

move_control.cpp

This example demonstrates how to switch robot modes and execute planned motion commands.

This example demonstrates how to switch robot modes and execute planned motion commands. The robotic arm's default mode is `RobotMode::PLANNING_POS`. To execute different types of motion commands, the robot mode must be switched accordingly before sending commands.

This example covers:

- Switching back to planning position mode (`RobotMode::PLANNING_POS`) for planned joint and Cartesian motions.
- Sending joint position commands (in radians) and Cartesian position commands (position in meters and orientation as quaternion).

Note:

- All joint control commands require joint angles ordered from joint 1 to joint 6.
- Cartesian positions require XYZ order for position and XYZW order for orientation quaternion.

Example usage:

```
#include "airbot_sdk.hpp"
#include <array>
#include <cmath> // for M_PI_4
using namespace airbot;

int main() {
    auto sdk = std::make_shared<AirbotSdk>("192.168.0.xxx", 50051);

    // Switch mode to planning position for planned movements
    sdk->SwitchMode({"arm"}, {RobotMode::PLANNING_POS});

    // Define target joint positions (radians) for joints 1 to 6
    std::array<double, 6> initial_pos = {0.0, -M_PI_4, M_PI_4, 0.0, 0.0, 0.0};
    if (sdk->MoveToJointPos(initial_pos)) {
        sdk->GetLogger()->info("Movement to joint position initiated successfully");
    } else {
        sdk->GetLogger()->warn("Movement to joint position failed.");
    }
}
```

```
// Define target Cartesian pose: position (meters) and orientation quaternion
const std::pair<std::array<double, 3>, std::array<double, 4>> &target_pose{
    {0.45, 0.20, 0.45}, {0.0, 0.0, 0.0, 1.0}};
if (sdk->MoveToCartPos(target_pose)) {
    sdk->GetLogger()->info("Movement to Cartesian position initiated successfully");
} else {
    sdk->GetLogger()->warn("Movement to Cartesian position failed.");
}

return 0;
}
```