ECE 276C: Project
Co-operative Control and Trajectory Optimization of two synchronous Panda robots

Arnav Wayade
A59013263
Department of Mechanical and Aerospace Engineering
University of California San Diego
awayade@ucsd.edu

Chaitanya Tambat
A59013275
Department of Mechanical and Aerospace Engineering
University of California San Diego
ctambat@ucsd.edu

Abstract - The interest in robotics has been on the rise because of the practical value it provides. Due to their very nature, robots are used for tasks that are repetitive, require a high degree of precision, and for tasks where it is unsafe for humans to work. The problem of localization and control of a robot to perform certain tasks has many novel uses in Modern Robotics. A specific utility of robots is for picking up and placing an object in a specified location and orientation while using multiple robotic arms such as in an industrial setting. A robot that can perform this task of pick-up-and-place is the Franka Emika Panda robot. This project discusses and provides a solution to control the motion of two Panda robots to perform a certain task. This project will run the simulation of the robot control in Pybullet.

Keywords— Franka Emika Panda Robot, Pybullet, Trajectory, End-effector, Transformation matrices, Forward and Inverse Kinematics, Optimal Control, Cooperative Control, Velocity Control

I. INTRODUCTION

Pick and place robots are commonly used in modern manufacturing environments. Pick and place automation speeds up the process of picking up parts or items and placing them in other locations. Pick and place robots handle repetitive tasks while freeing up human workers to focus on more complex work. The Franka Emika Panda robot is one such type of robot designed to perform the above specified task.

The Franka Emika Panda robot is an advanced collaborative robot designed to work alongside humans in various industrial and research settings. Developed by Franka Emika, a German robotics company, the Panda robot is known for its versatile and dexterous capabilities. With seven degrees of freedom, it offers a high level of flexibility and precise control in its movements. The robot's design includes sensitive torque sensors in each joint, allowing it to interact safely and intuitively with its environment and human counterparts. Equipped with a wide range of sensors, the Panda robot can perceive its surroundings, making it capable of performing complex tasks with accuracy. Its user-friendly interface and intuitive programming enable users to quickly teach the robot new skills and adapt it to different applications. The Franka Emika Panda robot has proven to be a valuable tool in industries such as manufacturing, assembly, and research, facilitating increased efficiency, productivity, and collaboration between humans and robots. This project focuses on employing two Panda robots to pick up, handover and place a block from its current location and place it at its desired location. This baseline will be researched and upgraded to an optimal control problem to find the best location on the trajectory to find the best exchange point for both robots to handover the cube such that manipulability of both robots and exchange is smooth. The figure below shows the Panda robot which is a 7 DOF robot.

Fig 1: Franka Emika Panda Robot

As we see above, the Panda robot has seven revolute joints that correspond to its seven degrees of freedom. Additionally, it has seven links which contribute to the seven DOFs.

II. BACKGROUND

Cooperative control of two robotic arms refers to the coordinated operation of multiple robots working together to accomplish a common goal. This approach enables increased efficiency, flexibility, and versatility in a wide range of applications across industries such as manufacturing, logistics, healthcare, and more.

In cooperative control, the two robotic arms are designed to communicate and synchronize their actions to perform tasks that require collaboration or simultaneous actions. This can involve tasks such as assembly, manipulation of objects, lifting heavy loads, or even performing complex surgical procedures. By combining the capabilities of multiple robotic arms, the overall system can accomplish tasks that would be challenging or impossible for a single robot to handle alone.

The coordination between the robotic arms can be achieved through various methods. One approach is centralized control, where a central controller manages and coordinates the actions of both robots. The central controller receives input from sensors and algorithms, calculates the desired trajectories and movements, and sends commands to each robotic arm. This allows for precise synchronization and cooperation between the arms, ensuring smooth and efficient execution of tasks.

Another approach is distributed control, where each robotic arm has its own local controller and intelligence. In this setup, the robots exchange information and collaborate through communication protocols to achieve the desired coordination. This distributed control method offers advantages such as increased fault tolerance, scalability, and adaptability, as each robot can make autonomous decisions based on local information.

Cooperative control of two robotic arms brings several benefits. It can significantly improve productivity by dividing complex tasks between the arms, reducing the overall execution time. The robots can also share the workload, enabling them to handle heavier objects or perform tasks that require strength and precision simultaneously. Moreover, cooperative control enhances flexibility and adaptability, as the robotic arms can adjust their actions in response to changes in the environment or task requirements.

Overall, cooperative control of two robotic arms opens up new possibilities for automation, allowing for more complex and challenging tasks to be accomplished with precision and efficiency. As technology continues to advance, we can expect to see further advancements in cooperative control, leading to even more sophisticated and capable multi-robot systems in the future.

III. METHODS

For co-operative control of robotic manipulators multiple algorithms and methods can be used. These methods range from Motion Planning, Trajectory Planning, Linear Control, Non-linear Control, etc.

1. Let’s briefly discuss all of these methods.

A. Motion Planning

Motion planning algorithms for cooperative control can be based on different techniques such as geometric planning, optimization-based planning, or sampling-based planning. Geometric planning methods focus on finding feasible paths and trajectories for the robots, considering their geometric constraints and the workspace layout. Optimization-based planning aims to optimize specific criteria, such as energy consumption, time efficiency, or task completion. Sampling-based planning algorithms use randomized sampling to explore the configuration space and find collision-free paths for the robots.
The issue with using Motion Planning algorithms for high DOF robots is the computation time and power it might need. Additionally, the Configuration space for high DOFs robots is multi dimensional and complicated. Hence, we will explore methods further to implement co-operative control.

B. Trajectory Planning

Using trajectory planning for cooperative control of robotic manipulators offers several advantages but also comes with certain disadvantages.

One significant advantage is the ability to achieve precise and coordinated movements among multiple robotic arms. Trajectory planning algorithms consider the kinematic and dynamic constraints of each manipulator, allowing them to generate smooth and collision-free trajectories. This ensures that the robots can work together efficiently and accurately, reducing the risk of collisions or interferences during task execution. The synchronized movements facilitated by trajectory planning result in improved productivity, as the robots can complete complex tasks faster and with higher precision.

However, there are some disadvantages associated with trajectory planning for cooperative control. One significant challenge is the computational complexity involved in generating optimized trajectories for multiple robotic manipulators. As the number of robots increases, the complexity of finding feasible and coordinated trajectories grows exponentially. This computational overhead can limit the real-time capabilities of the system, potentially leading to delays or reduced efficiency in task execution.

C. Optimal Control

Optimal control techniques play a crucial role in enhancing the performance and efficiency of robotic manipulators. By applying optimal control principles to robotic manipulator systems, it becomes possible to compute control inputs that minimize energy consumption, reduce execution time, or achieve precise and accurate movements. Optimal control for robotic manipulators involves formulating an optimization problem that considers the system dynamics, constraints, and a defined objective function. The objective function may include criteria such as minimizing torque, maximizing manipulability, or optimizing a combination of factors. By solving this optimization problem, optimal control algorithms can generate control policies or trajectories that enable the robotic manipulator to achieve desired outcomes while satisfying constraints. This approach enables the robots to perform tasks more effectively and efficiently, resulting in improved productivity, accuracy, and resource utilization. Optimal control for robotic manipulators continues to be an active area of research, with ongoing developments focusing on incorporating various factors such as sensor feedback, safety considerations, and real-time adaptability to further enhance the performance and capabilities of robotic systems.

2. Baseline Method for Co-operative Control of the two Panda robots

The baseline for controlling the two Panda robots was focused on controlling both robots such that they are able to individually manipulate their end effectors to pick up, handover and place the robot in a different location.

Essentially, the robots will follow a decided trajectory to pickup the block, hand it over at a fixed point and the second robot will place it at the desired location.

We will discuss the Trajectory Generation baseline here.

In this part of the problem, we specify the initial configuration of the cube given by T0 and the final desired configuration of the cube T3.

Secondly, we specify the configuration of the end effector when grasping the cube as T1 and the robot handing over to the second robot as T2.

Similarly, we find the end effector configuration in the world frame using the final desired configuration of the cube and the grasp and standoff matrices using the same above formula.

Then we use the ctraj function from the Peter Corke Robotics library to generate interpolated transformation matrices for the end effector to move from one location to the next desired location.

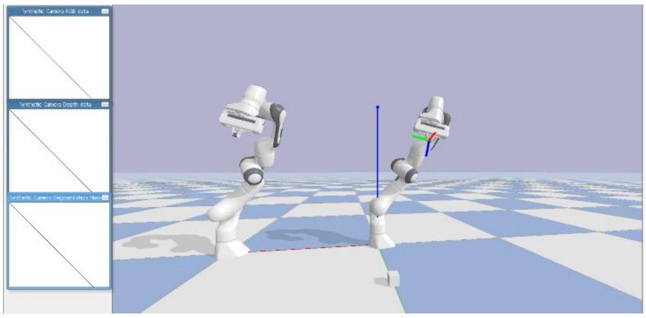

Fig 2: The two Panda robots and the cube in the Pybullet environment

3. Augmented Method for Co-operative Control of the two Panda robots

To upgrade the baseline method of exchange between robots, the exact exchange point on the planned trajectory for the object will be decided using the Optimal Control approach. The decision of where the object (block) will be exchanged is decided on the basis of manipulability of the two robotic arms and minimization of the power required for each arm to reach to desired locations. We will discuss this in detail in the following paragraphs.

The measure of optimization for the baseline method was set to be the manipulability of the two Panda robots. Manipulability of robotic manipulators refers to the measure of their ability to control and manipulate objects in their workspace. It quantifies the dexterity and versatility of a manipulator by considering its kinematic structure and the range of achievable positions and orientations.

1. Manipulability Jacobian (Jm):

The manipulability Jacobian relates the rate of change of manipulability to the rate of change of joint configurations. It provides insights into how variations in joint angles affect the manipulability of the arm. The formula for the manipulability Jacobian is:

$$Jm = \|J * \mathrm{inv}(J^T * J) * J^T\|$$

Where:

- J is the manipulator's Jacobian matrix.
- inv denotes the inverse operation.
- ||...|| denotes the Frobenius norm, which measures the magnitude of a matrix.

2. Manipulability Ellipsoid:
The manipulability ellipsoid represents the shape and orientation of the region in the workspace that the manipulator can effectively control. It provides information about the arm's ability to reach different positions and orientations. The formula for the manipulability ellipsoid is:

ME = sqrt(det(J * J^T)^(-1/n) * I)

Where:
- ME is the manipulability ellipsoid.
- J is the manipulator's Jacobian matrix.
- det denotes the determinant operation.
- n is the number of degrees of freedom of the manipulator.
- I is the identity matrix.

3. Manipulability Index:
The manipulability index is a scalar value that represents the overall manipulability of a robotic manipulator. It is calculated using the singular value decomposition (SVD) of the manipulator's Jacobian matrix. The formula for the manipulability index is:

MI = sqrt(|Σ|_max / |Σ|^2)

Where:
- MI is the manipulability index.
- Σ is the diagonal matrix containing the singular values of the Jacobian matrix.
- |...|_max denotes the maximum singular value.
- |...|^2 denotes the sum of the squared singular values.

These formulae provide quantitative measures to evaluate and optimize the manipulability of robotic manipulators, aiding in their design and control for efficient and effective performance in various applications.

Formulation of Manipulability in the code:
The manipulability of the entire system of robots was found using taking the Singular value decomposition of the Jacobians of both robots and multiplying them.
The point of maximum manipulability was then fixed to be the exchange point. We will look at the results which will explain how the augmented method worked.

IV. Results

Home position the environment
The start position of the robots and the cube are as shown below.
Cube start = [0, 0.5, 0.025]

Start position of end effectors
Robot1 = [0, 0.5, 0.5]
Robot2 = [1, 0.5, 0.5]

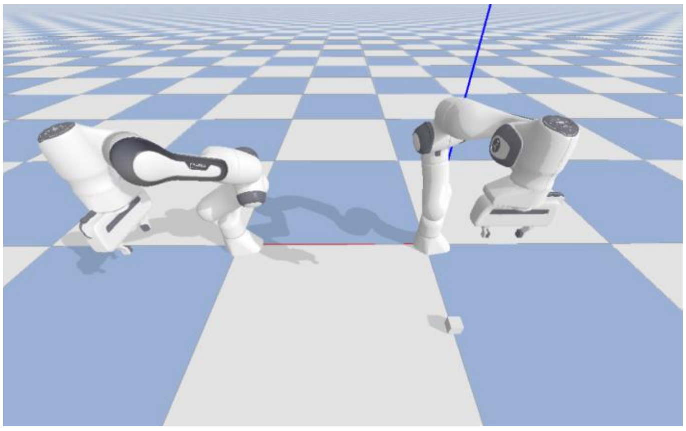

Fig 3: Home positions of the environment

Pickup position

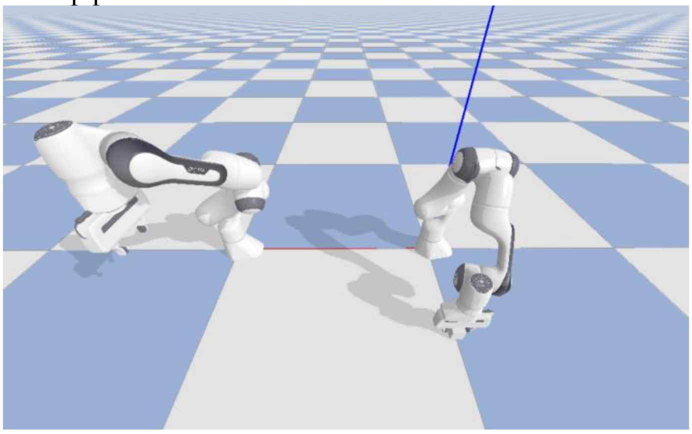

Fig 4: Pickup position

Handover position

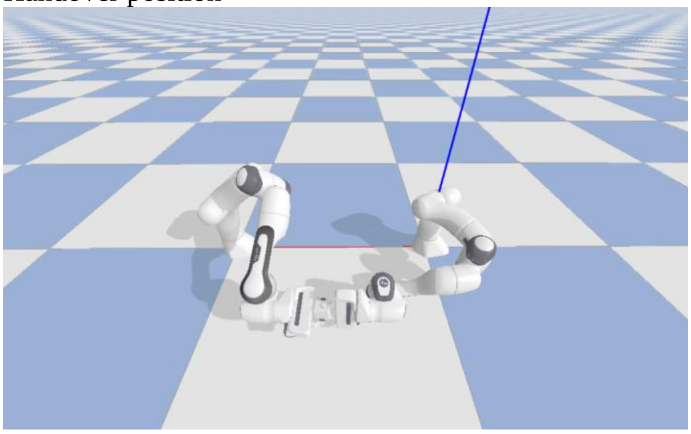

Fig 5: Handover position

From running the augmented method, the position of exchange was decided based on these results.

Maximum value of product of manipulability measures of both robots: 0.00813641192960433
Z coordinate value at maximum: 0.1500306140308610

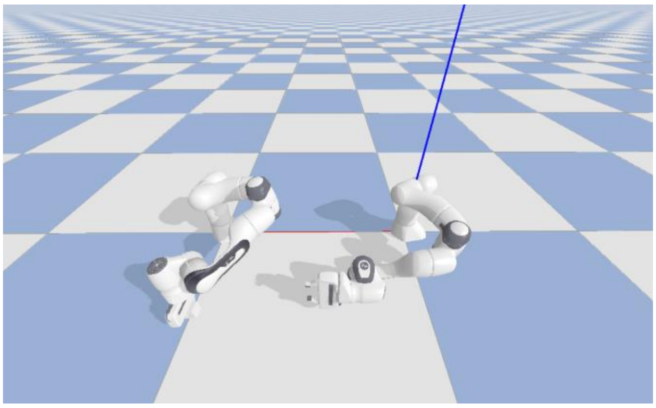

Fig 6: Place position

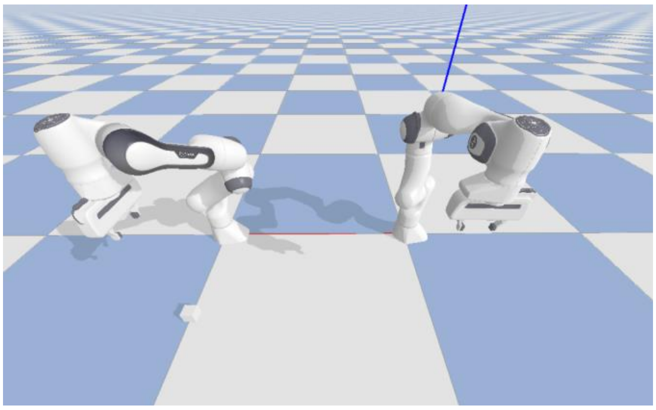

Fig 7: Rest position

Cube target coordinates = [1, 0.5, 0.025]

V. ACKNOWLEDGEMENT

The project was done with reference to lecture notes from ECE276C and following research papers. We would like to thank Prof. Michael Yip and TA Jacob Johnson for all the help and the whole class.