Developing controllers to enable robots to perform complex tasks

Shakeel Ahamed Mansoor Shaikna[1] and Venkatesh Prasad Venkataramanan[2]

***Abstract*— In this paper, we present an implementation of a method to control a team of two robots to work together to achieve a common objective. The robots are tasked with picking up cubes scattered across the floor and placing them inside a bin within the scene. The method uses the control methodology of Jacobian-based twist calculation, and we prove that this control methodology comfortably trumps the traditional inverse kinematics-based approach.**

I. INTRODUCTION

Control of robotic manipulators is one of the oldest, most well-researched topics in robotics. The first commercial use of robots can be traced back to the 1960s, when robots were first introduced in assembly line manufacturing to reduce human labour and automate monotonous tasks. Robotic manipulators, thus became a necessity for commercial manufacturing. Over time, there has been extensive research into coordinating a team of multiple robots to complete complex tasks.

In this work, we present one such task. The task involves two 7-DOF Franka Panda robots who have to scoop up cubes by working together and dump them in a bin placed in the scene. The robots have spades affixed to them, instead of traditional grippers, and hence this provides an interesting problem to solve. The robots have to work together to execute a forklift mechanism to be able to lift the cubes. It would not be possible for a single robot to be able to achieve this with a spade as the gripper and hence we are required to develop a very precise controller to be able to achieve coordination between these two robots.

The following sections provide the approach taken by the authors to tackle the problem. Section II deals with the problem formulation where we formally state the problem that we are trying to solve in mathematical terms. Section III would provide a detailed explanation of the method. Section IV would then detail the experiments which were conducted to prove that our method is in fact better than the inverse kinematics based approach, leading us to Section V where we would conclude our results and present possible future work.

II. PROBLEM FORMULATION

The problem formulation is broken down into the following two subsections.

A. *Inverse Kinematics based approach*

Given the forward kinematics $T(\theta)$ and the target pose $\mathrm{T}_{target} \in SE(3)$, find solutions θ that satisfy $T(\theta) = T_{target}$

B. *Jacobian Based Twist Calculation Approach*

Given the geometric jacobian and the twist, compute the angular velocity of each joint required to actuate the robot to achieve the required force.

$$\xi = \begin{bmatrix} \omega \\ v \end{bmatrix} = J(\theta)\dot{\theta}$$

Both these approaches are solved in very different ways, and a detailed description of their implementation is described in the following sections.

III. TECHNICAL APPROACH

The technical approach is broken down into the following subsections.

A. *Outline of Robot Maneuver*

1) Use of Visual Information: For this task, the scene has four cameras located at top, left, right and front. These cameras have an established visual system to be able to identify boxes and return their center of mass coordinates in the image space. These cameras use a uniform identification system to assign an ID to these boxes so that they can be easily referenced.

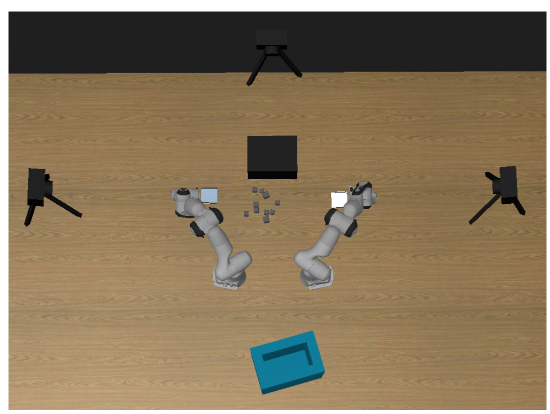

Fig. 1. Visual System

The process of converting from the image to global coordinates happens through the following two-step method.

- Convert from image space to view space.
- Convert from view space to world space.

One difference between our approach and that of the baseline code is that we have preferred to use the front

*This work was supported by SU Lab, UCSD CSE

[1]Shakeel Ahamed Mansoor Shaikna is with Department of Mechanical and Aerospace Engineering Engineering, University of California, San Diego, San Diego, CA 92092, USA samansoo@ucsd.edu

[2]Venkatesh Prasad Venkataramanan is with the Department of Mechanical and Aerospace Engineering Engineering, University of California, San Diego, San Diego, CA 92092, USA vvenkata@ucsd.edu

camera instead of the top camera for one major benefit. This will be discussed in much more detail in the subsection on Next Scene Strategy. View Fig. 2

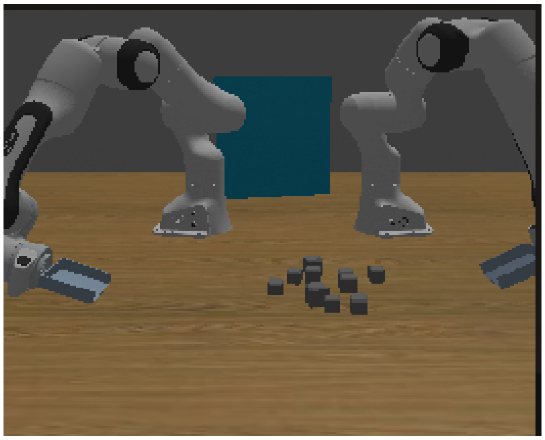

Fig. 2. Front camera

2) Strategy to pick up objects: Once the camera identifies the boxes and we compute their global coordinates, we randomise their IDs to get a box to grab. The reason we randomise is because we don't want the robot to get stuck in an endless loop trying to grab a box which it is not able to. By randomising their IDs, we are able to avoid this and try picking up other boxes.

Once the box has been chosen and its global coordinate computed, we will proceed to explaining the various phases in the robot's maneuver. The robot's maneuver is broken down into four phases. Here, picking up happens in Phase 1,2 and 3.

Phase 1 : The robot moves to a standoff point, which the authors choose to refer to as the swoop-in point. The two robots can be considered to be part of a circle with the box to be grabbed considered to be the centre of the circle. The two robots are diametrically opposite. A picture of this position can be found here. Fig. 3

Phase 2 : The robots swoop in for the pick-up maneuver. The robots are commanded to follow slightly different trajectories. One of the robots has to be positioned above the other, in order to scoop the boxes into the other box. This is required in order to execute a perfect forklift mechanism. A picture of this position can be found here. Fig. 4

Phase 3 : Once this has been completed, the robot which was positioned above moves slightly higher and then tips over any remaining boxes into the one positioned lower. A picture of this position can be found here. Fig. 5

3) Planning a trajectory: Now that the cubes have been picked up, we move on to Phase 4. Since Phase 4 is such an important maneuver, it has been broken down into two parts. First, the robots need to arrange themselves in a suitable manner in order to be create enough space for the robot to move towards the bin. So, the robot positioned higher(the left robot) is made to sway away from the centre, towards the left. The right robot retracts itself into a much more closed position, ready to rotate towards the bin.

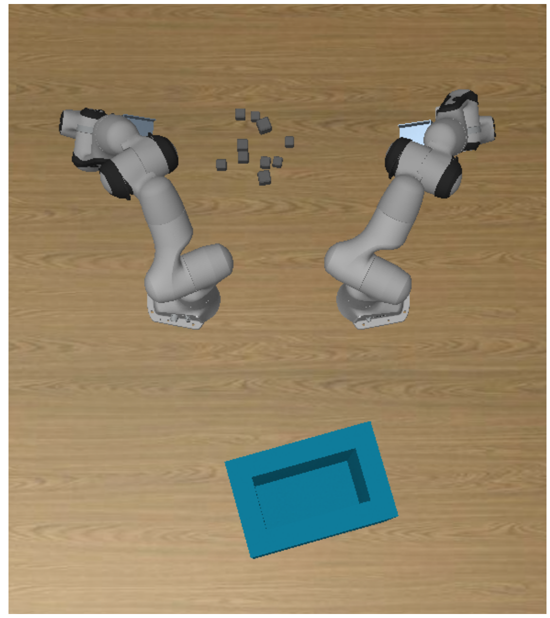

Fig. 3. Swoop in Point

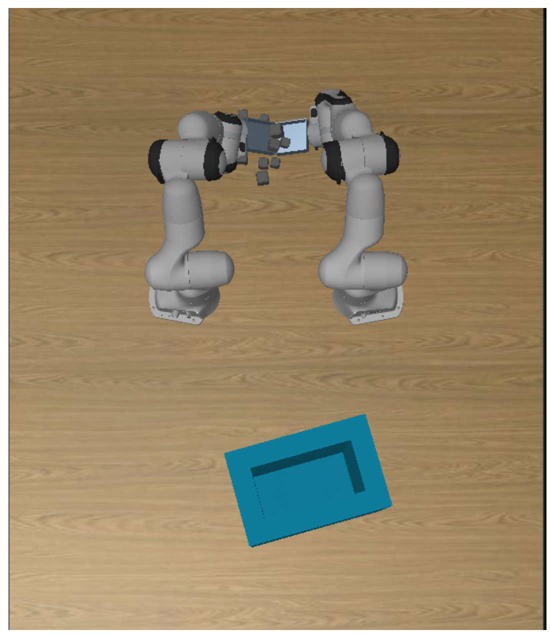

Fig. 4. Picking up Position

Secondly, the retracted robot rotates in position and stops in front of the bin. The other robot maintains its position. A picture of this position can be found here. Fig. 6

4) Strategy to place objects: Now that the robot is in front of the bin, it detracts and stretches in front of the bin. It is commanded to position the end-effector(spade) at the center of the bin. Once the space is positioned straight above, it is rotated in position and then the cubes are dropped inside. A picture of this position can be found here. Fig. 7 and Fig. 8

5) Next scene strategy: We choose when to move to the next scene based on when our front camera is no longer able to see the cubes. This is one of the major differences

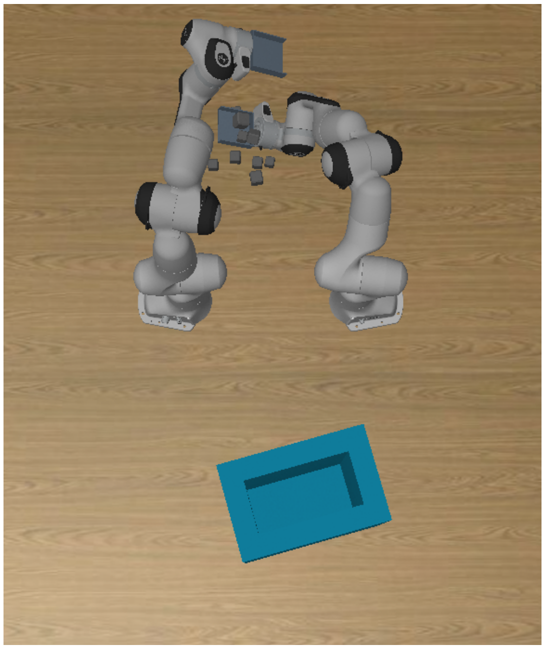

Fig. 5. Tip Position

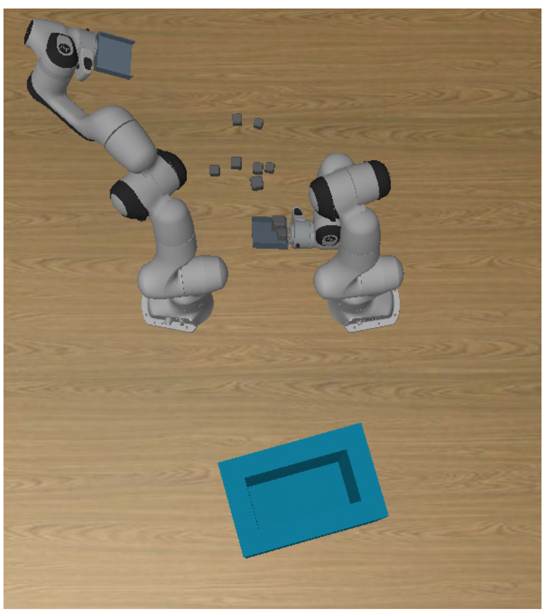

Fig. 6. Retracting Position

between the baseline code and our approach. Since the top camera is able to see the cubes even when the cubes are inside the bin, the robot keeps on trying to grab those cubes even if it isn't possible. Hence, if we use the front camera, the camera doesn't sight the bins placed inside and so the robot stops trying to grab those in the bin. So as soon as we stop getting those coordinates, we can stop trying to grab any more boxes and move on to the next scene.

B. Controller Design

The above mentioned robot maneuver needs precise control. This was achieved by developing a PID-based controller

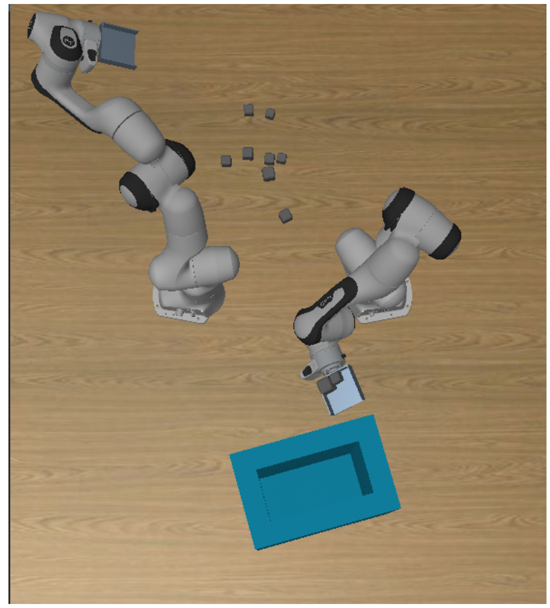

Fig. 7. Raising to drop

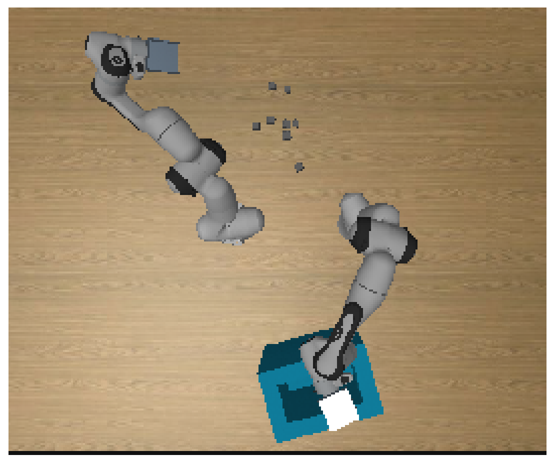

Fig. 8. Dropping position

borrowed from HW-2. The parameter tuning was done one joint at a time. At each step, the error plot was checked and then values were tuned. The controller outputs twist, which is then used to calculate the force that needs to act on the robot. The total force is calculated as the sum of the force that was calculated and the gravitational and coriolis forces, which are calculated using Sapien API which enables us to compute passive forces necessary to perform a correct simulation. The progress of the error plot for one of the joint is displayed below in Fig. 9

IV. RESULTS

The performance of three methods was tested to identify the best possible deterministic control method for this scenario. These were compared on the benchmark server.

The three methods tested were

- Inverse Kinematics based approach.

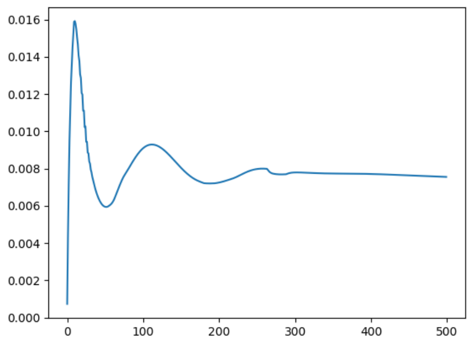

Fig. 9. Error Plot

Method	Performance	Efficiency
Inverse Kinematics	18.00%	0.53/min
Jacobian and Internal Controller	84.16%	3.02/min
Jacobian and PID Controller	89.09%	3.29/min

TABLE I
COMPARISON OF RESULTS

- Jacobian and Internal Controller based approach
- Jacobian and PID Controller based approach

Their results are tabulated in Table I. As you can see, the Jacobian and PID controller based method performs way better than other methods.

V. DEVELOPMENT PROCESS

We followed the following steps for development of the project.

- We first ran the baseline code, which gave very poor results. It gave us a success rate of 18%. So then we went about modifying the baseline code, by making appropriate changes to the parameters and the trajectory to get a better result. This actually gave us a really decent performance, with about 40% success rate.
- Then from the Piazza post that we put up, we realised that we could use Jacobian based twist calculation, like we did in HW-2. Hence, we tried to implement that, using the inbuilt internal controller. This gave us really good performance, with our success rate shooting up to 84%. This performance was obtained by optimising certain parameters like bin offset, using front camera instead of top camera for deciding when to move to the next scene, etc.
- Then we tried to use our own developed controller from HW-2, which uses PID control. During HW-2, the parameters were tuned by tuning each joint separately and moving on to the next joint when the error for that joint goes to zero. The controller was ported over to this project and certain parameters had to be tweaked to get it to work. The pick up strategy continued to fail even after a lot of tuning. Hence we developed a hybrid controller which uses a combination of internal and the self-developed PID controller. The PID controller performed really well for sequences where the robots had fixed joint targets, and hence it was employed there, whereas for all other sequences, we continued to use the internal controller. This hybrid controller gave a further boost in the success rate, taking it to 89.09%.

VI. LIMITATIONS AND FUTURE WORK

Some of the limitations are

- In certain cases, when the robots were unable to pick up the cubes, they did not stop trying. For example if the cubes were very small, and the robots were unable to pick them up, they ended up choosing the same box ID every time and failed repeatedly.
- There are some cases when the robots get stuck trying to move back to the swoop-in position. In those cases, we coded it to move on to the next scene.

Future work might include working on using Machine Learning techniques to improve the performance. There is a huge body of literature proving that machine learning can help improve the performance in such repetitive tasks, and so that is something that could definitely be tried on in the future.

ACKNOWLEDGMENT

We would like to thank Prof Su Hao for giving us a great learning opportunity by giving us this project. We would also like to acknowlegde the excellent effort by all the Teaching Assistants of this course, namely Yuzhe Qin, Jiayuan Gu, Fanbo Xiang and everybody who was extremely responsive on Piazza whenever we had doubts. We really learned a lot as a result of you guys consistently clearing our doubts. It really was a great learning experience and we would like to thank all who were involved in the course.