

## Article

# An Omnidirectional Platform for Education and Research in Cooperative Robotics

Majd Kassawat \*, Enric Cervera  and Angel P. del Pobil 

Department of Engineering and Computer Science, Universidad Jaume I, 12071 Castellon de la Plana, Spain; ecervera@uji.es (E.C.); pobil@uji.es (A.P.d.P.)

\* Correspondence: majd.kassawat@gmail.com

**Abstract:** In this paper we present a new, affordable, omnidirectional robot platform which is suitable for research and education in cooperative robotics. We design and implement the platform for the purpose of multi-agent object manipulation and transportation. The design consists of three omnidirectional wheels with two additional traction wheels, making multirobot object manipulation possible. It is validated by performing simple experiments using a setup with one robot and one target object. The execution flow of a simple task (Approach–Press–Lift–Hold–Set) is studied. In addition, we experiment to find the limits of the applied pressure and object orientation under certain conditions. The experiments demonstrate the significance of our inexpensive platform for research and education by proving its feasibility of use in topics such as collaborative robotics, physical interaction, and mobile manipulation.

**Keywords:** multirobot transportation; mobile robot; omnidirectional



**Citation:** Kassawat, M.; Cervera, E.; del Pobil, A.P. An Omnidirectional Platform for Education and Research in Cooperative Robotics. *Electronics* **2022**, *11*, 499. <https://doi.org/10.3390/electronics11030499>

Academic Editor: Jeha Ryu

Received: 22 December 2021

Accepted: 4 February 2022

Published: 8 February 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Designing multi-robot systems has captured the attention of many researchers/industries due to its scalability and various applications. As discussed in [1], the majority of the methods used can be categorized into three main strategies: Pushing-only, grasping and caging strategy. Caging strategy was followed in [2] where, using a composition of three behaviors (approach, surround and transport), a group of eight robots were able to displace an L-shaped object to a target location. A pure pushing strategy was discussed in [3] where the swarm takes advantage of the object occluding the visibility of the goal position to decide the direction of pushing. Unlike [4,5], mentioned examples do not realize a solid connection with the object as a preparation step for transporting. The authors in [4] explore a decentralized sliding mode control strategy to move a load along a straight line at a desired velocity. The object is pre-grasped by all robots and the paths of the robots and object are monitored. The authors demonstrate that this strategy does not depend on inter-robot communication, team size, or load related information. Experiments using different object sizes and shapes were performed in [5]. The possibility of having chains of robots or mixed typologies linked to the object was discussed and experimented using an evolutionary algorithm applied to up to 16 robots. These experiments showed that using this method, the swarm can transport heavier objects than using simple one layer caging. While most of the mentioned work concentrates on moving the object from point A to point B, the object is assumed to be pre-grasped or it is pushed on the flat surface of the robots. A mechanical manipulator has been designed to be mounted on mobile robots to be used to collaboratively lift an object to the base of the robots [6,7]. This mechanism relies on friction induced between the set of manipulators and the surface of the object. The solution uses the flexibility of a parallelogram shape with 2 DOF. The structure passively raises once pressure is applied to the end effector. Having two robots applying pressure on opposite sides elevates the object off the ground to be later transported. In [8,9], the problem of

object transportation is moved to aerial robots, which have more degrees of freedom and have more possibilities to manipulate the object.

In this article, we will describe a design of a collaborative manipulator which senses the forces induced in the target object by other robots. This technique has been used in leader–follower strategy in [4,10], where force readings are used to analyze the direction of the leader’s movements considering that the leader could be a human, a human operated robot, or an autonomous robot. Our concept builds on the approach taken in [6,7], where lift is achieved by friction between the robots and the object, rather than direct grasping as in [4]. Indeed, our design adds the ability to tilt the object while in contact and provides free planar motion to the mobile platform to be used for cooperative transportation compared to a two-motor driven system which is not capable of doing omnidirectional motion. The benefits of omnidirectional motion were shown in [11,12], where a group of two or three omnidirectional robots transport an object to a target destination while avoiding near obstacles. This approach uses an expensive platform equipped with a robotic arm [13] compared to our setup, which simplifies the main components of the agents in order to perform simple, yet effective, lift-transport operation. In [14,15] the authors present interesting low-cost designs for multi-robot manipulators. While those robots are a good fit for research and education, they have drawbacks, such as a two-wheel drive system and having to modify the target object for grasping.

We present a new concept for cooperatively lifting, manipulating, and transporting an object. The concept has already been validated in simulation in [16]. In this paper, we implement an improved version using real hardware and we validate the design in a one-robot scenario with the aim of illustrating the use of the robot platform for research and education in cooperative robotics and manipulation. We believe that it presents a more simplified and affordable approach to multi-robot load sharing and transportation compared to other state-of-the-art approaches. Indeed, it can be applied to lift and manipulate objects that are not prepared for a the usual transporting mechanism—like a forklift—or large objects that could not otherwise be transportable using one agent. Moreover, as it depends on traction, it does not require a special grasping area for each agent or modifying the target object for transportation. Furthermore, having a modular system integrated using ROS and using off-the-shelf components positions it perfectly for use in research and education, since this design allows the addition of new features and modifications without affecting its base platform.

The paper is organized as follows. Section 2 describes the implemented system, Section 3 presents the concept and some additional considerations, and Section 4 reports three experiments with their results in order to validate the system. The paper concludes with a discussion and conclusions, including future work.

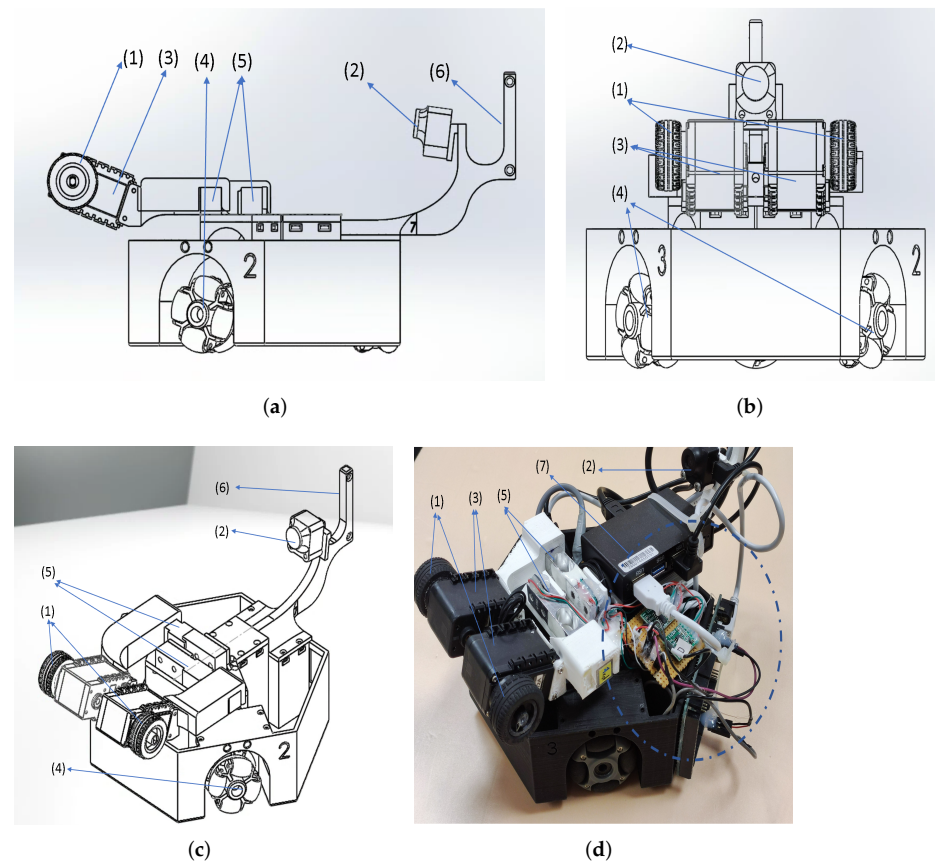
## 2. Description of Implemented System

### 2.1. Hardware Platform

The design is based on an omnidirectional mobile platform consisting of: three omnidirectional wheels attached to servo motors DYNAMIXEL AX-12A (4), one uEye XS camera (2), two 1KG load cells (5), traction wheels (1) mounted on two servo motors (3), a USB hub (7), and a cable guide (6). Figure 1. Robot chassis and custom links are 3D printed.

The three wheels and motors enable the robot to perform full omnidirectional movements, which simplifies the planning problem for one robot [17–19] and makes a multi-robot object manipulation possible. Load cells (or force sensors) will measure the force exerted from the robot to the object (expected to be less than 1KG for our experiments) [20]. Note that having a higher max value would result in lowering the resolution of measurements. The data collected is passed through a micro-controller to the processing node. Traction motors work as a differential drive to control height and orientation of the target object. The camera is used to know the position and orientation of the target object using a fixed marker. In this model, all sensors and controllers are directly wired to a PC, ensuring cables

do not influence the motion of the robot at any time. Power (5v,12v) is provided by an external power supply. This setup allows easy remote access to the experiment.



**Figure 1.** Robot Design: (a) side; (b) front; (c) corner; and (d) real.

## 2.2. Controller Software

### 2.2.1. Overview

We use ROS nodes as an interface between the low level drivers for the motors and sensors, as well as for implementing the control logic for the robot.

The trajectory file contains a list of trajectory points. Each point is described by a number of constraints (specific required inputs); these inputs could be the position of the robot relative to the marker, force values read using the force sensors, certain torque limit, or simple time conditions. Each condition has its own target, error tolerance, bias, and gain, which should be used in the corresponding error correcting methods in the controller. As usual,  $K$  represents the gain in the low-level controller and the bias is a constant value that is summed to the output to the actuators. One point is reached when all the conditions included in that point are reached. Separate points are processed sequentially until the end of the trajectory. Figure 2 shows trajectory points and robot control stages. As mentioned, it is possible to have more than one condition in one point and, our current implementation supports the use of 4 types of control, Figure 3, that we can use to reach said conditions:

- Single: this indicates one correction is going to be processed. This is usually the leaf type for single access corrections;
- Sequential: this indicates that child conditions will be processed sequentially. Once the correction is done, the controller will not check this condition again;
- Parallel: child conditions will be processed at the same time so in each tick all the child corrections are processed and the result output is calculated from the outputs of all child corrections;

- Semi-sequential: in one tick, the controller goes over child corrections sequentially, calculates the first unfinished correction, ignores the rest, and outputs the result. This means that each time, the controller will not proceed to next condition until it has fixed the first ones. The difference with sequential is that this method checks the finished child corrections each tick for any new deviation from target.

It is possible to nest these types as needed. This feature brings certain flexibility to the user to form the trajectory which fits the application the most.

In each tick, the controller checks for all the inputs/references required to process the current point. This check is performed to ensure no null data is introduced to the controller while in action. The frequency used is 40 hz, which is enough, taking into account that camera frame rate is about 20 frame/s.

### 2.2.2. Available Types of Conditions

- Planar conditions: the conditions are to meet certain position(X,Y)/orientation(Alpha) relative to a reference frame (the marker in our case);
- Boolean: target value is True/False;
- Time (wait): target is reached after certain time;
- Force (single access): target is reached when force sensor reading reach set target.
  1. Total pressure collected from both sensors is within allowed range. This is corrected by moving the robot forward;
  2. Difference between force sensors readings is below a certain tolerance, which is corrected by rotating the robot while applying pressure on target object.

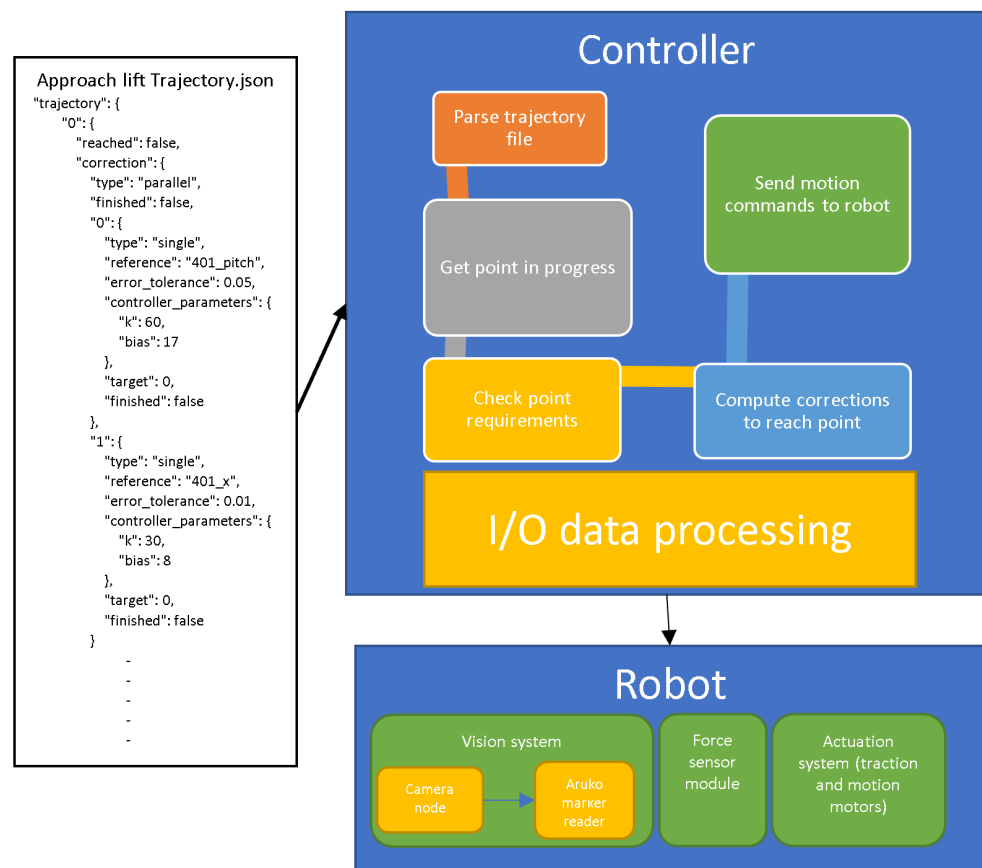
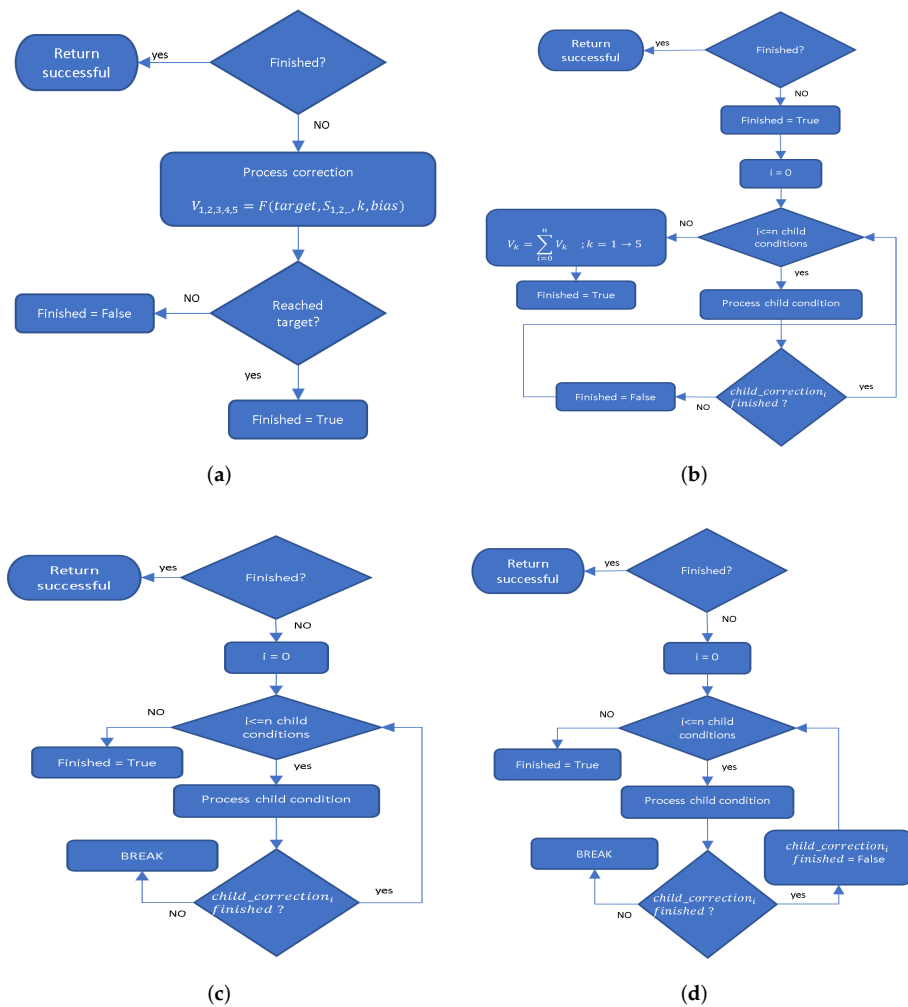


Figure 2. Software components.





**Figure 3.** Control Types: (a) single correction; (b) parallel corrections; (c) sequential corrections; and (d) semi-sequential corrections.

### 3. Concept, Approach, and Considerations

The platform described previously is designed for multi-robot object lifting and transportation. This concept has been proven to work in simulation in [16] where we were able to lift a 2 KG object using three robots and displace it about 2 m on a plane surface. Figure 4 shows the forces exerted by each robot on the target object and the total force by which direction the object would move. Having an omnidirectional platform allows each individual robot to move sideways while still maintaining pressure in the contact point. This allows for the possibility to rotate the target object by moving the individual robots at the same local direction.

Using simulation, it is possible to analyze and debug a system of three robots. However, working with hardware adds a certain amount of complexity due to the very dynamic and contact-full nature of the project. Therefore, a minimal scenario is needed (Figure 4). This setup uses one robot which applies force on an object contacting a non-friction surface on the opposite side of the robot. The system still is performing force control and adjusting the altitude of the target object. Using this simplified setup, debugging the system and analyzing force control algorithms becomes more manageable.

Unlike [16], in the following experiments we will not depend on an external observer to guide the robots towards the target object. The object is equipped with a visual marker, which is detected by the camera to guide the robot automatically to the target object.

$$\vec{F}_{total} = \sum_{i=1}^{i=N*2} \vec{F}_{Pi}, \&N = \text{number of robots} \tag{1}$$

The total amount of pressure applied on the object is all the forces applied that didn't participate in total force to move the object, which can be expressed as the following:

$$|F_{pressure}| = \sum_{i=1}^{i=N*2} |F_{Pi}| - |F_{total}| \tag{2}$$

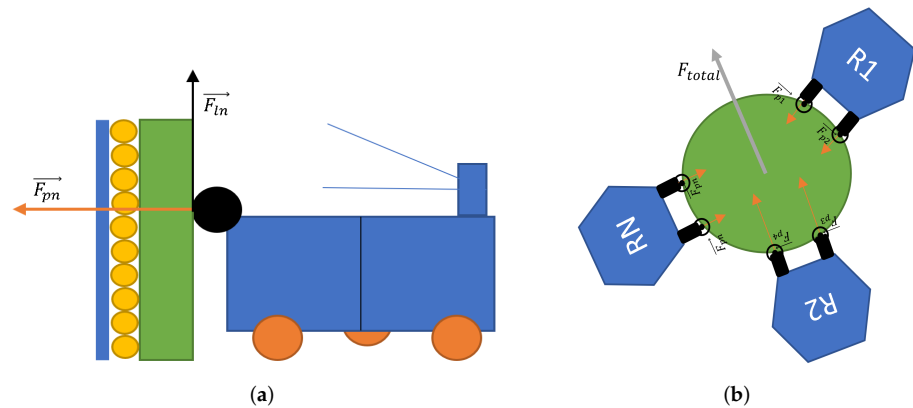


Figure 4. Concept: (a) single robot scenario; and (b) multi-robot scenario.

#### 4. Experiments and Results

For all our experiments, we will use the setup shown in Figure 5. Our setup is composed of the robot which is wired directly to the controlling PC, the target object in the shape of a rectangular box, and a non-friction surface of free rolling wheels that gives the ability for the box to move freely on the vertical plane. A marker is attached to the target object to easily identify its position and orientation.

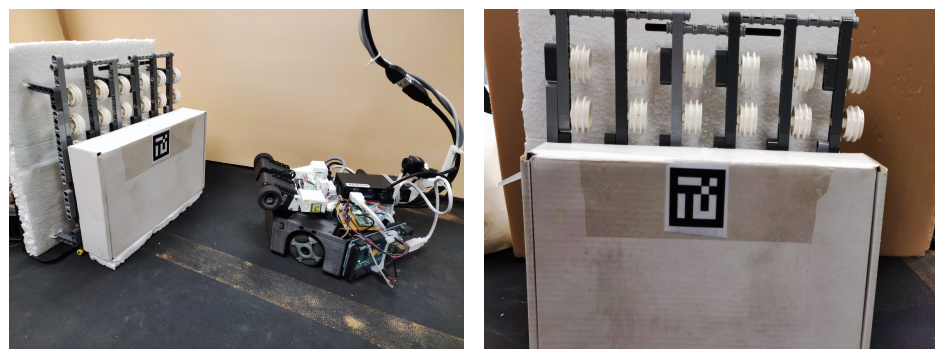


Figure 5. Setup and target object.

##### 4.1. Experiment 1

##### 4.1.1. Description

In this experiment, we set a trajectory of points consisting of force, position, and time conditions. This sequence of points is executed and the corresponding output is shown in Figure 6 where six different phases are identified, Figure 7:

1. Approach: the robot starts in a random location where the target object is visually accessible. The robot detects the target object and performs parallel control to reach a 2 cm distance from the object while keeping the robot parallel and centered relative to the target object.  
\*When at the target, the robot calibrates the height of the target object marker and starts next correction;

2. Press: having the robot perfectly parallel to the opposing surface, the system sets the target pressure to 120 g with a total tolerance of 40 g. Once the system reads force values in this range, the target object is “grasped” and ready to be lifted;
3. Lift: the trajectory specifies the system to perform parallel control to lift the object up 4 cm using the traction wheels in front while maintaining the same pressure and keeping the target bottom surface parallel to the ground. Please note that the Z axis value is inverted;
4. Hold: the robot holds the object in the same conditions for 5 s adding a time constraint to the previous parallel control. Note that in this phase, the control is already stable and, since there are not changes on the inputs, the system does not react;
5. Set down: keeping force conditions but changing the target height to original, the robot lowers down the target object;
6. Regress: now the robot can release pressure of the object and leave the area.

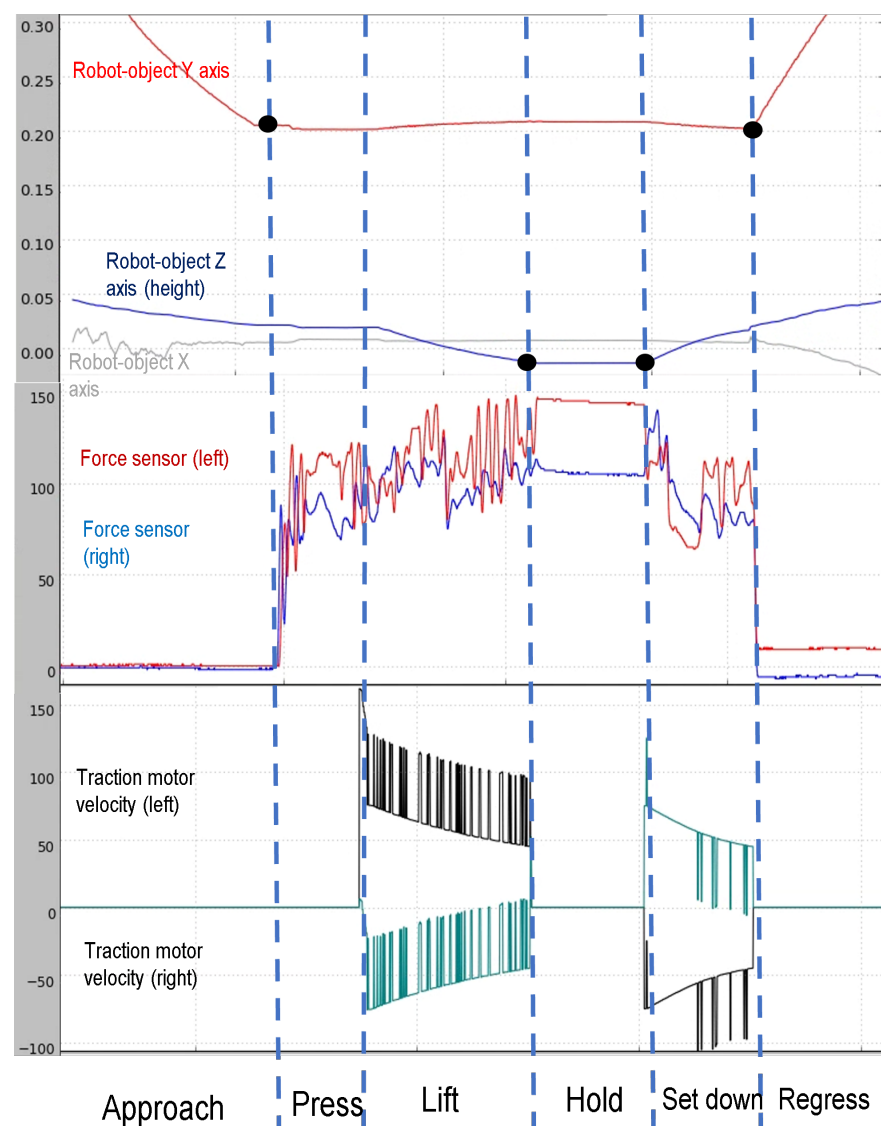
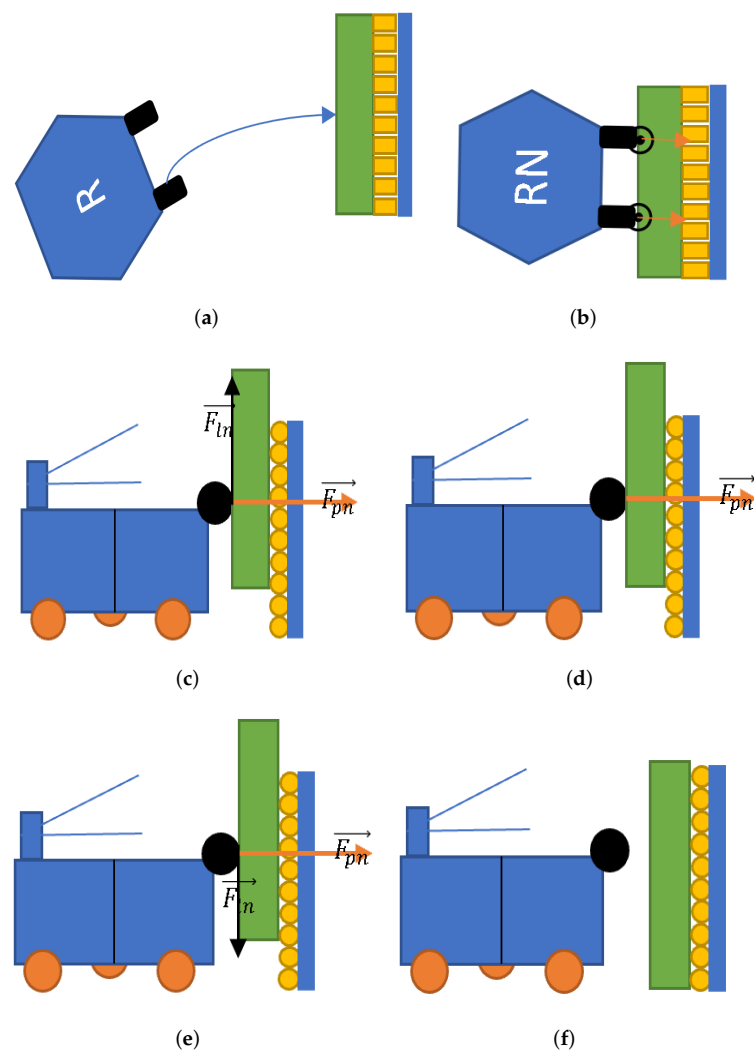


Figure 6. Experiment-1 execution flow.



**Figure 7.** Execution stages: (a) approach; (b) press; (c) lift; (d) hold; (e) set down; and (f) regress.

#### 4.1.2. Results

As seen in Figure 6, the robot was able to complete the entire trajectory as planned. However, in Press and Lift phases, the robot control overshoots the target force threshold and isolates with a maximum variance of 50 g. This problem arises due to the relatively high minimum speed at which the motors can run. Since the motors do not move on very low speeds, the error accumulates until the error is high enough to cause a response from the actuators. Despite this flow, the system didn't lose grip on the object because the target force is at a higher value above the oscillation range.

### 4.2. Experiment 2

#### 4.2.1. Description

In this experiment, we set the angle of the target object bottom surface with the ground to  $\alpha$  and we measured the average time of 10 iterations of same flow as experiment-1. We then calculated the success rate and defined the most common reason of failure for a certain value of  $\alpha$ .

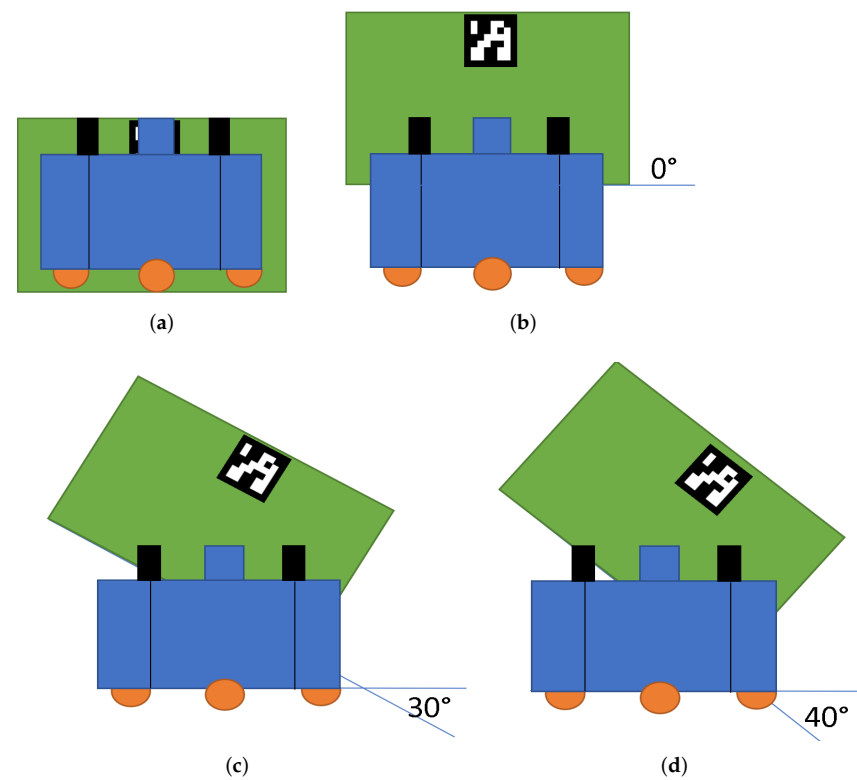
#### 4.2.2. Results

As demonstrated in Table 1, we can conclude that the possibility of failure increases when  $\alpha$  increases. The performance does not seem to be affected if the angle is below 20 degrees. The reason for this behavior is related to the shape of the target object. In our case, when the angle is higher than 20 degrees, one part of the object would start touching

the ground because the object can't be held at a very high altitude. As we see, for 40 degrees, we did not succeed in securely lifting the object; the object would not be in contact with the traction wheels around this angle because the wheels are in a fixed position on the robot, see Figure 8. The average time is higher with lower success rates due to timeouts at 100 s.

**Table 1.** Results of single robot approach and lift process changing angle  $\alpha$ .

Alpha (Degrees)	Average Time (Seconds)	Reason of Failure	Success Rate
0	35	-	100%
5	33	-	100%
10	41	PRESS	90%
20	53	HOLD	80%
30	64	LIFT	40%
40	110	LIFT	0%



**Figure 8.** Lifting angle limitation: (a) ready to lift; (b) lifted ( $\alpha = 0$  deg); (c) lifted ( $\alpha = 30$  deg); and (d) failed to lift ( $\alpha = 40$  deg).

### 4.3. Experiment 3

#### 4.3.1. Description

Here, we try to find the limits of pressure that our platform can apply. When the robot is capable of applying more pressure, the target load could have higher weight. In this experiment, we will set the total target pressure to different values to see the optimal pressure for the unloaded robot (900 g) and a very light target object (40 g).

#### 4.3.2. Results

After 10 iterations, we have results in Table 2. It was noted during the executions that on high pressure values ( $>140$ ), the robot skids on the ground because the platform does not have enough weight to generate more friction between the wheels and the floor. This limitation could be overcome by adding weights to the robot or modifying the flow to start lifting before the pressure threshold is reached to take advantage of the weight of the load



and put more pressure on the robots wheels to generate more friction. Low total target pressure results in the robot not having enough pressure to grasp the object and, therefore, the object sometimes falls while being lifted.

**Table 2.** Results of single robot approach and lift process changing total target pressure.

Total Target Pressure (Grams)	Average Time (Seconds)	Reason of Failure	Success Rate
90	110	PRESS + LIFT	0%
100	45	PRESS	80%
110	33	-	100%
120	35	-	100%
130	52	PRESS	70%
140	90	PRESS	10%

## 5. Discussion and Conclusions

As observed in the performed experiments, the robot is capable of performing an (approach–press–lift) operation with relatively high repeatability. However, the current implementation of the system has certain performance limitations which can be summarized in 3 issues:

- **Maximum manipulation angle:** as shown in Experiment 2, with more than 20 degree angle the system would have a high failure rate since the object is no longer gaining traction from the wheels' pressure. This issue can be solved by having more traction points with higher altitude so that the lifted object is always in contact with at least two points. Nevertheless, the solution was not implemented because the current limit is more than enough to perform a stable operation at maximum lift height;
- **Pressure overshooting:** in all experiments, it has been noted that our system tends to overshoot the pressure set point. This is due to the relatively high minimum speed of the used servo motors. Although this does not affect the overall results of the system, it has introduced a certain complexity to the process of tuning the low level controller constants to be able to overcome this issue;
- **Maximum pressure limit:** our 3D experiment draws the maximum pressure limit at 120 g. This value correlates with the maximum weight of the object that can be lifted (around 500 g). Counter-intuitively, the problem does not arise from the torque of the wheel motors, but arises from not having enough friction with the ground. The performance could be effortlessly improved by adding passive weight to the robot, increasing its friction force and, therefore, its lifting capacity.

Despite having certain limitations in our current implementation, we believe that it presents a more simplified and affordable approach to multi-robot load sharing and transportation compared to having an entire robotic arm attached to a heavy duty mobile robot. Having a modular system integrated using ROS and using off-the-shelf components positions it perfectly for use in research. The simplified control makes modifying the behavior of the robot straightforward by tuning certain parameters. The process of detecting, analyzing, and fixing a wrong behavior provides an understanding of the physics and control used in this problem. In this paper, we aimed at moving forward with our platform from a simulation environment to a simple hardware environment to validate our method of object lifting and transportation. This is a second step towards having a full multi-robot system. Such a system could be applied to lift and manipulate objects which are not prepared for a the usual transporting mechanism, like a forklift or large objects that could not be otherwise transportable using one agent. Since our system depends on traction, it does not require a special grasping area for each agent and, therefore, it provides a solution for such situation where modifying the target object for transportation is a challenge.

Our approach has been to downgrade the problem to one robot for a better understanding of the possible issues in a basic system rather than exploiting the entire problem

all at once. Our next step is to increase the complexity of the system by adding more lifting and transporting agents and improving the current design to solve the two pressure problems mentioned.

We validated our platform implementing a simple grasp-release-like experiment and tested the system under various target pressure conditions. The results demonstrate the feasibility of the platform for research and education on cooperative robotics and object manipulation.

**Author Contributions:** Conceptualization, M.K., E.C., A.P.d.P.; Methodology, M.K., E.C., A.P.d.P.; Validation, M.K., E.C.; Investigation, M.K., E.C.; Experimentation, M.K., E.C.; Writing—Original Draft Preparation, M.K., E.C.; Writing—Review & Editing, M.K., A.P.d.P.; project administration, A.P.d.P.; funding acquisition, A.P.d.P. All authors have read and agreed to the published version of the manuscript.

**Funding:** This paper describes research conducted at UJI Robotic Intelligence Laboratory. Support for this laboratory is provided in part by Universidad Jaume I (UJI-B2021-42), and by Generalitat Valenciana (PROMETEO/2020/034).

**Data Availability Statement:** The data that support the findings of this study are available in shared folder at: <https://drive.google.com/file/d/1Ue3n79pFKu0FwFRPIOM39qXh5G1KReXq/view?usp=sharing> (accessed on 21 December 2021). The repository to the project source code can be found in git hub repository at: <https://github.com/majdkassawat/maomt> (accessed on 21 December 2021).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Tuci, E.; Alkilabi, M.H.M.; Akanyeti, O. Cooperative Object Transport in Multi-Robot Systems: A Review of the State-of-the-Art. *Front. Robot. AI* **2018**, *5*, 59. [[CrossRef](#)] [[PubMed](#)]
2. Fink, J.; Hsieh, M.A.; Kumar, V. Multi-robot manipulation via caging in environments with obstacles. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 1471–1476. [[CrossRef](#)]
3. Chen, J.; Gauci, M.; Li, W.; Kolling, A.; Groß, R. Occlusion-Based Cooperative Transport with a Swarm of Miniature Mobile Robots. *IEEE Trans. Robot.* **2015**, *31*, 307–321. [[CrossRef](#)]
4. Farivarnejad, H.; Wilson, S.; Berman, S. Decentralized sliding mode control for autonomous collective transport by multi-robot systems. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 1826–1833. [[CrossRef](#)]
5. Gross, R.; Dorigo, M. Towards group transport by swarms of robots. *Int. J. Bio-Inspired Comput.* **2009**, 1–13. [[CrossRef](#)]
6. Hichri, B.; Fauroux, J.-C.; Adouane, L.; Doroftei, I.; Mezouar, Y. Design of cooperative mobile robots for co-manipulation and transportation tasks. *Robot. Comput. Manuf.* **2019**, *57*, 412–421. [[CrossRef](#)]
7. Hichri, B.; Adouane, L.; Fauroux, J.-C.; Mezouar, Y.; Doroftei, I. Flexible co-manipulation and transportation with mobile multi-robot system. *Assem. Autom.* **2019**, *39*, 422–431. [[CrossRef](#)]
8. Mellinger, D.; Shomin, M.; Michael, N.; Kumar, V. Cooperative Grasping and Transport Using Multiple Quadrotors. In *Distributed Autonomous Robotic Systems*; Springer Tracts in Advanced Robotics; Springer: Berlin/Heidelberg, Germany, 2013; Volume 83. [[CrossRef](#)]
9. Kim, S.; Seo, H.; Shin, J.; Kim, H.J. Cooperative Aerial Manipulation Using Multirotors with Multi-DOF Robotic Arms. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 702–713. [[CrossRef](#)]
10. Wang, Z.; Schwager, M. Kinematic multi-robot manipulation with no communication using force feedback. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 427–432. [[CrossRef](#)]
11. Alonso-Mora, J.; Knepper, R.; Siegwart, R.; Rus, D. Local motion planning for collaborative multi-robot manipulation of deformable objects. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 5495–5502. [[CrossRef](#)]
12. Alonso-Mora, J.; Baker, S.; Rus, D. Multi-robot formation control and object transport in dynamic environments via constrained optimization. *Int. J. Robot. Res.* **2017**, *36*, 1000–1021. [[CrossRef](#)]
13. Bischoff, R.; Huggenberger, U.; Prassler, E. KUKA youBot-a mobile manipulator for research and education. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011. [[CrossRef](#)]
14. McLurkin, J.; McMullen, A.; Robbins, N.; Habibi, G.; Becker, A.; Chou, A.; Li, H.; John, M.; Okeke, N.; Rykowski, J. et al. A robot system design for low-cost multi-robot manipulation. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 912–918. [[CrossRef](#)]

15. Campbell, E.; Kong, Z.C.; Hered, W.; Lynch, A.J.; O'Malley, M.K.; McLurkin, J. Design of a low-cost series elastic actuator for multi-robot manipulation. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 5395–5400. [[CrossRef](#)]
16. Kassawat, M.; Cervera, E.; del Pobil, A.P. Multi-robot user interface for cooperative transportation tasks. In *From Bioinspired Systems and Biomedical Applications to Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 77–81, ISBN 978-3-030-19650-9. [[CrossRef](#)]
17. Doroftei, I.; Grosu, V.; Spinu, V. Omnidirectional Mobile Robot – Design and Implementation. In *Bioinspiration and Robotics: Walking and Climbing Robots*; Habib, M.K., Ed.; I-Tech: Vienna, Austria, September 2007; p. 544, ISBN 978-3-902613-15-8. [[CrossRef](#)]
18. Sharbafi, M.A.; Indiveri, G. Swedish Wheeled Omnidirectional Mobile Robots: Kinematics Analysis and Control. *IEEE Trans. Robot.* **2009**, *25*, 164–171.
19. West, M.; Asada, H. Design of Ball Wheel Mechanisms for Omnidirectional Vehicles With Full Mobility and Invariant Kinematics. *J. Mech. Des.* **1997**, *119*, 153–161. [[CrossRef](#)]
20. Lebosse, C.; Renaud, P.; Bayle, B.; De Mathelin, M. Modeling and Evaluation of Low-Cost Force Sensors. *IEEE Trans. Robot.* **2011**, *27*, 815–822. [[CrossRef](#)]