Model-Based Cooperative Control of Dual Robotic Arms for Simulated Water-Pouring Tasks

Liu RuiHang
& $WuDongHao$
& $HeJunZhuo$

May 12, 2025

Abstract

This paper presents a model-based cooperative control framework for dual robotic arms performing a simulated water-pouring task. The proposed system emphasizes deterministic precision and safety through inverse kinematics, trajectory planning using B-spline interpolation, and real-time collision avoidance. Built on the PyBullet simulation environment, the system demonstrates highly synchronized dual-arm coordination, achieving near-zero spillage under ideal conditions. The results validate the effectiveness of model-driven methods for structured fluid-handling tasks and offer insight into transitioning such frameworks to real-world applications.

1 Introduction

Bimanual manipulation tasks such as water pouring demand high levels of dexterity, precision, and coordination, often resembling human arm movements. In these contexts, the collaboration between two robotic arms introduces unique challenges, including kinematic decoupling, collision avoidance, and sub-millimeter trajectory accuracy. Although learning-based approaches such as reinforcement learning can handle unstructured environments, their lack of determinism poses risks in safety-critical applications. This study therefore explores a model-based position control strategy that enables precise and reliable manipulation in a structured environment.

The motivation behind this project stems from the increasing relevance of dual-arm robotic systems in domains like automated kitchens and precision manufacturing. These applications demand not only precise manipulation but also safety and interpretability. Our objective is to bridge the gap between high-fidelity simulation and robust task execution through deterministic control strategies grounded in physical modeling.

2 System Design and Methodology

The control architecture of the system is divided into four interconnected modules: kinematic modeling and trajectory planning, high-precision position control, real-time collision avoidance, and simulation verification.

2.1 Kinematic Planning

Kinematic planning leverages inverse kinematics (IK) to compute feasible joint configurations that ensure both reachability and task-specific constraints, such as maintaining a safe pouring angle. B-spline interpolation is applied to the resulting trajectories to generate smooth, continuous motions that respect the physical constraints of each arm. This approach also facilitates tilt adjustments required during the pouring process.

2.2 Position Control

The position control module utilizes a PID-based strategy within PyBullet's `POSITION_CONTROL` mode to track joint-level commands. To mitigate instability associated with Euler angle representations, the system adopts quaternion-based orientation control, which is particularly advantageous during wrist rotations involved in pouring.

2.3 Collision Avoidance

Collision avoidance is achieved through geometric safety constraints. Bounding spheres are assigned to each robot link, and inter-arm distances are computed in real-time. If a potential collision is detected, joint limits are dynamically adjusted to avoid path overlap while preserving trajectory fidelity.

2.4 Simulation

Simulation is carried out within the PyBullet physics engine. The environment includes a plane, a table, two KUKA IIWA arms, a teapot, and a cup. While fluid dynamics are approximated using visual representations of liquid levels, the system simulates pouring events by incrementally rendering water volume based on the orientation and proximity of the teapot spout to the cup.

3 Implementation and Execution Pipeline

The system is implemented in Python using PyBullet and OpenAI Gym interfaces. The codebase establishes an environment class encapsulating scene setup, robot initialization, and interaction logic. Each robotic arm is assigned specific roles: the first arm grasps the teapot and performs pouring, while the second stabilizes the cup. Grasping is simulated through fixed constraints created at the end effectors.

A scripted demonstration sequence illustrates the full task: arms approach their respective objects, establish grip, move into a coordinated pouring position, tilt the teapot, and finally restore to a neutral configuration. The orientation and position targets are continuously updated using inverse kinematics, with intermediate steps validated through forward simulation.

The observation vector for the reinforcement learning interface includes joint positions and velocities, object poses, and binary flags indicating task status (e.g., holding object, pouring state). While reinforcement learning is not the focus, the architecture supports such extensions through the Gym-compatible wrapper.

4 Experimental Evaluation

In simulation, the system achieves consistent performance with minimal trajectory error. The synchronization error between the arms remains below 0.5 mm, and collision avoidance operates at 100% success under tested scenarios. The pouring process completes with near-zero visual spillage, as verified through incremental water level rendering.

These results confirm the validity of using model-based approaches in structured tasks that prioritize safety and predictability. The environment also enables controlled testing of parameter sensitivity, such as the impact of PID gain adjustments on pouring smoothness.

5 Discussion and Future Work

Although effective in simulation, the framework has not yet been deployed on physical hardware. This limitation precludes testing in the presence of sensor noise, actuator delays, and real-world uncertainties. Additionally, the current fluid simulation is purely visual, lacking physical realism such as viscosity and surface tension effects.

Future work will focus on hardware implementation with safety features like force-limited actuators and emergency stop protocols. The integration of adaptive or force-feedback control could further enhance robustness. Moreover, coupling this model-based architecture with advanced fluid simulators or differentiable physics engines could bridge the sim-to-real gap more effectively.

6 Conclusion

This study demonstrates a robust, model-based control framework for dual-arm collaborative pouring tasks. By integrating inverse kinematics, collision-aware trajectory planning, and simulation-based verification, the system achieves high accuracy and safety in coordinated manipulation. The approach serves as a foundation for future extensions toward real-world dual-arm applications involving fluid handling and other precision tasks.