :
    - Accumulative error may occur
    - The result may be different each time on the same input trajectory

Obstacle Avoidance



Obstacle Avoidance

- Move Group C++ Interface allows us to use API to create applications, such as collision detection, obstacle avoidance, and object attachment.
- Add a collision object to the world, and then drag the end of the arm to the target position again. Moveit will generate a trajectory bypassing the obstacle.
- Rviz visual tool allows us to add and erase collision objects on the GUI panel.

Move Group C++ Interface

Full Demo

- demo (streamable.com)

References

- sw_urdf_exporter/Tutorials - ROS Wiki
- 【ROS學習】Solidworks模型轉化為URDF檔案格式+三連桿機械臂示例+逆運動學
- MoveIt Setup Assistant — moveit_tutorials Noetic documentation (ros-planning.github.io)
- ROS中Moveit生成的轨迹如何作用与实际的机械臂(二)
- trajectory_msgs/JointTrajectoryPoint
- Galil Gclib command reference