

# servo\_joint\_pos.cpp

This example demonstrates streaming control using SERVO\_JOINT\_POS mode at 250 Hz.

This example demonstrates streaming control using SERVO\_JOINT\_POS mode at 250 Hz.

## Key Concepts:

- The robot streams joint position and gripper opening in real-time.
- Recommended control rate is 250 Hz ( $\approx$  4ms cycle).
- This example creates a smooth oscillatory motion using a sine function.

## Notes:

- Before switching to `SERVO_JOINT_POS`, the robot must move to a valid pose using `PLANNING_POS`.
- Joint position and gripper commands are updated every 3ms.
- A phase shift is used to synchronize the sinusoidal motion of the arm and gripper.

## Example usage:

```
#include "airbot_sdk.hpp"
#include <array>
#include <cmath>
#include <thread>
#include <chrono>
using namespace airbot;

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

    // Step 1: Move to initial pose with PLANNING_POS
    std::array<double, 6> initial_pos = {0.0, -M_PI_4, M_PI_4, 0.0, 0.0, 0.0};
    sdk->SwitchMode({"arm"}, {RobotMode::PLANNING_POS});
    sdk->MoveToJointPos(initial_pos);

    // Step 2: Switch to SERVO_JOINT_POS mode
    sdk->SwitchMode({"arm"}, {RobotMode::SERVO_JOINT_POS});

    constexpr double total_duration_sec = 30.0;      // Total streaming time
    constexpr double one_cycle_duration_sec = 10.0;  // One full sine cycle du
```

```

auto start_time = std::chrono::steady_clock::now();

while (true) {
    auto now = std::chrono::steady_clock::now();
    std::chrono::duration<double> elapsed = now - start_time;

    if (elapsed.count() >= total_duration_sec)
        break;

    // Step 3: Compute sinusoidal control values
    double phase_shifted_time = elapsed.count() - 0.5;
    double timed_coef = std::sin(phase_shifted_time * 2 * M_PI / one_cycle_dur);

    std::array<double, 6> joint_pos = {
        M_PI / 2 * timed_coef,
        -M_PI / 4 - M_PI / 4 * timed_coef,
        M_PI / 4 + M_PI / 4 * timed_coef,
        0.0, 0.0, 0.0
    };

    std::vector<double> eef_pos = {(1 - timed_coef) * 0.07 / 2}; // Gripper c

    // Step 4: Send streaming commands
    sdk->ServoJointPos(joint_pos);
    sdk->ServoEefPos(eef_pos);

    std::this_thread::sleep_for(std::chrono::milliseconds(3)); // ~333Hz loop
}

// Step 5: Wait for final movement to finish (optional)
std::this_thread::sleep_for(std::chrono::seconds(static_cast<int>(total_dura

return 0;
}

```