

Article

Path Driven Dual Arm Mobile Co-Manipulation Architecture for Large Part Manipulation in Industrial Environments

Aitor Ibarguren *  and Paul Daelman 

Industry and Transport Division, TECNALIA, Basque Research and Technology Alliance (BRTA),
20009 San Sebastián, Spain; paul.daelman@tecnalia.com

* Correspondence: aitor.ibarguren@tecnalia.com

Abstract: Collaborative part transportation is an interesting application as many industrial sectors require moving large parts among different areas of the workshops, using a large amount of the workforce on this tasks. Even so, the implementation of such kinds of robotic solutions raises technical challenges like force-based control or robot-to-human feedback. This paper presents a path-driven mobile co-manipulation architecture, proposing an algorithm that deals with all the steps of collaborative part transportation. Starting from the generation of force-based twist commands, continuing with the path management for the definition of safe and collaborative areas, and finishing with the feedback provided to the system users, the proposed approach allows creating collaborative lanes for the conveyance of large components. The implemented solution and performed tests show the suitability of the proposed architecture, allowing the creation of a functional robotic system able to assist operators transporting large parts on workshops.



Citation: Ibarguren, A.; Daelman, P. Path Driven Dual Arm Mobile Co-Manipulation Architecture for Large Part Manipulation in Industrial Environments. *Sensors* **2021**, *21*, 6620. <https://doi.org/10.3390/s21196620>

Academic Editor: Abolfazl Zarak and Hamed Rahimi Nohooji

Received: 23 August 2021

Accepted: 3 October 2021

Published: 5 October 2021

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



Copyright: © 2021 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/>).

Keywords: mobile co-manipulation; force control; human-robot interaction; robotic application; assistant robots

1. Introduction

The emergence of collaborative robotics is changing the way of developing new robotic applications, especially in those cases involving cooperative tasks between humans and robots. This kind of cooperative task raises many technical challenges, ranging from the use of force feedback to guide the collaborative process [1] to the analysis of the social and psychological aspects of the acceptance of these new technologies [2].

Robotic part transportation is a compelling application as many industrial sectors such as the aeronautic [3,4] or automotive [5,6] require moving large parts among different areas of the workshops, using a large amount of the workforce on tasks with no added value. Therefore, the design and development of collaborative solutions based on flexible robotic systems able to transport large pieces [7–9] would benefit a wide range of companies worldwide. However, these systems need to deal with multiple technical topics like the management of the force feedback to generate movement commands or the definition of collaborative areas within the workshops where the robots can move freely.

This paper presents a path-driven dual-arm co-manipulation architecture for large part transportation. This architecture addresses three key aspects of the collaborative part transportation task: (1) Human-driven mobile co-manipulation, (2) soft superposition of navigation trajectories to the co-manipulation task to ensure safety zones within the workshop, and (3) robot-to-human feedback to guide and facilitate the collaborative task. The architecture tackles these three topics, proposing a new scheme that addresses issues for the industrial implementation of this kind of systems like safety and real-time feedback of the process state. The implementation and evaluation of the architecture demonstrate its suitability for cooperative applications in industrial environments.

The paper is organized as follows. Section 2 provides information about related work. Section 3 presents the proposed architecture, including the different modules of

the approach. Details about the implementation of the architecture are given in Section 4. Section 5 provides further information about the experiments carried out to test the suitability of the proposed architecture. Finally, Section 6 contains information about the conclusions and future work.

2. Related Work

Human-robot manipulation is a broad research topic, with multiple approaches proposed for a wide range of scenarios and applications. From classical scenarios with industrial robotic manipulators [10], to the use of new humanoid robots [11], many papers deal with the co-manipulation topic. From the industrial perspective, mobile manipulators offer an appealing and flexible solution as they allow extending manipulation capabilities of robots along the whole production area. As an example, Engemann et al. [12] present an autonomous mobile manipulator for flexible production, which includes a novel workspace monitoring system to ensure safe human-robot collaboration.

Force control is one of the most studied fields in human-robot collaboration, with a wide range of algorithms and approaches to offer a seamless interaction based on force. Peternel et al. [13] propose an approach for co-manipulation tasks such as sawing or bolt screwing through a human-in-the-loop framework, integrating online information about manipulability properties. Lichiardopol et al. [1] pose a control scheme for human-robot co-manipulation with a single robot, a system able to estimate an unknown and time-varying mass and the force applied by the operator. Focusing on mobile manipulators, Weyrer et al. [14] present a natural approach for hand guiding a sensitive mobile manipulator in task space using a force-torque sensor.

Additionally, the use of guides and predefined trajectories for dual-arm and co-manipulation tasks appear in different research activities. Gan et al. [15] present a position/force coordination control for multi-robot systems where an object-oriented hierarchical trajectory planning is adopted as the first step of a welding task. Jlassi et al. [16] introduce a modified impedance control method for heavy load co-manipulation; an event-controlled online trajectory generator is included to translate the human operator intentions into ideal trajectories. Continuing with the topic of trajectory generation, Raiola et al. [17] propose a framework to design virtual guides through demonstration using Gaussian mixture models. The use of these paths and guides are interesting concepts that could be transferred to mobile co-manipulation.

Multi-robot systems are an alternative for force-based large part co-manipulation. Hichri et al. address the flexible co-manipulation and transportation task with a mobile multi-robot system, tackling issues like the mechanical design [7] and optimal positioning of a group of mobile robots [18]. However, these approaches leave human operators out of the control strategy.

The use of Artificial Intelligence for mobile manipulation and cooperative tasks is also a recurrent research topic, as it adds mechanisms to tune and optimize control parameters. Zhou et al. [19] propose a mobile manipulation method integrating deep-learning-based multiple-object detection to track and grasp dynamic objects efficiently. Following this same path, Wang et al. [20] present a novel mobile manipulation system that decouples visual perception from the deep reinforcement learning control, improving its generalization from simulation training to real-world testing. Additionally, Iriondo et al. [21] include a deep reinforcement learning (DRL) approach for pick and place operations in logistics using a mobile manipulator. Moreover, deep learning algorithms have also been used in force-based human-robot interaction, specifically for the identification of robot tool dynamics [22], allowing a fast computation and adding noise robustness.

From the user point of view, it is also important to include human and social factors in the design of collaborative applications as the lack of transparency in robot behavior can worsen the user experience, as shown by Sanders et al. and Wortham and Theodorou [23,24]. Therefore, it is crucial to add mechanisms to communicate robot intentions and provide continuous feedback as it helps improving task performance [25] and

increasing trust [23]. As an example of these previous works, Weiss et al. [26] present a work where operators use and program collaborative robots in industrial environments in various case studies, providing an additional anthropocentric dimension to the discussion of human-robot cooperation.

The main contribution of the presented work is the usage of navigation paths in force-based mobile co-manipulation. This addition allows modifying the commands generated from the force information and providing mechanisms to limit the robot's workspace. This feature enables the definition of safety areas within the workshops, a critical issue in the implementation of this kind of application in a production environment.

3. Proposed Architecture

Collaborative part transportation is an appealing application, as many industrial sectors such as the aeronautic or automotive one require moving large parts among different areas of the workshops, spending a large quantity of workforce on tasks with no added value. Therefore, it is interesting to offer robotic solutions that allow collaborative transportation, paying attention to different aspects such as the control algorithm or robot-to-human feedback.

In an initial step of the development, the large part transportation process has been analyzed to extract the key elements of the task, as well as the requirements to be transferred to the robotic solution:

- During large part transportation by humans, both actors agree (implicit or explicitly) on an approximate path to follow during the process. Additionally, humans can indicate their destination during the task. Therefore, the robot should include some feedback indicating which direction they are (or should be) moving.
- As robots act as assistants of the human, robots will only advance in the path when the operator moves the part in the defined direction. Therefore, humans always have the master role in the co-manipulation task.
- Industrial workshops generally include lanes used for the transit of vehicles and large machines. The mobile co-manipulation system should offer mechanisms to limit the co-manipulation areas and restrict them to the allowed zones when required. These safety measures are mandatory for the industrial implementation of this kind of solution.

To cope with the previously listed requirements, a four-layer architecture is proposed:

- **Force Management Layer:** This initial layer is in charge of generating twist commands for the mobile robotic platform based on the force information received from the arms. This force information could be provided by force/torque sensors attached to the grippers or directly by robots equipped with internal sensors.
- **Moving Average Filter Layer:** This second layer filters the received twist commands to smooth the velocity.
- **Path Management Layer:** This third layer modifies, if necessary, the smoothed twist commands to ensure that the robotic platform is within the safety lanes defined in the workshop.
- **User Interface Management Layer:** This last layer is in charge of presenting the co-manipulation feedback to operators, using different cues to this end.

This four-layer architecture allows the mobile co-manipulation task, including different modules for the robot control besides the generation of robot-to-human feedback. It ensures that the system controls all the process steps, from low-level control to high-level interaction feedback. Figure 1 illustrates the presented architecture.

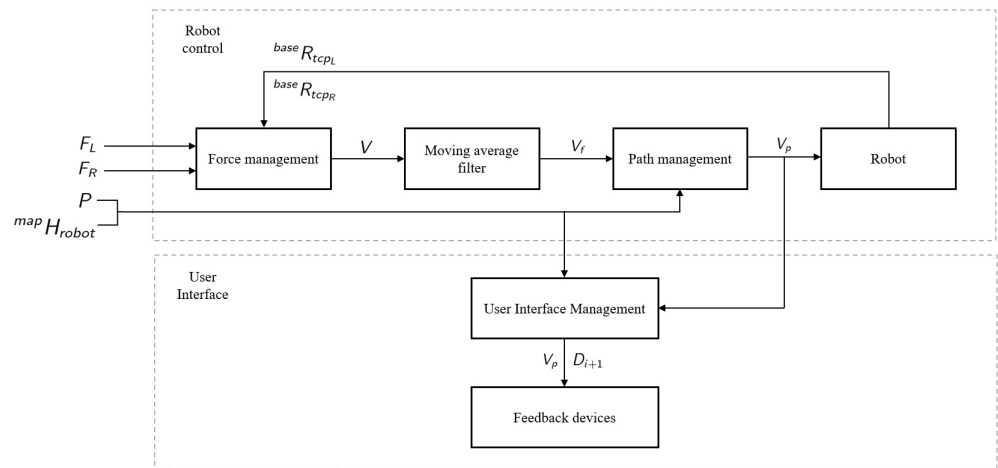


Figure 1. Proposed architecture for mobile part co-manipulation.

The following sections provide further information about the different layers and their specific features.

3.1. Force Management Layer

The first step is to estimate the external forces applied on the left arm F_{eL} and right arm F_{eR} in the robot base frame as

$$F_{eL} = {}^{base}R_{tcpL} F_L - m_L G, \quad (1)$$

$$F_{eR} = {}^{base}R_{tcpR} F_R - m_R G, \quad (2)$$

where F_L and F_R are the forces sensed in the TCP of the left and right arm, ${}^{base}R_{tcpL}$ and ${}^{base}R_{tcpR}$ define the rotation of the left and right arm TCP in robot's base frame, m_L and m_R are the mass of the left and right arm tool and G is the gravity vector $[0, 0, -g]^T$.

The external torques applied on the left arm T_{eL} and right arm T_{eR} in the robot base frame are calculated as

$$T_{eL} = {}^{base}R_{tcpL} (T_L - m_L r_L \times ({}^{base}R_{tcpL}^{-1} G)), \quad (3)$$

$$T_{eR} = {}^{base}R_{tcpR} (T_R - m_R r_R \times ({}^{base}R_{tcpR}^{-1} G)), \quad (4)$$

where ${}^{base}R_{tcpL}$ and ${}^{base}R_{tcpR}$ define the rotation of the left and right arm TCP in robot's base frame, T_L and T_R are the torques sensed in the TCP of the left and right arm, m_L and m_R are the mass of the left and right arm tool, r_L and r_R are the center of mass of the left and right arm tool and G is the gravity vector $[0, 0, -g]^T$.

The overall external force F_e and torque T_e are calculated as

$$F_e = (F_{eL} + F_{eR})/2, \quad (5)$$

$$T_e = (T_{eL} + T_{eR})/2, \quad (6)$$

using the applied forces of the left arm F_{eL} and right arm F_{eR} , as well as the applied torques of the left arm T_{eL} and right arm T_{eR} estimated previously.

These force and torque vectors $F_e = [f_{eX}, f_{eY}, f_{eZ}]^T$ and $T_e = [t_{eX}, t_{eY}, t_{eZ}]^T$ are used afterwards to calculate the twist vector V that will be sent to the mobile platform

$$V = [\nu_X, \nu_Y, \omega_Z]^T, \quad (7)$$

where ν_X and ν_Y define the linear velocity in axis X and Y, respectively, while ω_Z indicates the angular velocity in axis Z.

In the generation of this twist vector V , the idea is to define a force and torque range where the robot will move. If the sensed forces and torques are below a predefined threshold, the twist values will be equal to zero. These twist values will increase linearly as forces and torques increase, defining a maximum allowed force and torque, as well as a maximum robot speed. It will allow defining different force/torque and velocity profiles, profiles that can be used on different phases of the part transportation such as the fast movement between stations of the workshop or the precise and slow positioning of the robotic platform at the part loading area.

The linear velocities v_X and v_Y are calculated as

$$v_X = \begin{cases} 0, & \text{if } f_{e_X} < \min_F \\ \max_v * \text{MIN}(f_{e_X}, \max_F) / \max_F, & \text{otherwise,} \end{cases} \quad (8)$$

$$v_Y = \begin{cases} 0, & \text{if } f_{e_Y} < \min_F \\ \max_v * \text{MIN}(f_{e_Y}, \max_F) / \max_F, & \text{otherwise,} \end{cases} \quad (9)$$

where f_{e_X} and f_{e_Y} are the external forces in axis X and Y , \min_F and \max_F define the minimum required force and the maximum allowed force and \max_v represents the maximum allowed linear velocity.

The angular velocity ω_Z is calculated as

$$\omega_Z = \begin{cases} 0, & \text{if } t_{e_Z} < \min_T \\ \max_\omega * \text{MIN}(t_{e_Z}, \max_T) / \max_T, & \text{otherwise,} \end{cases} \quad (10)$$

where t_{e_Z} is the external torque in axis Z , \min_T and \max_T define the minimum required torque and the maximum allowed torque and \max_ω represents the maximum allowed angular velocity.

This twist vector $V = [v_X, v_Y, \omega_Z]^T$ is the information sent to the next *Moving Average Filter Layer*.

3.2. Moving Average Filter Layer

This layer applies a classical *moving average filter* [27] to the twist vector obtained in the previous layer to ensure a smooth navigation of the mobile platform. The filtered twist vector V_f is calculated as

$$V_f = \frac{1}{K} \sum_{i=0}^K V_i, \quad (11)$$

where K defines the size of the filter and V_i represents the last received K twist vectors.

This filtered vector $V_f = [v_{f_X}, v_{f_Y}, \omega_{f_Z}]$ will be sent to the *Path Management Layer*, layer that will modify it in order to fit in the path provided to the algorithm.

3.3. Path Management Layer

The main idea of this layer is to define a corridor where the robotic platform can move freely, limiting the movements when the robot is trying to go outside the borders of the lane. It allows the creation of some virtual walls at both sides of the lane, as shown in Figure 2, defining the area where the operator can co-manipulate the part.

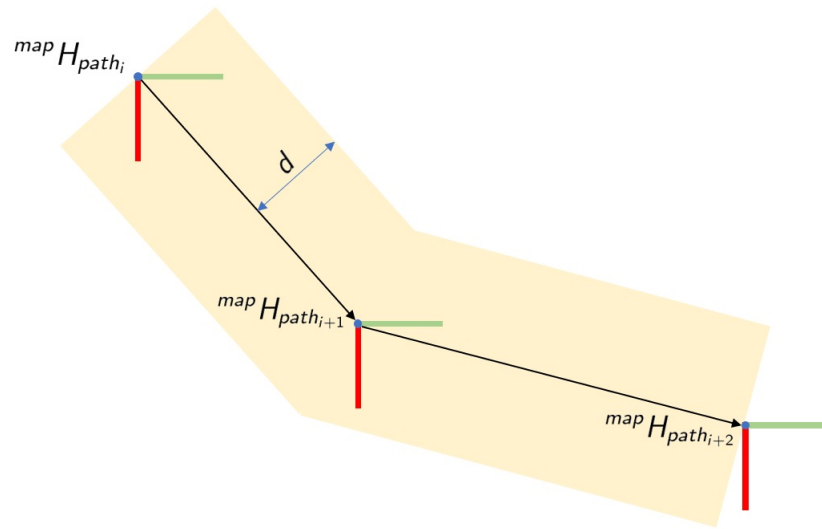


Figure 2. Definition of path.

Therefore, the filtered twist vector V_f is corrected based on the provided path P and the current pose of the robotic platform in the map ${}^{map}H_{base}$. Specifically, the path is composed of a list of M poses as

$$P = \{{}^{map}H_{path_1}, {}^{map}H_{path_2}, \dots, {}^{map}H_{path_M}\}, \quad (12)$$

where ${}^{map}H_{path_i}$ defines the i^{th} pose of the path in the map frame. Additionally, the current pose of the robotic platform ${}^{map}H_{base}$ is defined as

$${}^{map}H_{base} = \left[\begin{array}{c|c} {}^{map}R_{base} & {}^{map}T_{base} \\ \hline 0 & 1 \end{array} \right], \quad (13)$$

where ${}^{map}R_{base}$ and ${}^{map}T_{base}$ represent the rotation and translation part of the transformation matrix.

In an initial step, a new coordinate system L is calculated, a coordinate system with the origin in ${}^{map}T_{path_i}$ (translation part of i^{th} path pose) with axis X pointing to pose ${}^{map}H_{path_{i+1}}$ (Figure 3). The calculus of this new coordinate system allows for creating a local frame for the current path segment, which facilitates the calculations of the corrections to be applied. The coordinate system L is calculated using the following equations:

$$A = \frac{{}^{map}T_{path_{i+1}} - {}^{map}T_{path_i}}{|{}^{map}T_{path_{i+1}} - {}^{map}T_{path_i}|} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}, \quad (14)$$

where A defines the unitary vector pointing from path pose i to pose $i+1$,

$$Y = ({}^{map}T_{path_{i+1}} - {}^{map}T_{path_i}) \times \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \quad (15)$$

$$C = \frac{A \times Y}{|A \times Y|} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, \quad (16)$$

where C defines the unitary vector pointing upwards in the map and

$$B = \frac{C \times A}{|C \times A|} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}, \quad (17)$$

where B defines the unitary vector pointing perpendicularly to the lane border.

Based on these three unitary vectors, the new coordinate system is calculated as

$$L = \left[\begin{array}{ccc|c} a_1 & b_1 & c_1 & \text{map}T_{path_i} \\ a_2 & b_2 & c_2 & \\ a_3 & b_3 & c_3 & \\ \hline 0 & 0 & 0 & 1 \end{array} \right]. \quad (18)$$

where unitary vectors A , B , and C define the rotational part of the transformation matrix and vector $\text{map}T_{path_i}$ indicates the translational part of the matrix.

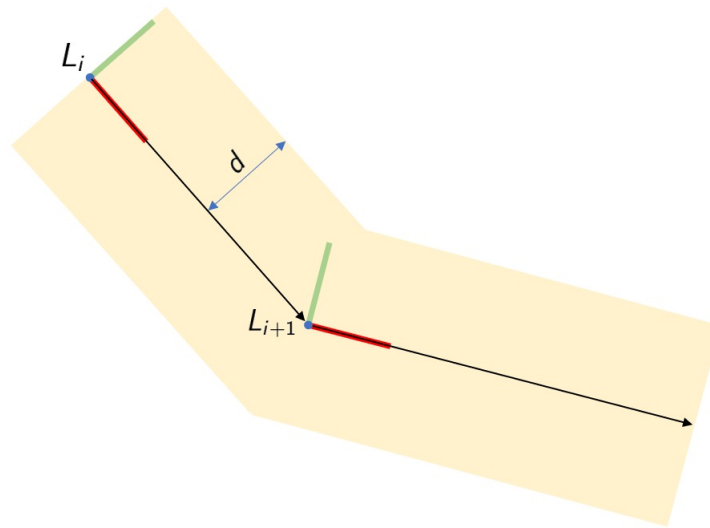


Figure 3. Calculation of path frame L .

Once this path frame L is defined, the poses of the robotic platform ${}^LH_{base}$ and next path pose ${}^LH_{path_{i+1}}$ in the path frame are calculated as

$${}^LH_{base} = L^{-1} \cdot \text{map}H_{base} = \left[\begin{array}{c|c} {}^LR_{base} & {}^LT_{base} \\ \hline 0 & 1 \end{array} \right], \quad (19)$$

$${}^LH_{path_{i+1}} = L^{-1} \cdot \text{map}H_{path_{i+1}} = \left[\begin{array}{c|c} {}^LR_{path_{i+1}} & {}^LT_{path_{i+1}} \\ \hline 0 & 1 \end{array} \right], \quad (20)$$

where ${}^LR_{base}$ and ${}^LT_{base}$ represent the rotation and translation part of the robotic platform pose in the path frame and ${}^LR_{path_{i+1}}$ and ${}^LT_{path_{i+1}}$ represent the rotation and translation part of the next path pose in the path frame.

At this point, the theoretical next robotic platform pose is calculated as

$${}^LT_{base}^{t+1} = {}^LH_{base} \cdot \begin{bmatrix} v_X \\ v_Y \\ 0 \end{bmatrix} \Delta t = \begin{bmatrix} {}^L t_{baseX}^{t+1} \\ {}^L t_{baseY}^{t+1} \\ {}^L t_{baseZ}^{t+1} \end{bmatrix}, \quad (21)$$

where ${}^LH_{base}$ defines the current robotic platform pose in path frame, v_X and v_Y represents the linear twist values in axis X and Y , and Δt defines the period of the control loop.

In the next step, the corrected next platform pose ${}^L T_{base}^*$ is calculated

$${}^L T_{base}^* = \begin{bmatrix} {}^L t_{baseX}^* \\ {}^L t_{baseY}^* \\ {}^L t_{baseZ}^* \end{bmatrix}, \quad (22)$$

ensuring that the robotic platform does not cross the borders of the defined lane, lane created based on the path P and the maximum distance d from the path. Specifically, the corrected next platform pose is calculated as

$${}^L t_{baseX}^* = \begin{cases} {}^L t_{baseX}^{t+1}, & \text{if } |{}^L t_{baseX}^{t+1}| \leq d \\ {}^L t_{baseX}, & \text{otherwise} \end{cases}, \quad (23)$$

$${}^L t_{baseY}^* = \begin{cases} {}^L t_{baseY}^{t+1}, & \text{if } |{}^L t_{baseY}^{t+1}| \leq d \\ {}^L t_{baseY}, & \text{otherwise} \end{cases}, \quad (24)$$

$${}^L t_{baseZ}^* = {}^L t_{baseZ}^{t+1}, \quad (25)$$

where it is ensured that the limits are not crossed in axes X and Y .

Finally, the corrected twist values V_p are calculated as

$$V^* = \frac{{}^L H_{base}^{-1} \cdot {}^L T_{base}^*}{\Delta t} = \begin{bmatrix} v_X^* \\ v_Y^* \\ v_Z^* \end{bmatrix}, \quad (26)$$

$$V_p = \begin{bmatrix} v_X^* \\ v_Y^* \\ \omega_Z \end{bmatrix}, \quad (27)$$

where ${}^L H_{base}$ defines the current pose of the robotic platform in the path frame, ${}^L T_{base}^*$ represents the corrected next platform pose, Δt sets the period of the control loop, and ω_Z represents the original angular velocity in axis Z as it will not be limited nor modified by the *Path Management Layer*.

This new twist vector V_p will be sent to the robot for its execution, as it ensures that the robot maintains within the limits defined by the path P and the maximum distance d . Additionally, this vector V_p is also sent to the *User Interface Management Layer* to generate the appropriate feedback for the co-manipulation process.

3.4. User Interface Management Layer

This last layer will manage the generated twist commands, besides the current robot status, to provide helpful feedback to operators. Specifically, two different feedback cues have been considered:

- **Movement direction:** The movement direction is directly generated using twist vector V_p , constantly indicating where the robot is moving.
- **Direction to the next path pose:** The second cue defined to guide the co-manipulation process is the direction vector to the next path pose D_{i+1} , which is calculated as

$$D_{i+1} = {}^{map}H_{base}^{-1} \cdot {}^{map}T_{path_{i+1}} = \begin{bmatrix} d_X \\ d_Y \\ d_Z \end{bmatrix}, \quad (28)$$

where ${}^{map}H_{base}$ is the pose of the robotic platform in the map frame and ${}^{map}T_{path_{i+1}}$ indicates the translation from the map frame to the next path pose.

Additionally, some audio signals have been added to indicate when the path-driven mobile co-manipulation starts and when the destination has been reached. These audio

cues would complete the feedback sent from the robot to the user to guide the mobile co-manipulation task.

4. Implementation

The proposed architecture has been implemented in a dual-arm mobile platform designed for the manipulation of large parts in industrial environments, see Figure 4. The robotic platform is composed of this hardware:

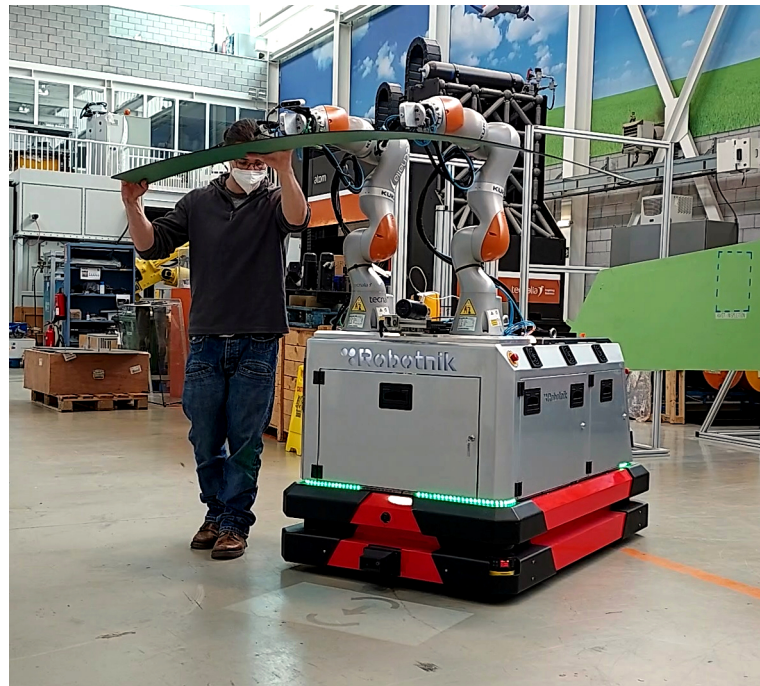


Figure 4. Dual-arm omnidirectional mobile robot.

- An omnidirectional mobile platform equipped with Mecanum wheels [28]. The platform includes led lights placed in the front part, rear part, and both sides of the base. These lights are used to provide the feedback of the *movement direction* generated in the *User Interface Management Layer*. Therefore, the robot's front and rear lights blink when the robot moves forward or backward, while the sidelights blink when the robot moves in these directions.
- Two *Kuka LBR iiwa* robots with a payload of 7 kg, collaborative arms equipped with torque sensors in each joint. These robotic arms can provide force and torque information estimated from the torque sensors placed in each joint. Therefore, this implementation does not require an external sensor to acquire force and torque information.
- The arms are also equipped with automatic tool exchangers, a suction system, and vacuum cups to grasp different types of large objects and parts.
- An additional IO module is also available to manage the tool exchangers and the suction of the vacuum cups.
- A small size projector has been installed in the front part of the robot. This projector is used to display the *direction to the next path pose* on the ground in the front part of the robot as shown in Figure 5.



Figure 5. Projector installed on robotic platform to display the direction to the next path pose.

From the software point of view, a PC placed inside the robotic platform executes the four modules of the architecture. All these modules have been implemented in C++ as ROS nodes. An additional HTML5 server provides the web interface that shows the arrows and cues used to guide the co-manipulation process.

Finally, an HTML5 based general User Interface (UI) has been developed. This interface allows commanding the system, triggering and canceling the execution of the mobile co-manipulation at any moment. A tablet is used to display this UI, enabling operators to utilize this interface at any place within the workshop.

5. Experiments

As the last step, the presented architecture has been evaluated through a set of tests. Specifically, an experiment was defined in which various users had to transport a carbon fiber part between two stations placed in an industrial workshop using the mobile co-manipulation architecture presented above. The next lines provide information about the experiment:

- Two different stations were defined in the workshop, *shelf* and *inspection station*, 10 m away from each other (see Figure 6).

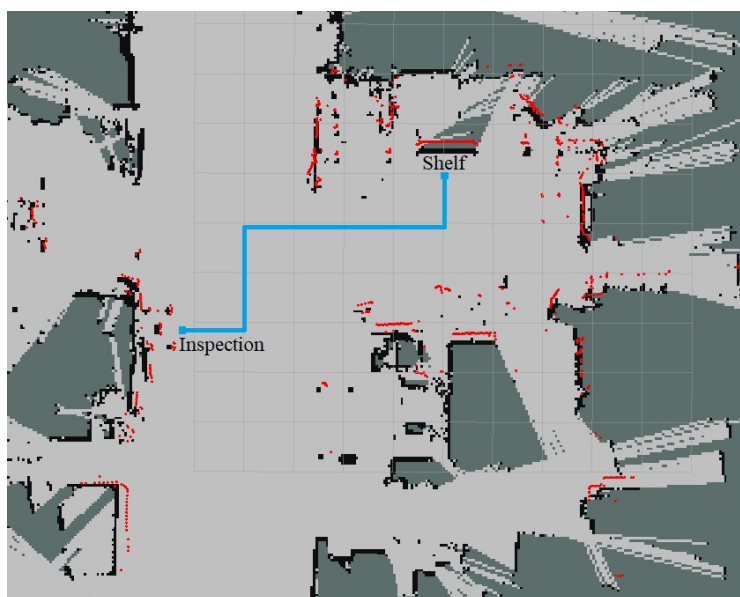


Figure 6. Co-manipulation path on workshop map.

- Five different users transported a 2-m carbon fiber part between stations alongside the robot. The subjects did not have any previous knowledge about the defined path.
- To get a better insight into the process, the percentage of the covered trajectory and the deviation from the nominal path were recorded with a frequency of 15 Hz. The deviation from the nominal path indicates the perpendicular distance between the current robot pose and the central line of the path, as shown in Figure 7.

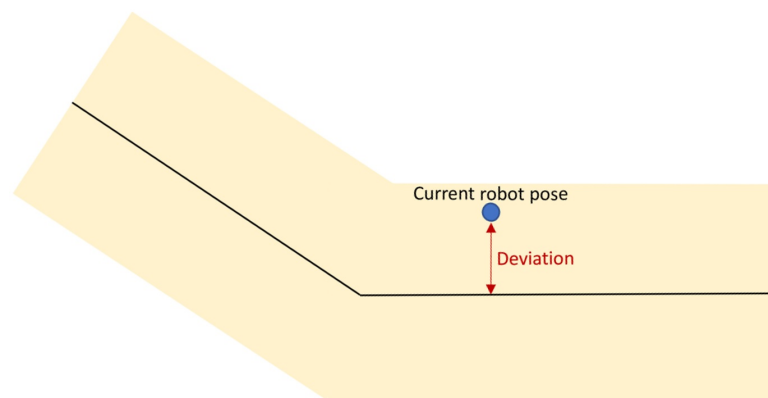


Figure 7. Deviation from the nominal path.

- During the transportation, two different maximum distances from the path (d) were used, 0.250 m and 0.500 m, defining lanes with a width of 0.500 m and 1.0 m, respectively. It will serve to assess the impact of the lane width on the trajectory deviation.

Therefore, each of the five subjects performed four different transportation processes, moving the part in both directions (*shelf to inspection* and *inspection to shelf*) with two different lane widths. Figure 8 shows trajectory percentage and deviation plots extracted from two transportation processes.

Table 1 summarizes the obtained results. The first column indicates the maximum allowed distance from path d used during the transportation. The second column includes the route of the transportation. The third and fourth columns provide information about the mean time to cover the route and the standard deviation. Finally, the last three columns include information about the deviation from the nominal path during the process. Specifically, the maximum and mean deviation and the standard deviation.

Table 1. Results of the experiment.

d	Route	Time		Deviation		
		Mean	σ	Max.	Mean	σ
0.250 m	shelf to insp.	26.6 s	5.69 s	0.249 m	0.129 m	0.075 m
0.250 m	insp. to shelf	27.4 s	6.93 s	0.250 m	0.107 m	0.068 m
0.500 m	shelf to insp.	23.5 s	3.49 s	0.499 m	0.158 m	0.102 m
0.500 m	insp. to shelf	23.2 s	6.93 s	0.492 m	0.168 m	0.113 m

Additionally, these results have been compressed in a table where the different routes' information is merged for each maximum distance. Table 2 includes a synthesized version on the previous table with the first column indicating the maximum distance, leaving the rest of the columns to the time and deviation information, as presented in the previous table.

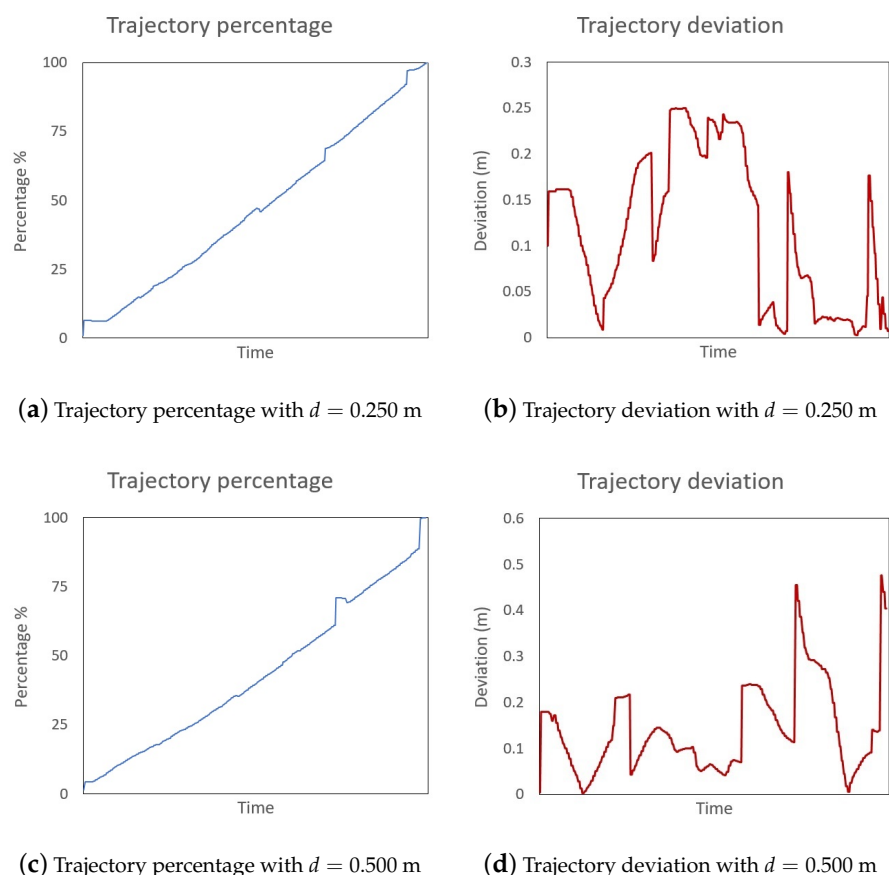


Figure 8. Trajectory percentage and deviation of two transportation processes.

Table 2. Results of the experiment grouped by maximum distance.

d	Time		Deviation		
	Mean	σ	Max.	Mean	σ
0.250 m	27.0 s	5.99 s	0.250 m	0.118 m	0.071 m
0.500 m	23.4 s	3.97 s	0.499 m	0.163 m	0.108 m

A close look at both tables surfaces some relevant information. Initially, in all cases, the maximum allowed deviation has been reached during the transportation process (mainly in sharp bends of the route). Therefore, users tend to use all the available space to guide the robot; it is especially notorious when users reach bends along the route. Additionally, lower mean deviation values have been observed with a maximum allowed distance of 0.250 m. However, the difference with the maximum distance 0.500 m is not that large (only 0.050 m more) based on the difference in the lane width. Finally, users can cover the path in less time with a wider lane, as they can guide the robot easily and even take shortcuts in sharp bends due to the width of the path.

As a general conclusion, the experiment subjects were able to carry out the part transportation in less time and more comfortably with a greater maximum distance d . Additionally, it did not impact the mean deviation value greatly as the projected arrows guide the users and allow them to re-enter the path after sharp bends or unexpected obstacles.

6. Conclusion and Future Work

This paper presents a novel architecture for mobile co-manipulation. The architecture includes the capability to add a path to guide the part transportation process and ensure safety by creating virtual lanes in the robot's workspace. Additionally, the architecture incorporates a specific module to generate robot-to-human feedback to assist the process and improve the user experience in the collaborative task. Therefore, the proposed work covers all the different steps of the mobile co-manipulation for large part transportation, starting from the force-based control algorithm and finishing with the generation of suitable feedback for the system users.

The proposed approach has been implemented and tested in an industrial workshop environment, assisting operators in the transportation of large carbon fiber parts within different workstations. The obtained results show the suitability of the architecture, highlighting the utility of the robot-to-human feedback generation to guide operators along the defined safe paths of the workshop. Moreover, the addition of projected arrows helps operators to keep close to the nominal path even with wide co-manipulation lanes.

As future steps, several research paths have been identified. On the one hand, it would be interesting to investigate different cues and communication methods to improve co-manipulation tasks to create a seamless interaction between humans and robots. On the other hand, many parameters of the algorithm must be tuned manually by experts. Therefore, it would be interesting to include Artificial Intelligence techniques to extract these data from the information gathered during the co-manipulation processes as currently all this data is not further used nor exploited.

Author Contributions: Conceptualization, A.I.; methodology, A.I. and P.D.; software, A.I. and P.D.; validation, A.I. and P.D.; formal analysis, A.I.; investigation, A.I. and P.D.; resources, A.I. and P.D.; data curation, A.I.; writing—original draft preparation, A.I.; writing—review and editing, P.D.; visualization, A.I. and P.D.; supervision, P.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work has received funding from the European Union Horizon 2020 research and innovation programme as part of the project SHERLOCK under grant agreement No 820689.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the type of recorded data as trajectory and deviation information are considered as internal robot control parameters and metrics.

Informed Consent Statement: Patient consent was waived as users interacted with the robotic system to generate internal robot control parameters and metrics.

Data Availability Statement: Experiment data is available at 10.5281/zenodo.5163197.

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

References

1. Lichardopol, S.; van de Wouw, N.; Nijmeijer, H. Control scheme for human-robot co-manipulation of uncertain, time-varying loads. In Proceedings of the 2009 American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 1485–1490, doi:10.1109/ACC.2009.5160062.
2. Hayes, B.; Scassellati, B. Challenges in shared-environment human-robot collaboration. *Learning* **2013**, *8*, pp. 1–6.
3. Rizzo, C.; Lagrãña, A.; Serrano, D. Geomove: Detached agvs for cooperative transportation of large and heavy loads in the aeronautic industry. In Proceedings of the 2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Ponta Delgada, Portugal, 15–17 April 2020; pp. 126–133.
4. De Schepper, D.; Moyaers, B.; Schouterden, G.; Kellens, K.; Demeester, E. Towards robust human-robot mobile co-manipulation for tasks involving the handling of non-rigid materials using sensor-fused force-torque, and skeleton tracking data. *Procedia CIRP* **2021**, *97*, 325–330, doi:10.1016/j.procir.2020.05.245.
5. Mizanoor Rahman, S.M. Performance Metrics for Human-Robot Collaboration: An Automotive Manufacturing Case. In Proceedings of the 2021 IEEE International Workshop on Metrology for Automotive (MetroAutomotive), Bologna, Italy, 1–2 July 2021; pp. 260–265, doi:10.1109/MetroAutomotive50197.2021.9502881.
6. Blatnický, M.; Dižo, J.; Gerlici, J.; Sága, M.; Lack, T.; Kuba, E. Design of a robotic manipulator for handling products of automotive industry. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1729881420906290.

7. Hichri, B.; Fauroux, J.C.; Adouane, L.; Doroftei, I.; Mezouar, Y. Design of cooperative mobile robots for co-manipulation and transportation tasks. *Robot. Comput.-Integr. Manuf.* **2019**, *57*, 412–421, doi:10.1016/j.rcim.2019.01.002.
8. Jlassi, S.; Tliba, S.; Chitour, Y. On Human-Robot Co-manipulation for Handling Tasks: Modeling and Control Strategy. *IFAC Proc. Vol.* **2012**, *45*, 710–715. doi:10.3182/20120905-3-HR-2030.00092.
9. Kim, W.; Balatti, P.; Lamon, E.; Ajoudani, A. MOCA-MAN: A MOBILE and reconfigurable Collaborative Robot Assistant for conjoined huMAN-robot actions. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 10191–10197, doi:10.1109/ICRA40945.2020.9197115.
10. Dimeas, F.; Aspragathos, N. Reinforcement learning of variable admittance control for human-robot co-manipulation. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1011–1016, doi:10.1109/IROS.2015.7353494.
11. Otani, K.; Bouyarmane, K.; Ivaldi, S. Generating Assistive Humanoid Motions for Co-Manipulation Tasks with a Multi-Robot Quadratic Program Controller. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 3107–3113, doi:10.1109/ICRA.2018.8463167.
12. Engemann, H.; Du, S.; Kallweit, S.; Cönen, P.; Dawar, H. OMNIVIL—An Autonomous Mobile Manipulator for Flexible Production. *Sensors* **2020**, *20*, 7249, doi:10.3390/s20247249.
13. Peternel, L.; Kim, W.; Babič, J.; Ajoudani, A. Towards ergonomic control of human-robot co-manipulation and handover. In Proceedings of the 2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids), Birmingham, UK, 15–17 November 2017; pp. 55–60, doi:10.1109/HUMANOIDS.2017.8239537.
14. Weyrer, M.; Brandstötter, M.; Husty, M. Singularity Avoidance Control of a Non-Holonomic Mobile Manipulator for Intuitive Hand Guidance. *Robotics* **2019**, *8*, 14, doi:10.3390/robotics8010014.
15. Gan, Y.; Duan, J.; Chen, M.; Dai, X. Multi-Robot Trajectory Planning and Position/Force Coordination Control in Complex Welding Tasks. *Appl. Sci.* **2019**, *9*, 924, doi:10.3390/app9050924.
16. Jlassi, S.; Tliba, S.; Chitour, Y. An Online Trajectory generator-Based Impedance control for co-manipulation tasks. In Proceedings of the 2014 IEEE Haptics Symposium (HAPTICS), Houston, TX, USA, 23–26 February 2014; pp. 391–396, doi:10.1109/HAPTICS.2014.6775487.
17. Raiola, G.; Lamy, X.; Stulp, F. Co-manipulation with multiple probabilistic virtual guides. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 7–13, doi:10.1109/IROS.2015.7353107.
18. 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.
19. Zhou, H.; Chou, W.; Tuo, W.; Rong, Y.; Xu, S. Mobile Manipulation Integrating Enhanced AMCL High-Precision Location and Dynamic Tracking Grasp. *Sensors* **2020**, *20*, 6697, doi:10.3390/s20226697.
20. Wang, C.; Zhang, Q.; Tian, Q.; Li, S.; Wang, X.; Lane, D.; Petillot, Y.; Wang, S. Learning Mobile Manipulation through Deep Reinforcement Learning. *Sensors* **2020**, *20*, 939, doi:10.3390/s20030939.
21. Iriondo, A.; Lazkano, E.; Susperregi, L.; Urain, J.; Fernandez, A.; Molina, J. Pick and Place Operations in Logistics Using a Mobile Manipulator Controlled with Deep Reinforcement Learning. *Appl. Sci.* **2019**, *9*, 348, doi:10.3390/app9020348.
22. Su, H.; Qi, W.; Yang, C.; Sandoval, J.; Ferrigno, G.; Momi, E.D. Deep Neural Network Approach in Robot Tool Dynamics Identification for Bilateral Teleoperation. *IEEE Robot. Autom. Lett.* **2020**, *5*, 2943–2949.
23. Sanders, T.L.; Wixon, T.; Schafer, K.E.; Chen, J.Y.; Hancock, P. The influence of modality and transparency on trust in human-robot interaction. In Proceedings of the 2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), San Antonio, TX, USA, 3–6 March 2014; pp. 156–159.
24. Wortham, R.H.; Theodorou, A. Robot transparency, trust and utility. *Connect. Sci.* **2017**, *29*, 242–248.
25. Lakhmani, S.G.; Wright, J.L.; Schwartz, M.R.; Barber, D. Exploring the Effect of Communication Patterns and Transparency on Performance in a Human-Robot Team. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*; SAGE Publications Sage CA: Los Angeles, CA, USA, 2019; Volume 63, pp. 160–164.
26. Weiss, A.; Huber, A.; Minichberger, J.; Ikeda, M. First Application of Robot Teaching in an Existing Industry 4.0 Environment: Does It Really Work? *Societies* **2016**, *6*, 20, doi:10.3390/soc6030020.
27. Smith, S. *Digital Signal Processing: A Practical Guide for Engineers and Scientists*; Demystifying Technology Series; Elsevier Science: Amsterdam, The Netherlands, 2013.
28. Gfrerrer, A. Geometry and kinematics of the Mecanum wheel. *Comput. Aided Geom. Des.* **2008**, *25*, 784–791.