A Cartesian Platform for Cooperative Multi-Robot Manipulation Tasks

Silvio Müller, Stefan Ilić, Vincenzo Scamarcio and Josie Hughes

***Abstract*— For many manipulation tasks in environments such as laboratory or a kitchen, the presence of two robot arms is important to enable collaborative tasks requiring two arms (e.g. lid removal or tool use) or to improve the efficiency of scheduling of tasks. Currently, the development of multi-arm manipulation solutions has largely focused on 6 degrees of freedom articulated robot arms. However, cartesian robots have many advantages, including their precision, reliability, efficiency, and simple path planning. By developing a cartesian platform such that the end effectors of two mirrored systems can interact freely without collisions in 5 degrees of freedom, we can leverage the advantages of cartesian robots (high precision, simple planning, and low-cost hardware) and show robot cooperation. We equip each robot with end-effectors with different skills to increase the range of tasks the robots can cooperatively complete. To exploit this robotic hardware, we have developed a task-allocation and path-planning algorithm that enables these two mirror robots to work together to solve tasks collaboratively, exploiting the different skills and workspace of the two robots. We show how this robot can be used for cooperative tasks in lab automation, including pick and place, unscrewing vial caps, liquid pouring, and weighing. These demonstrate the feasibility and capabilities of the proposed robotic system for cooperative automation using cartesian robots.**

Humans can perform a wide variety of manipulation tasks with ease [1]. We can perform tasks such as pick and place or in-hand manipulation with a single hand, but we can also coordinate our two hands to perform more complex tasks such as unscrewing the lid of a bottle [2]. Interestingly, we also have a dominant hand and may transfer objects between hands to utilize a hand that shows greater capabilities [3]. An example of this would be passing a pen from one hand to another to write. Developing robots that can perform multi-hand manipulation in a similar manner to humans is a growing research area. This largely focuses on utilizing multi-degree of freedom (DoF) articulated robot arms, which offer a general purpose approach to manipulation [4], [5]. However, despite their efficiency, these robots are mechanically complicated and require the calculation of inverse kinematics, which contributes to their financial and computational cost [6]. To make advances in multi-robot manipulation, we require a simpler, more elegant approach to the design and implementation of collaborative manipulators. This would enable deployment in applications ranging from lab automation to manufacturing [7].

Cartesian robots have shown impressive capabilities in terms of precision, robustness, and simplicity for tasks in the domain of laboratory automation [8], [9] or agriculture [10], [11]. However, the design of cartesian robots still limits their ability to collaborate and perform multi-robot tasks, as their axes are typically aligned such that this is impossible. Backdrivable cartesian systems have been created that enable human-robot collaboration [12]. There have also been some limited examples of the design of cartesian robots for robot-robot cooperation. This includes a system that allows cooperation between an end-effector in one axis and a single direction [13], providing very limited opportunities for cooperation. A secondary challenge that arises is the need for planning algorithms to identify the nature of cooperation and the task split between different robots [14]. There has been a creation of a taxonomy to describe different forms of robot cooperation [15]. For the case of two robots collaborating on tasks that require precedence, there is ongoing work for task planning and optimization [16].

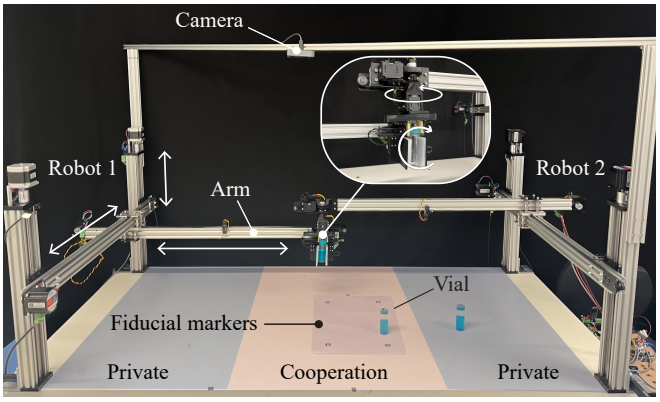

Fig. 1. Developed cartesian robots cooperate to unscrew a vial's lid. The projection of the workspaces is indicated with colors on the workbench. The overlapping workspace of the robots is the cooperation workspace. The remaining workspaces are private to each robot.

We propose a novel design of a two-robot cartesian manipulation platform that allows full cooperation between end-effectors whilst maintaining the precision and simplicity in design offered by cartesian systems. The developed platform shown in Fig.1 consists of two mirrored cartesian robots. The free-standing arms enable cooperative manipulation within a 3D space but can also be retracted to avoid collisions. To utilize cooperation between these systems, we propose configuring them such that each has a private area, with no obstruction from other robots, and also a collaborative area where the two arms can both operate. We also take inspiration from humans 'handed-ness' and provide the end-effectors with different skills and capabilities to simplify the robot end-effector design whilst maximizing the system's capabilities. We propose an algorithm for splitting tasks between the two robots based on different modes of cooperation that exploit the different skills and operational regions of the robots. We demonstrate the capabilities of the robot on example tasks inspired by lab automation scenarios.

In the remainder of the paper, we present the design of the robot and methods used for cooperation and task

planning. Following this, we provide a description of the robot's physical implementation and present experimental results. Finally, we conclude with a summary of our findings and suggestions for future research.

I. Methods

A. Problem Definition

The robot and task planning methods have been designed for performing manipulation tasks over a desk-sized area, similar in size to typical laboratory benches. We focus on a range of items and objects that may be found in lab environments and performing typical lab automation tasks. This includes pick and place of objects, opening bottle lids, pouring fluids, weighing objects, and also re-orientation of objects. These tasks include some that require only one end effector and some that require coopration and co-ordination between the two.

B. Collaborative Cartesian Robot Design

a) Carteisan Robot Design: Two mirrored cartesian robots form the collaborative manipulation system. Each Z-axis is formed from a lead screw mechanism actuated from each end, which moves the X-Y axes. Using a lead screw for this axis prevents back-drivability, ensuring it is safe when powered down and the load-carrying capacity is high. The X-Y axes are belt-driven, providing speed and also minimizing their mass. The robots have a serial configuration with the Z-axis at the start of the chain and the Y-axis at the end. The entire Y-axis is moved when actuated. This means that it can fully retract, allowing the two robots to interact in 3D space freely and around 5 degrees of freedom.

The mirrored design of the robots causes their two Y-axes to extend towards each other. They have a reach of y_r in the y direction. They are separated by a distance y_s, which satisfies $y_s > y_r$ such that each robot has a 'private' zone in which only it can operate and also a 'collaborative zone' in which both robots can operate. This has been implemented such that in the private workspace, there is no need for collision avoidance, and one-arm tasks (e.g. pick and place) can be performed with ease, yet, there is sufficient area for mutual collaborative tasks in the shared workspace.

b) End-Effector Capabilities: To maximize the 'skills' of the robot whilst minimizing the complexity of the design of the end-effectors, each end-effector has different capabilities. By sharing rather than duplicating capabilities, we extend the robot's capabilities. To fully exploit these skills, it does become necessary to exchange objects between effectors which increases the complexity of the task planner.

Both end effectors are equipped with parallel grippers designed to grasp objects. Gripper 1, located on robot 1, grips objects from the side and can rotate around the X-axis up to one full rotation. It also has a load cell to enable weighing and force detection. Gripper 2 has the parallel gripper mounted such that it grips vertically and can continuously rotate around the Z-axis. Distributing the skills between the two grippers simplifies the design. The

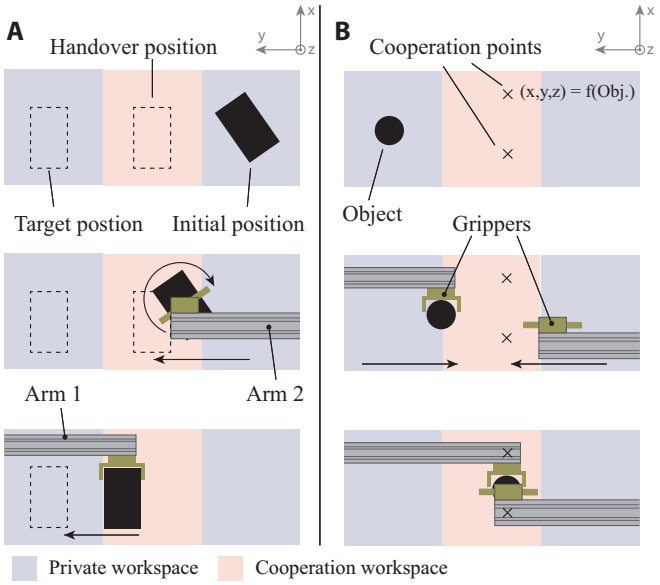

Fig. 2. Methodology for cooperation. (A) Visualizing the handover of objects between robots. (B) Illustration of the dual-arm cooperation.

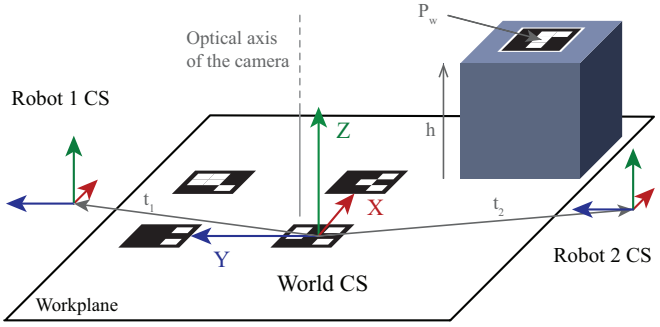

Fig. 3. Definition of the world coordinate systems (CS) with fiducial markers. The fiducial marker on top of the object is used for localization, as seen in the back right position.

two robots must then collaborate to keep the same skillset as one complex gripper.

c) Cooperation Methods: The robot has two arms, each with a different reach and different skills or capabilities of the end-effector. To utilize these different skills and perform collaborative actions, we must identify the algorithm to distribute the task and work collaboratively. We can consider two modes of collaborative task-solving.

- **Mode 1.** If an object requiring specific robotic skills is located in another robot's private workspace, the object must be handed over to the other robot capable of performing the task (Fig. 2A). The second arm picks up the object in his private area and places it at a predefined position in the cooperation area. The first arm, which has the required skill set, can then execute the task.
- **Mode 2.** To simultaneously manipulate an object, both arms meet at predefined points inside the cooperation workspace. This protocol for two-handed cooperation is shown in Fig. 2B. The cooperation points are specific to each object, as the object's shape dictates the needed gripper positions. The robots enter the cooperation zone only while following these two protocols to avoid collisions.

C. Computer Vision & Coordinate Systems

To perform a task, we must first identify the location of objects in the workspace. Specifically, we require the 2D position on the workplane of an object and its height h. To enable identification of the different objects, all have a fiducial marker on the top which also identifies the object identity and location.

The global coordinate system is defined with four fiducial markers placed in a rectangle on the workplane, as depicted in Fig. 3. The translation vectors t_1 and t_2 from the world coordinate system to the robot coordinate system are manually measured. To map from world coordinates to image coordinates, the camera is modeled as a pinhole camera, $p = \mathbf{P}P_w$, where P is the camera matrix, p is a 2D point on the image plane, and P_w is a 3D point defined in world coordinates. All points are given in homogeneous coordinates. The extrinsic parameters are estimated using the four markers of the world coordinate system in a Perspective-n-Point algorithm (Levenberg-Marquardth Method [17]).

To obtain a mapping from image coordinates to world coordinates, the pinhole camera model must be solved for P_w. However, the camera matrix $\mathbf{P}$ is non-invertible. This is solved by introducing a coordinate plane on which point P_w lies. Therefore, the modified pinhole camera model can be given as $p = \mathbf{PM}P_p$. Where $\mathbf{M}$ maps from the coordinate plane to the world coordinate $P_w = \mathbf{M}P_p$. The matrix $\mathbf{M}$ has the following form:

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & h \\ 0 & 0 & 1 \end{pmatrix}$$

We can now solve for the world coordinates $P_w = \mathbf{M}(\mathbf{PM})^{-1}p$, and this can be applied to the fiducial marker's center pixel to get the corresponding coordinates of the object in the world coordinate system.

D. Task planning & Allocation

We have a multi-robot task allocation problem (MRTA) for heterogonous robots; each robot can execute at most one task at a time, but the robots must collaborate on some tasks. The tasks we consider also require synchronization and also have precedence. Thus, we have a single task, multiple robots, time-extended with synchronization and precedence problem, as described in the taxonomy of MRTA problems [15]. To solve the task allocation and cooperation problem, we adapt an algorithm proposed in [16].

We describe all possible motions that can be applied to an object in a Mode Graph (Fig. 4A), specifying the possible order of precedence in which they can be applied. All possible motion primitives the robots can perform are described by $M = \{m_1, ..., m_i\}$, which includes all motions described in the Mode Graph and includes pick, place, rotate, unscrew, weigh and pour.

The tasks are to be completed by a set of n robots, $R = \{r_1, \cdots, r_n\}$, in our case $n = 2$. Each robot r_n has a set of possible skills $S_n = \{s_1, \cdots, s_m\}$, which are a sub-set

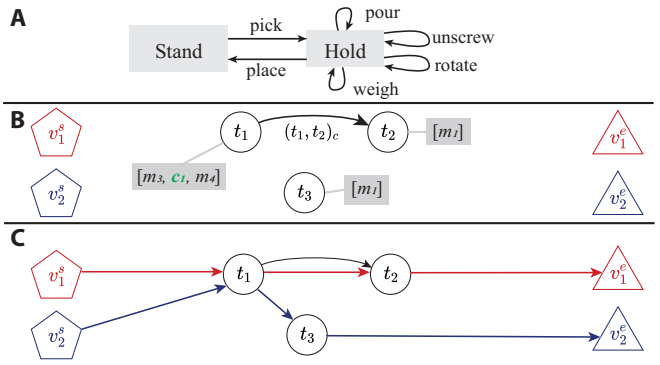

Fig. 4. Illustration of Task planning & allocation: (A) Mode Graph for a liquid container with a cap. (B) Problem definition for the example task. The start vertices are depicted with pentagons, and the end vertices with triangles. The directed edge between t_1 and t_2 indicates a precedence constraint. The motion primitives for each task are shown in brackets, including the inserted cooperation primitive c_1 (C) Optimized mission plan for the sample tasks. The tasks assigned to a robot are depicted with a path connecting the start and end vertex of that robot.

of possible motion primitives such as $S_n \subseteq M$. The robots also have a region of operation, O_n, which is a subset of $O \subset \{p1, p2, c\}$ (i.e. private 1, private 2, cooperation).

The specific problem can be defined as a directed graph $G_C = (V, E_C)$. The set of vertices $V = \{v_l^s, T, v_l^e\}$, $\forall l \in \{1, \cdots, n\}$, contains the tasks T, as well a starting vertex v_l^s and an end vertex v_l^e for each robot. A task $T = \{t_1, ..., t_p\}$ is defined by a set of t_p sub-tasks, where each sub-task only involves one object. If sub-task t_i must be finished before sub-task t_j, the graph G_C includes a directed edge $(t_i, t_j)_C$. Each sub-task (t_p) is defined by a sequence of motion primitives such that $t_p = \{m_1, m_2, ...m_q\}$. Each motion primitive m_i has a start and end configuration $m_{i,locs}$ and $m_{i,loce}$ respectively, and can be described as operating in a given operation region $m_{i,O}$. This reflects the location and orientation. The problem definition for an example task where sub-task t_1 has to be executed before sub-tasks t_2 is shown in Fig. 4B.

The task allocation must determine which robot or robots should perform each sub-task in G_C. The possible robot alliances which can be used to solve the task T are given by $A = \{a_1, \cdots, a_k\}$, with $a_k \subseteq R$. Our robotic setup has three possible alliances, two where each of the robots are alone, and one where they are working cooperatively $A = \{\{r_1\}, \{r_2\}, \{r_1, r_2\}\}$.

To identify the optimal alliance a_j to complete the given sub-task, we must consider the location of the object and also the possible skills of each robot. Starting from the precedence graph, G_C, the semicomplete directed graph called a mission plan $M = (V, E)$ has to be determined. Through the edges, E, an alliance is assigned to a sub-task. The appropriate alliance is found by evaluating a cost metric associated with sub-task alliance pairs and the edges E.

Alliance a_j can execute a sub-task if the alliance has the required skills for each motion primitives of the sub-task and the start and end location of the motion primitives is in the reachable space. If an alliance can execute a sub-task, the

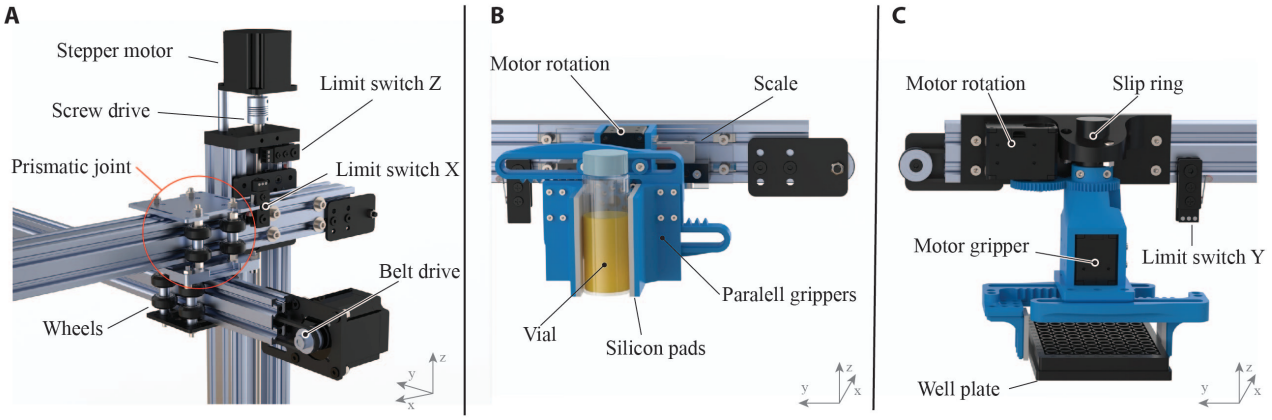

Fig. 5. Illustration of the platforms mechanical design: (A) Configuration of the linear stages and their connections. (B) Horizontal gripper with continuous X-axis rotation and built-in scale. (C) Vertical gripper with continuous rotation around the Z-axis.

sub-task alliance pair has an associate cost of zero:

$$c(t_i, a_j) = \begin{cases} \infty & \text{if } S(a_j) \text{ can not execute } t_i, \\ 0 & \text{otherwise} \end{cases}$$

To determine the optimal alliance, a second cost metric is used. This dynamic cost depends on the mission plan's incoming edges $E_{in}(v)$ of a specific vertex v. In this case, the cost function (c_{dyn}) is the euclidean distance the robot moves between the end position of its previous tasks and the start position of the next task.

For any sub-task which involves the alliance of two robots, it is then necessary to identify the cooperation method required (see Section II.Bc). This requires analysis of the motion primitives m_i for each sub-task t_i. When there is a change of operational areas (O) between motion primitives, such that $m_{i,O} \neq m_{i+1,O}$ this indicates the need for cooperation of mode 1, and the motion primitive for this c_1 in inserted between m_i and m_{i+1}. When the skillset S_n of more than one robot in the alliance is needed, there is a need for cooperation of mode 2.

The resultant graph, $M = (V, E)$ (Fig. 4C), now represents the optimized mission, distribution of tasks, and series of motion primitives. From this, the robot's path planning can be generated using pre-programmed motion primitives, which are parameterized by the object's location and height. For each sub-task t_i, a robot executes its tasks until it reaches mode 1, mode 2 motion primitive, or a task where the precedence is not fulfilled, after which the other robots start executing. This prevents robot-robot collisions as only one robot is simultaneously in the cooperation zone.

II. Robot Implementation

A. Mechanical design

The platform is formed from two mirrored cartesian robots, each with a workspace of 1.5m x 0.93m. The robots are constructed from V-slot aluminum profiles allowing the design of a lightweight, rigid, and easy-to-assemble structure, Fig. 5A. The robot uses two linear screw actuators in parallel for the motion on the Z-axis. Using two actuators in parallel reduces the moment about Y-axis and provides high load capacity. The Z-axis is not back-drivable and thus allows the

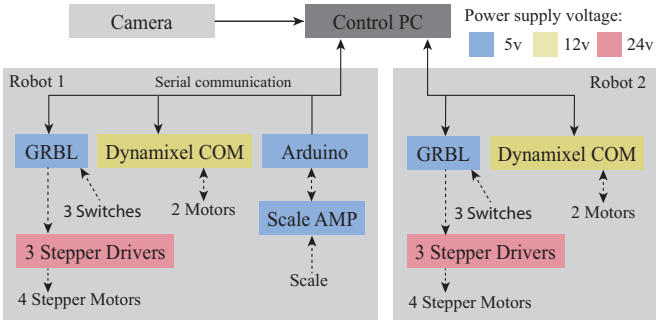

Fig. 6. Hardware and control architecture utilized in the robot. The colors indicate the needed supply voltages for the different systems.

robot to maintain its Z position during power loss. A lighter, faster, and non-locking belt drive is used for the X and Y axes. These are connected to different axes via prismatic joints designed to minimize mechanical play and moments. The linear actuators in all axes are driven by stepper motors allowing positioning with 0.1mm accuracy. During startup, each robot is calibrated using limit switches, and the Y-axis retracts from the cooperation zone to avoid any collision.

The parallel grippers developed for the two robots are shown in Fig. 5B and C. These are powered by DC motors with integrated position detection and speed control capabilities (Dynamixel XL430-W250-T). An identical rack and pinion gripping mechanism is used for both grippers and is driven by a central gear attached to a DC motor. Soft silicone pads have been applied on the fingers of the grippers to increase friction. Both grippers are made out of 3D-printed parts, which are assembled using screwed connections.

The horizontal gripper is designed to pick objects from the side, and thanks to the integrated load cell, it can measure the weight of the gripped object (Fig. 5B). It provides an additional degree of freedom in terms of continuous rotation around the X-axis. The vertical gripper is suited to pick objects from above and offers continuous rotation around Z-axis thanks to the integrated slip ring (Fig. 5C). Grippers enhance the motion capabilities of each robot by providing an additional rotational degree of freedom besides the three existing translational degrees of freedom. This means that each robot has a total of 4 DoF, while cartesian platform has 5 DoF, with only rotation around Y-axis being absent.

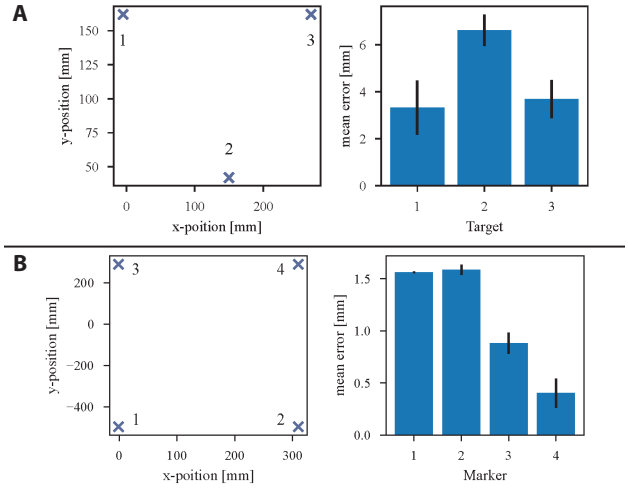

Fig. 7. Experimental results of positional accuracies: (A) Measured positioning error after transferring three well plates n=6. (B) The error of the vision system for four spread-out ArUco markers, n=40.

B. Control

The top-level platform is controlled by a Python script that runs the computer vision, the task allocation, and the path planning algorithms. The low-level robot motion control is performed using GRBL running on Arduino Uno, as shown in Fig. 6. Each robot has separate low-level control modules, while one stepper driver is used for each robot axis. Communication between the top and low-level controllers is performed via USB serial. The gripper positional control is performed via proprietary protocol with a smart actuator communication module (Dynamixel U2D2). At the same time, the load cell amplifier connected to Arduino Uno provides weighing data in plain text at the frequency of 10Hz. The vision system uses a 1080p webcam mounted 80cm above the workbench. As fiducial markers, 17.5mm ArUco markers which contain 4x4 bits with a minimal hamming distance of four, are used.

III. Experimental Results

The robotic platform's performance was evaluated by conducting experiments to assess its accuracy in pick and place operations and its full range of capabilities, including multi-arm cooperative manipulation.

A. Positional precision

The overall positional accuracy is estimated by repeatedly transferring a 96-well plate from one side of the platform to the other using the transfer protocol shown in Fig. 2B. For the experiment, three plates at different initial positions were transported six times, measuring only the final positional error. The location of these three different start points and the mean error is shown in Fig. 7A. The accuracy is in the order of 3.3-6.6mm. This shows high reliability and repeatability, resulting in a low accumulated error.

The accuracy of the computer vision system is assessed by repeatedly calibrating extrinsic camera parameters, locating four ArUco markers, and calculating the difference between the measured and actual position. To evaluate how well the vision system performs with the highest camera distortion, markers are positioned on the edges of a 780x310mm rectangle which covers most of the camera's field of view. Despite the high image distortion, the computer vision system shows high accuracy of 0.4-1.6mm, as illustrated in Fig. 7B. The conducted experiments show that camera vision is 15 times more accurate than the overall robot, suggesting that positioning errors come from the robot's actuation. Although multiple sources can impact variability, gripper compliance seems to be the most likely source of error.

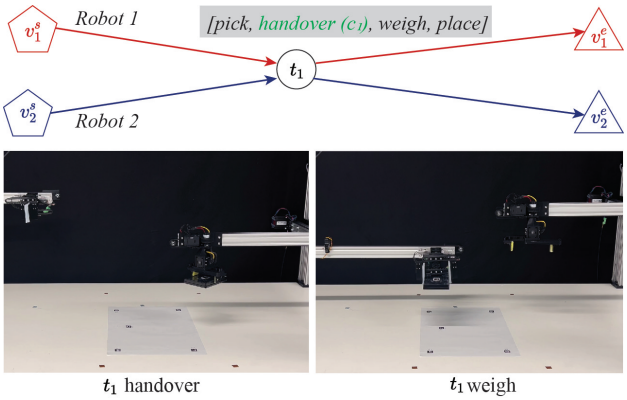

Fig. 8. Handover of a 96-well plate (Mode 1), with mission plan.

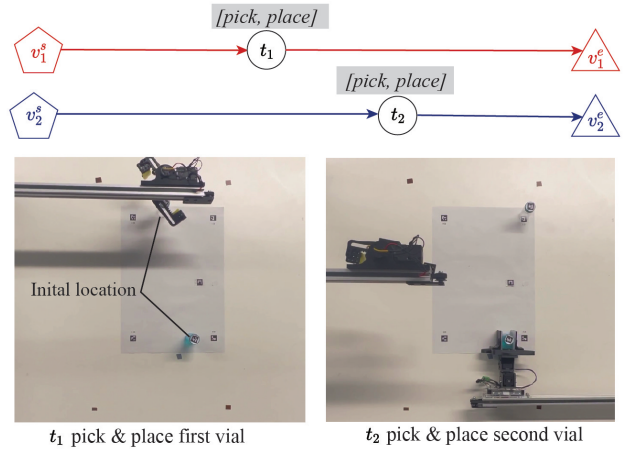

Fig. 9. Demonstration of the task allocation algorithm for pick and place.

B. Cooperative manipulation

To demonstrate the planning algorithm and the cooperation capabilities, we showcase several experiments highlighting the task allocation algorithm and the proposed cooperation protocols. In the first experiment, the task is to weigh an object not reachable by the end effector with the scale, requiring cooperation between two robot arms using handover protocol as shown in Fig. 8A. The alliance $\{r_1, r_2\}$ is assigned by the task allocation algorithm. It requires a robot arm to pick an object, place it at the handover position and allow the second robot arm to perform picking and weighing.

The second experiment illustrates how the objective function within the task allocation algorithm minimizes the robot motion in pick-and-place applications. It involves the relocation of two vials that are accessible by both robots to different positions in a cooperative space. As seen in Fig. 9, the robot closest to the vial is assigned for the pick and

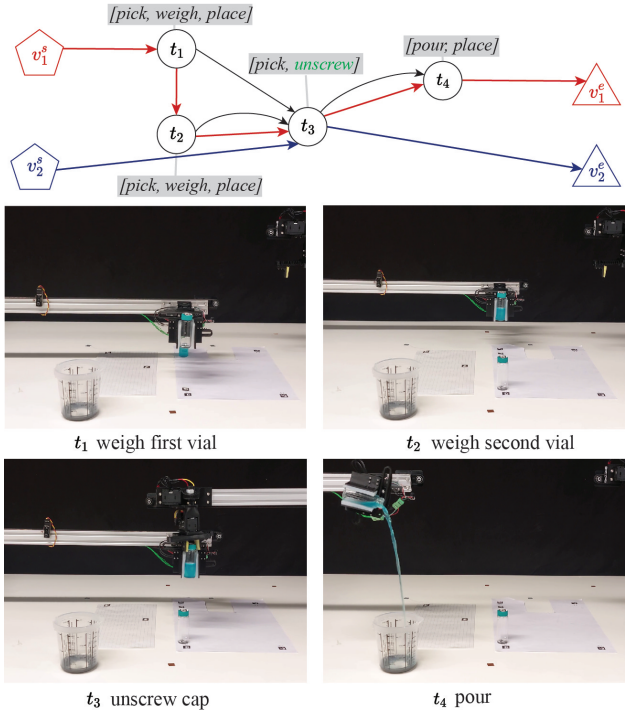

Fig. 10. Demonstration of the robot's capabilities with many precedence constraints: pick and place, weighing and unscrewing of a cap. The black lines shown in the mission plan on top indicate the precedence constraint.

place operation, minimizing travel distance.

In the third experiment, we demonstrate a complex series of tasks with many precedence constraints involving weighing of two vials and pouring the contents of the heavier one into a beaker, as seen in Fig. 10. To identify the heavier vial, both weights must be determined first. Pouring cannot be performed before vial cap removal. Dual-arm control is required for the cap removal since this cooperative task requires that the horizontal gripper holds the vial while the vertical gripper performs unscrewing.

IV. Conclusion

In this study, we propose a novel configuration of a two-robot cartesian platform that enables various cooperation modalities between the robots. Our approach takes advantage of the accuracy and precision of the cartesian robots and simple inverse kinematics to create a cooperative robotic system capable of performing pick and place and multi-arm tasks within a large workspace. We developed a task allocation algorithm to plan for cooperation between the robots, accounting for their different workspaces and skills.

Compared to general-purpose articulated robot arms with six degrees of freedom, our system offers a larger workspace, precision, and simpler control at a lower cost, estimated at 3000 USD. However, the platform's limited degrees of freedom impose some restrictions on the types of tasks and activities that can be performed. To increase the platform's versatility, the task planning could be extended to include the rearrangement of objects in the robot's workspace by implementing existing task and motion planning algorithms [18]. Moreover, improving the control software to enable the simultaneous operation of the two robots and exploring more sophisticated cost functions to reduce the total execution time are other potential research avenues.

Acknowledgment

This project was supported by the EU's Horizon 2020 research and innovation program under the Marie Skłodowska Curie grant agreement No 945363.

References

[1] R. R. Ma and A. M. Dollar, "On dexterity and dexterous manipulation," in *2011 15th International Conference on Advanced Robotics (ICAR)*. IEEE, 2011, pp. 1–7.

[2] I. M. Bullock and A. M. Dollar, "Classifying human manipulation behavior," in *2011 IEEE International Conference on Rehabilitation Robotics*. IEEE, 2011, pp. 1–6.

[3] M. Papadatou-Pastou, E. Ntolka, J. Schmitz, M. Martin, M. R. Munafò, S. Ocklenburg, and S. Paracchini, "Human handedness: A meta-analysis." *Psychological bulletin*, vol. 146, no. 6, p. 481, 2020.

[4] Z. Feng, G. Hu, Y. Sun, and J. Soon, "An overview of collaborative robotic manipulation in multi-robot systems," *Annual Reviews in Control*, vol. 49, pp. 113–127, 2020.

[5] F. Basile, F. Caccavale, P. Chiacchio, J. Coppola, and C. Curatella, "Task-oriented motion planning for multi-arm robotic systems," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 5, pp. 569–582, 2012.

[6] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, p. eaat8414, 2019.

[7] M. Pozzi, V. R. Garate, A. Ajoudani, K. Harada, M. Malvezzi, L. Rozo, and M. A. Roa, "Emerging paradigms for robotic manipulation: From the lab to the productive world [from the guest editors]," *IEEE Robotics & Automation Magazine*, vol. 28, no. 2, pp. 10–12, 2021.

[8] J. A. Bryant Jr, M. Kellinger, C. Longmire, R. Miller, and R. C. Wright, "Assemblytron: Flexible automation of dna assembly with opentrons ot-2 lab robots," *Synthetic Biology*, vol. 8, no. 1, p. ysac032, 2023.

[9] T. P. Mabate, K. Potgieter, P. P. Molokoane, R. Meijboom, and N. Bingwa, "Rapid kinetic evaluation of inorganic-perovskite-catalysed redox conversion of p-nitrophenol and morin aided by an opentrons robotic system," *Journal of Materials Science*, vol. 57, no. 25, pp. 11 590–11 611, 2022.

[10] A. Zahid, M. S. Mahmud, L. He, D. Choi, P. Heinemann, and J. Schupp, "Development of an integrated 3r end-effector with a cartesian manipulator for pruning apple trees," *Computers and Electronics in Agriculture*, vol. 179, p. 105837, 2020.

[11] C. J. C. Moscoso, E. M. F. Sorogastúa, and R. S. P. Gardini, "Efficient implementation of a cartesian farmbot robot for agricultural applications in the region la libertad-peru," in *2018 IEEE ANDESCON*. IEEE, 2018, pp. 1–6.

[12] R. Bloss, "Collaborative robots are rapidly providing major improvements in productivity, safety, programing ease, portability and cost while addressing many new applications," *Industrial Robot: An International Journal*, vol. 43, no. 5, pp. 463–468, 2016.

[13] J. Barnett, M. Duke, C. K. Au, and S. H. Lim, "Work distribution of multiple cartesian robot arms for kiwifruit harvesting," *Comput. Electron. Agric.*, vol. 169, no. 105202, p. 105202, Feb. 2020.

[14] J. K. Behrens, R. Lange, and M. Mansouri, "A constraint programming approach to simultaneous task allocation and motion scheduling for industrial dual-arm manipulation tasks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8705–8711.

[15] E. Nunes, M. Manner, H. Mitiche, and M. Gini, "A taxonomy for task allocation problems with temporal and ordering constraints," *Robotics and Autonomous Systems*, vol. 90, pp. 55–70, 2017.

[16] E. Bischoff, F. Meyer, J. Inga, and S. Hohmann, "Multi-robot task allocation and scheduling considering cooperative tasks and precedence constraints," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 3949–3956.

[17] K. Madsen, H. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems (2nd ed.)," p. 60, 01 2004.

[18] J. Chen, J. Li, Y. Huang, C. Garrett, D. Sun, C. Fan, A. Hofmann, C. Mueller, S. Koenig, and B. C. Williams, "Cooperative task and motion planning for multi-arm assembly systems," *arXiv preprint arXiv:2203.02475*, 2022.