



A collaborative robot for the factory of the future: BAZAR

Andrea Cherubini¹ · Robin Passama¹ · Benjamin Navarro¹ · Mohamed Sorour^{1,2} · Abdellah Khelloufi^{1,3,4} · Osama Mazhar¹ · Sonny Tarbouriech^{1,5} · Jihong Zhu¹ · Olivier Tempier¹ · André Crosnier¹ · Philippe Fraisse¹ · Sofiane Ramdani¹

Received: 10 August 2018 / Accepted: 21 April 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

This paper introduces BAZAR, a collaborative robot that integrates the most advanced sensing and actuating devices in a unique system designed for the Industry 4.0. We present BAZAR's three main features, which are all paramount in the factory of the future. These features are: *mobility* for navigating in dynamic environments, *interaction* for operating side-by-side with human workers, and *dual-arm manipulation* for transporting and assembling bulky objects.

Keywords Efficient · Flexible and modular production · Robotics · Smart assembly · Human-robot co-working · Real industrial world case studies · Digital manufacturing and assembly system · Machine learning

1 Introduction

The concept of *cobots*, i.e., robots collaborating with human workers in manufacturing assembly lines, dates back to the pioneer work [3]. These devices should reduce ergonomic concerns due to on-the-job physical and cognitive loading, while improving quality and productivity.

Although some automotive manufacturers are gradually introducing cobots in their production line, a key question persists: how should a collaborative robot be designed? The ultimate goal is to have the adaptability of humans merged with the high performance of robots in terms of precision, speed and payload [31].

In this paper, we present BAZAR, the Bimanual Agile Zany Anthropomorphic Robot, designed and developed by

the LIRMM researchers since 2016. BAZAR (shown in Fig. 1) is a dual-arm mobile manipulator that combines the latest sensing and actuation technology to empower Factories of the Future. We are using it in the context of European Project H2020 VERSATILE (2017–2020), to address the needs of our industrial partners PSA Peugeot Citroën, Airbus, and BIC. VERSATILE targets the main challenges of the Industry 4.0, namely the radical change of the manufacturing environment, which has quickly shifted towards a novel production paradigm with high volumes of products individually personalized.¹ In this context, production flexibility and real-time machine reconfigurability become paramount [7].

We will mainly deploy BAZAR for the automotive kitting pilot case proposed by PSA in VERSATILE. In this application (which is a typical example of smart logistics), the robot must transport specific car parts to a “kit” carried by an autonomous ground vehicle. This requires the integration of many features, namely manipulation, interaction with humans, navigation, and docking.

Although many dual-arm mobile manipulators have been recently developed for similar applications, in our opinion, BAZAR has various advantages over the existing robots. These are listed below.

- The high quality of actuators and sensors. We have chosen the best devices available on the market. This design choice, while making the cost of the

¹ LIRMM, Université de Montpellier, CNRS, Montpellier, France

² Present address: Lincoln Center for Autonomous Systems (L-CAS) School of Computer Science, University of Lincoln, Lincoln, UK

³ Present address: Center for Development of Advanced Technologies CDTA, Algiers, Algeria

⁴ Present address: Faculty of Electronics and Computer Science, USTHB, Algiers, Algeria

⁵ Present address: Tecnia Research and Innovation, Derio, Spain

¹<https://versatile-project.eu/>

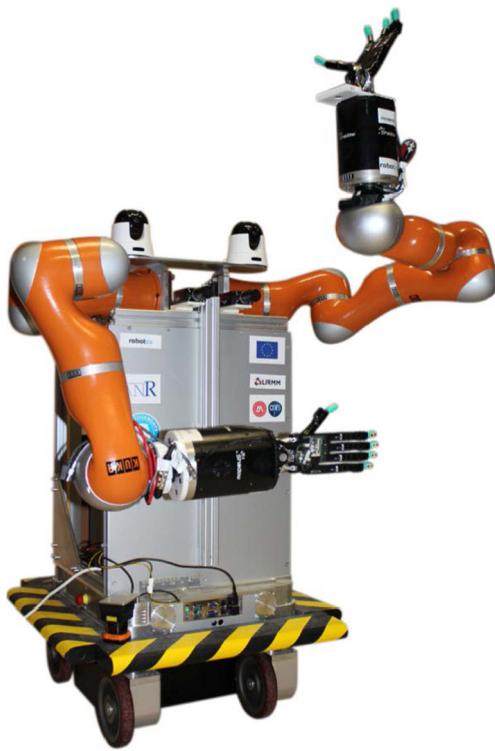


Fig. 1 The BAZAR (Bimannual Agile Zany Anthorpomorphic Robot) collaborative robot designed at LIRMM. The platform includes one Neobotix MPO 700 omni-directional mobile base with two Hokuyo UTM-30LX laser scanners, one Microsoft Kinect V2 RGBD camera (not show in this figure), two AVT Prosilica GT Gige/PoE RGB cameras, two DLINK DCS 5222L pan-tilt RGB cameras, and two KUKA LWR4+ arms. Each arm is equipped with one ATI Mini 45 force/torque sensor, one Shadow Dexterous Hand with SynTouch LLC Biotac tactile sensors and a Leapmotion vision sensor mounted on the hand wrist

robot increase substantially, is motivated by recent experiences of collaborative robots for the industry, which were driven by the opposite design choices. An emblematic example is Baxter from Rethink Robotics which—designed with cheap series elastic actuators, at the cost of precision and motion performance—caused Rethink Robotics to shut down in 2018.²

- The modular design of the hardware and software (which will be detailed in Section 2). All of BAZAR’s parts (arms, hands, sensors) can be easily removed or swapped with other ones (e.g., cheaper, see point above) if need be. First, this makes the prototyping quickly adaptable to the application and context. Second, such modular design can account for the—extremely short—lifetime of current day robotics devices, which evolve on a 2 to 3-year timescale. Modularity gives BAZAR

a crucial advantage over “closed” dual-arm mobile manipulators, such as Justin,³ PR2,⁴ or AILA [38].

- The use of omnidirectional wheels makes BAZAR capable of carrying higher payloads than its counterparts TUM-Rosie,⁵ MADAR [65], ARMAR⁶, or TOMM [23], all of which are equipped with swedish wheels. This is particularly relevant for industrial applications, such as the kitting pilot case mentioned above.

After a preliminary presentation of the *robot design* (Section 2), the paper focuses on the main features that we developed on BAZAR to make it capable of realizing such a complex task. Namely, these features are *mobility* for navigating in a dynamic and complex environment (Section 3), *interaction* for operating side-by-side with human workers (Section 4), and *dual-arm manipulation* for object transportation and assembly (Section 5).

All the results presented here are also illustrated in the videos of the IDH LIRMM channel.⁷

2 The BAZAR design

The irony of the name BAZAR, Persian for “marketplace,” lies in its meaning in the French urban dictionary, a “complicated grouping of different elements.” Indeed, in contrast with most existing mobile manipulators, which are completely designed by a single manufacturer, BAZAR combines the latest generation sensors and actuators—from various brands—within a unique robotic platform. These include laser scanners, 2D/3D cameras, force/torque sensors, and robotic arms and hands. In the rest of this section, we detail first the operational and then the control architecture of BAZAR.

2.1 Operational architecture

The BAZAR hardware is composed of robotic parts available on the market, and shown in Fig. 1. These are as follows:

- One *Neobotix MPO 700*, an omni-directional non-holonomic mobile robot base with four steerable wheels, for moving in the factory.
- Two *Hokuyo UTM-30LX* laser scanners, with 30 m and 270° scanning range, mounted on opposite corners

³[https://en.wikipedia.org/wiki/Justin_\(robot\)](https://en.wikipedia.org/wiki/Justin_(robot))

⁴<http://www.willowgarage.com/pages/pr2/overview>

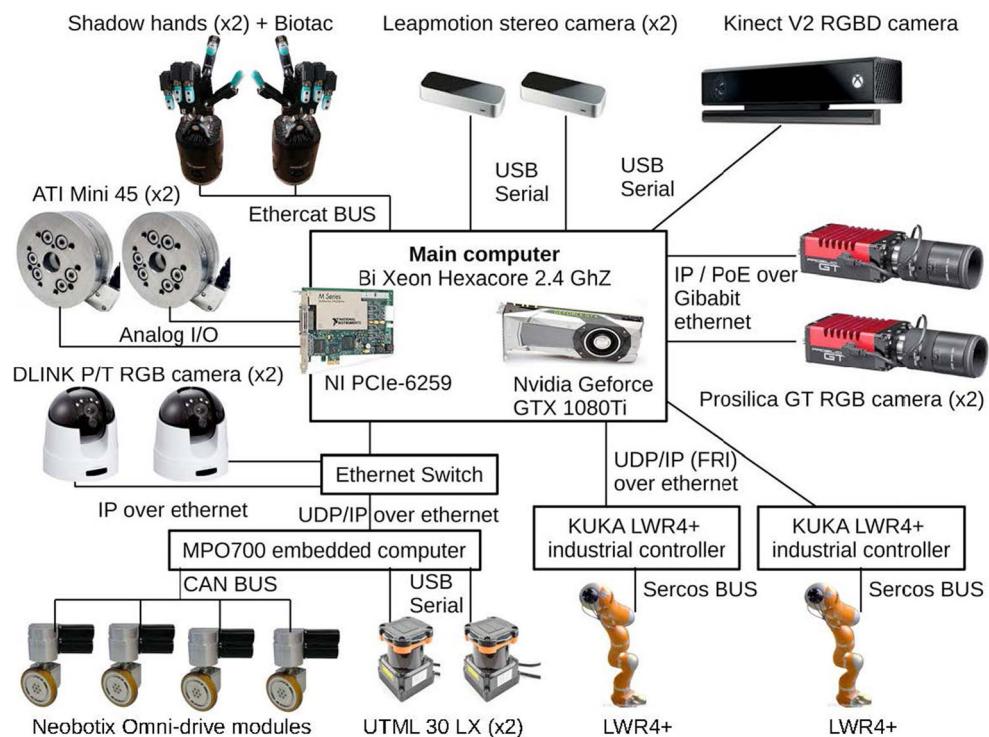
⁵<https://ias.cs.tum.edu/dokuwiki/robots/tum-rosie>

⁶<https://spectrum.ieee.org/automaton/robotics/industrial-robots/kit-armar6-humanoid>

⁷https://www.youtube.com/playlist?list=PLglynvGCAuFJe5Mct9L6B_woAcLRpaGgW

²<https://www.therobotreport.com/rethink-robotics-closes-its-doors/>

Fig. 2 The control architecture of BAZAR. The core component is BAZAR's embedded computer, with two Intel Xeon 2.4-Ghz hexacore processors with 32 Gb of RAM each and an Nvidia Geforce GTX 1080 Ti graphic card. This computer is connected to all the other devices (sensors and actuators) via the ethernet and USB links shown in the figure



of the mobile base, to get a complete 360° vision of the robot surroundings, for obstacle avoidance and environment mapping.

- One *Microsoft Kinect V2 RGBD* camera, mounted on BAZAR's torso⁸. This is used to detect humans in the environment.
- Two *AVT ProSilica GT Gige/PoE* RGB cameras, mounted on the robot torso as a stereo pair, used for far range object detection.
- Two *DLINK DCS 5222L* pan-tilt RGB cameras, also mounted on BAZAR's torso as a stereo pair. While having the advantage of being actuated, these have lower quality than the AVT cameras mentioned just above. Given their limitation for vision (bad image quality and slow frame rate) and control (slow movements and imprecise synchronization between images and servo position), we use them only for teleoperation, and not in the work presented here.
- Two *KUKA LWR4+* arms, each having 7 controllable joints (7 degrees of freedom) and a torque sensor per joint. These are used for object manipulation.
- Two (one per arm wrist) *ATI Mini 45* six axis force/torque sensors, mounted on each arm's end effector, to measure the external forces coming from physical interaction with environment. These are more precise than the joint torque sensors mentioned just above.

⁸The Microsoft Kinect V2 is not shown in Fig. 1.

- Two (one per arm wrist) *Shadow Dexterous Hands*, each with 19 degrees of freedom, mounted on the force sensors. They are used for grasping objects.
- Ten (one per hand fingertip) *SynTouch LLC Biotac tactile sensors*, for measuring contact points and grasping forces.
- Two (one per hand) *Leapmotion stereovision* sensors mounted on the hand wrists. These short range (0 to 0.5 m) sensors estimate the 3D position of objects that are near the hands' palms.

All these devices are mechanically assembled as shown in Fig. 1, with customized 3D-printed and machined hardware parts.

2.2 Control architecture

We implemented the control architecture (shown in Fig. 2) on an embedded computer with two Intel Xeon 2.4-Ghz hexacore processors with 32 Gb of RAM each. These support the Ubuntu 18.04 operating system and can provide up to 24 exploitable CPUs. An *Nvidia GeForce GTX 1080 Ti* is also embedded, and dedicated to GPU acceleration needed by deep learning algorithms.

This main computer is connected to all other sensors and subsystems using PCIe extension cards and motherboard default ports as detailed below.

- To communicate with the *MPO 700* embedded computer, we use an ethernet port of the main computer

motherboard, connected to a switch. A simple home-made protocol between the two, based on UDP/IP, gives access to the robot control modes (joint velocity or robot twist control) and to sensor data (joints, encoders, and laser scanners). The MPO700 computer support a ROS [57] architecture running on a Linux Ubuntu 10.04 operating system. It accesses the wheels' drive modules through a CAN bus and the *Hokuyo UTM-L30LX* laser scanners through USB serial links.

- We use a dual-port Ethernet extension card to communicate with the *AVT Prosilica GT* cameras. Communication with each camera is implemented using a dedicated Gigabit Ethernet link and cameras are powered using PoE (Power over Ethernet).
- To communicate with the *DLINK DCS 5222L* pan-tilt cameras, we rely on standard IP protocols for remote monitoring. These are supported by the same ethernet network used to control the MPO700.
- We use a dual-port Ethernet extension card to communicate with the control cabinets of the *KUKA LWR4+* arms, via a dedicated UDP/IP protocol called FRI⁹. The FRI protocol gives access to the robot control modes (position, impedance, and torque control) and sensor data (joint torques, joint position, etc.). The industrial controllers of *LWR4+* themselves support basic control functionalities and communicate with motors and sensors using a Sercos bus.
- We use a *National Instrument PCIe-6259* Data Acquisition card to acquire the analog signals from the two *ATI Mini45* force sensors.
- To communicate with both *Shadow hands*, we use an Ethercat bus. This gives access to the controlled torques of motors and to the sensors (encoders, Biotac) data.
- A USB3 extension card accesses—with individual USB serial—links *KinectV2 RGBD* camera and both *Leapmotion* stereo cameras.

As the reader can see, this architecture is quite heterogeneous. Its design is driven by the need for communication means specific to each vendor's subsystem. The direct consequence is the need to manage synchronization of these subsystems directly in the main computer software. This is still a challenging problem as one should concurrently consider the following specifications:

1. the correct execution of the control and perception pipelines imposes software real-time constraints,
2. the robot software should run on multiples CPUs, to have enough computing power to execute all algorithms on time,
3. the robot functioning mode should be dynamically reconfigurable, so that the global application can

quickly adapt to variations in the mission or environment.

Unfortunately, today, no generic software solution can deal with all these challenges. For instance, ROS [57] may manage communication between subsystems (running on different CPUs) but is not capable of dealing with global real-time *synchronization* constraints. Hence, for now, we have decided to manage synchronization “by hand.” We do this by managing creation, synchronization, data sharing, and periodic execution of threads with basic C++ APIs, to precisely control the execution of our software.

Regarding *modularity*, Ethernet and USB communication make the architecture easily adaptable to the different use cases, by simply connecting adequate subsystems. From the software perspective, this merely consists in using adequate drivers and related functionalities in the final application, i.e., the given use case. This way, we can implement minimal applications, to validate independently each subsystem. To achieve this in a reusable way, we integrate all software parts using a home-made standardization framework called *Knowbotics*. Our mid-term objective is to produce reusable software that is independent from any middleware, so that we can reuse it “on demand” within any context (e.g., OROCOS,¹⁰ ROS, ROS2¹¹).

Concerning *verification*, we have simulated all our algorithms using Matlab and V-REP¹² before experimenting on the real robot. While this is a crucial step in robotics, it has its limitations. In fact, simulations cannot account for the inaccuracy and incompleteness of real sensor data (e.g., vision or force measurements). The V-REP simulations are so far from realistic scenarios that we only use them to debug our software and to verify that it outputs adequate commands, but not to assess our algorithms' performance.

3 Mobility

In applications that require high mobility, industrials generally opt for steerable wheeled mobile robots (SWMR). These systems differ from classic wheeled ones by their capability of moving sideways. Their task space velocities are linear velocity \mathbf{v} (with components v_X and v_Y) and angular velocity ω (see Fig. 3). This mobility is obtained by employing fully steerable conventional wheels that have two active joints: one for steering and another for driving (noted respectively β_1 and ϕ_1 for wheel 1 in Fig. 3). SWMR are cheaper and have higher load

¹⁰<http://www.orocos.org/>

¹¹<https://github.com/ros2/ros2>

¹²<http://www.coppeliarobotics.com/>

⁹<https://cs.stanford.edu/people/tkr/fri/html/>

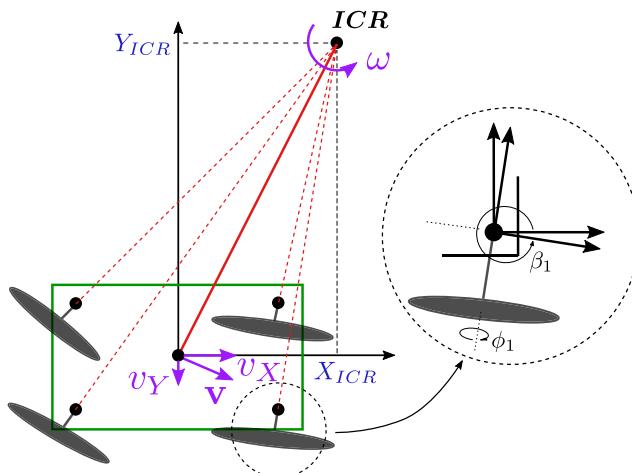


Fig. 3 Kinematic model of a steerable wheeled mobile robot with 4 wheels. The figure shows the task space velocities (linear velocity $\mathbf{v} = [v_X \ v_Y]^\top$ and angular velocity ω), the instantaneous center of rotation (ICR), and the steering and driving angles of wheel 1 (respectively β_1 and ϕ_1)

carrying capacities than other non-conventional wheels (e.g., Swedish or omni-directional). Hence, for BAZAR, we have chosen this solution that distinguishes it from similar mobile manipulators, such as MADAR [65] and TOMM [23].

Despite these clear advantages, SWMR present challenging control issues when operating in unpredictable dynamic environments. Indeed, in such environments, since the robot behavior cannot be preplanned due to unexpected events (e.g., moving obstacles or interacting humans), roboticists generally rely on reactive sensor-based methods. These methods use vision or distance feedback control to output the robot wheels velocities, which can vary unpredictably from one iteration to the next. This issue has not yet been treated in the literature, which currently focuses on “more classic” control problems specific to SWMR, such as:

1. proper steering coordination [10, 58, 61, 68],
2. avoidance of kinematic and representational singularities [19–21, 24, 30, 60, 64, 68],
3. fulfillment of steer joint performance limits [11, 18, 22, 50–53, 60, 63].

In the rest of this section, we present our solutions to two fundamental problems that must be addressed to deploy SWMR in factories. The first is a controller that can concurrently manage singularities and motion discontinuities. The second is a framework for visual navigation with simultaneous collision and occlusion avoidance. For further details on these two works, the interested reader is referred to [62] and [36], respectively.

3.1 Addressing kinematic singularities and motion discontinuities

We developed the kinematic model of BAZAR in Cartesian space coordinates, following the pioneer work of Betourne [5], Campion [8], Low [39], and Muir [42]. Although BAZAR has four wheels, the model is valid for any SWMR with three or more wheels. It consists of an inverse actuation kinematic model that gives the joint space velocities (steer $\dot{\beta}$ and drive $\dot{\phi}$) in function of the task space velocities (v_X , v_Y , ω), and of a forward kinematics model that estimates the robot odometry from the wheel encoders.

In a first work [63], we present a numeric treatment of SWMR kinematic singularities and a benchmark test trajectory for evaluating the performance of any SMWR controller with respect to these singularities. We also show that our approach outperforms the embedded controller of the BAZAR base (Neobotix-MPO700) in addressing the singularities.

In a second work [62], we design an ICR (instantaneous center of rotation, see Fig. 3)-based kinematic control framework that is:

1. robust with respect to trajectory discontinuity,
2. capable of handling kinematic singularities in the ICR space, and
3. compliant with the maximum steer joint limits in terms of velocity and acceleration (or seamlessly, jerk).

The framework consists of two decoupled kinematic controllers: a Cartesian velocity-based controller and an ICR-based one. The former controls the wheel driving speeds $\dot{\phi}$, by employing the Cartesian space kinematic model mentioned above. The latter controls the wheel steering rates $\dot{\beta}$, while respecting the maximum steer joint limits, by using optimization to locate the *next sample time* ICR coordinates. The benefit of using separate kinematic controllers for the drive and steering rates is that it is not necessary to map the 2D ICR coordinates (X_{ICR} , Y_{ICR}) to the 3D task space velocities (v_X , v_Y , ω). Thus, associated singularities and non-injectiveness are avoided. The 3D Cartesian space kinematic model is free from representational singularity, and the kinematic singularities are handled in the 2D ICR space controller.

In our framework, the task space velocities (v_X , v_Y , ω) are generated by a high level perception/navigation controller (that will be described in Section 3.2). This is then mapped to the 2D ICR space (X_{ICR} , Y_{ICR}), and the output-desired ICR motion is fed to a simple ICR velocity (proportional) controller, along with the current ICR coordinates. The output reference signal of the controller is then fed to an optimization algorithm that determines the *next sample time* ICR coordinates that will minimize the desired quadratic cost error, while respecting

the joint performance limits formulated as linear constraints. The corresponding steer joint reference signal β is then evaluated, differentiated, to obtain the command signal, and finally sent to the motor drivers. At the same time, a decoupled robot velocity controller is implemented. The initial output of this controller is projected in the null space of the *next sample time* robot configuration (represented by the kinematic constraint matrix), to obtain a feasible wheel rate command signal, $\dot{\phi}$. For more in depth details, the reader is referred to our work [62].

Once we are capable of managing discontinuous motion commands and kinematic singularities, we can have SMWR accomplish sensor-based navigation missions that have been traditionally deployed on non-holonomic robots, e.g., visual navigation [12] and lidar-based avoidance of mobile obstacles [13]. This will be the object of the next section.

3.2 Visual navigation with obstacle avoidance

A typical Industry 4.0 visual navigation application is kitting in automotive manufacturing, a pilot case proposed by PSA Peugeot Citroën in the context of the VERSATILE Project. The robot must transport car parts to a “kit” carried by a cart moving autonomously on the factory floor.

This task imposes several constraints related to the sensors (visibility, [6, 28, 55]) or to the environment (obstacles, [32, 37]). In this section, we address the problem of dealing with both collisions and occlusions caused by static and moving obstacles, during visual navigation of an omni-directional robot (here, BAZAR) equipped with fixed—limited field of view—on-board sensors.

Specifically, the navigation task consists in making the robot safely and autonomously navigate towards a static or moving *target*—the cart that carries the “kit”—in an unknown environment. Figure 4 gives an overview of this task with relevant variables. We detect the target T with the pair of *AVT Prosilica GT* cameras. Since these are fixed, their combined field of view (blue in the figure) is centered at the midpoint of the two optical centers. Concurrently, we build a local map of the obstacles (the *occupancy grid*, yellow in the figure) using the *Hokuyo* distance sensors.

Referring to Fig. 4, the task specifications are:

1. make the robot go towards the target T (blue triangle) until its position in the robot frame is T^* (cyan triangle),
2. orient the robot so that it points the visual sensors towards the target,
3. avoid collisions with the obstacles (gray), while realizing 1 and 2,
4. estimate the target pose (relative to the robot) at all times, even during sensor occlusions, in order to predict it in case of reappearance.

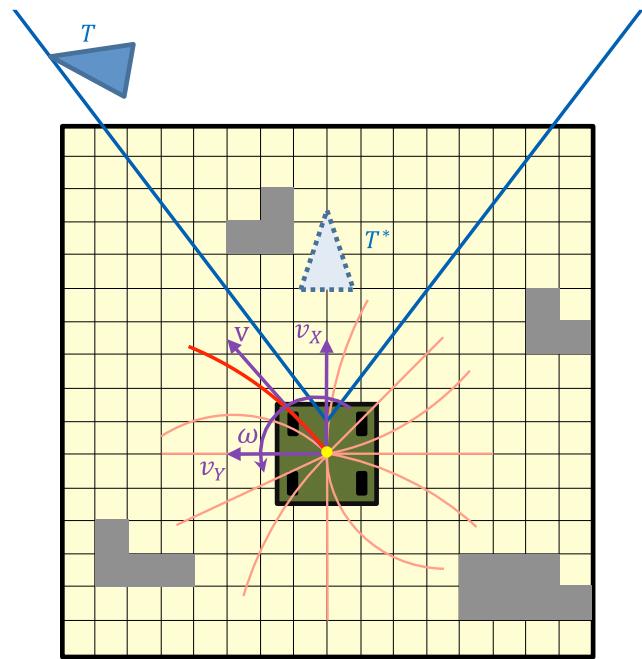


Fig. 4 Top view of the robot (green box with black wheels) with visual target T (blue). We show the occupancy grid (yellow), obstacles (gray), some omni-directional tentacles (red and orange), and the robot linear and angular velocities, respectively $\mathbf{v} = [v_X \ v_Y]^\top$ and ω (purple). In the figure, the robot is following the red tentacle: this tentacle is tangent to \mathbf{v} , with radius $\|\mathbf{v}\|/\omega$. We also represent the cameras field of view (blue) and the desired pose of the target T^* with respect to the robot

More formally, to fulfill specifications 1 and 2, the robot should move so that the visual target (that can be static or moving) is displaced in the robot frame, from the current pose T to a desired (final) constant pose T^* . This final pose is defined by the user depending on the application. In the VERSATILE kitting example, for BAZAR to place the car parts on the kit, the target is the kit and T^* will be some kit pose that is easily accessible by the robot hands.

In case dangerous obstacles (either static or moving) are present, the robot should circumnavigate them, while maintaining T in the field of view (specification 3). This is done by the collision avoidance framework that we designed in [35]. We use a set of drivable paths (tentacles) both for perception and motion execution. Each tentacle is a semicircle that starts in the robot center and is tangent to the robot linear velocity $\mathbf{v} = [v_X \ v_Y]^\top$. An example with 12 tentacles (5 linear and 7 circular) is shown in Fig. 4. A search algorithm selects the best tentacle that allows the robot to avoid the obstacles while keeping the target in the field of view. It may happen that none of the tentacles can guarantee both the field of view constraints and the robot safety, or that obstacles on the path provoke partial/total target occlusions. To overcome both problems (specification 4), we design a Kalman filter that estimates the target pose by using its previous location and the robot control inputs.

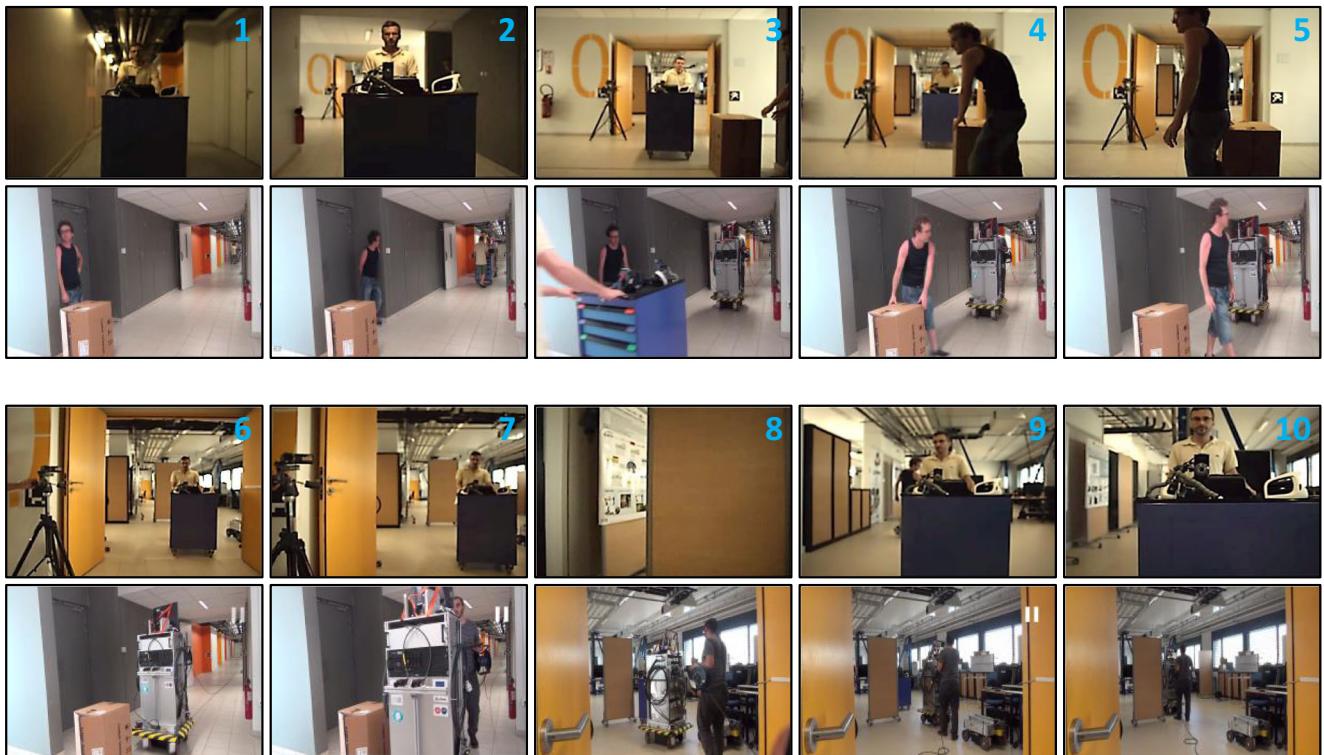


Fig. 5 Sequence of 10 image pairs during a visual navigation experiment. BAZAR (with arms and hands not mounted in this experiment) follows a blue moving cart (the target) in a cluttered environment. The environment is composed first of a corridor and then of a cluttered hall. For each of the 10 image pairs, we show on top the images

acquired by BAZAR's left AVT camera, and on the bottom an external view of the robot. The figure shows that with our approach, BAZAR can avoid obstacles, while simultaneously dealing with total target occlusion (images 5 and 8)

The effectiveness of our approach is validated through several experiments on BAZAR. One of them is shown in the 10 snapshots (pairs of images) of Fig. 5. For each pair, we show on top the images acquired by BAZAR's left camera, and on the bottom the external view of the robot. The purpose of this experiment is to assess the performance of our framework in a complex environment, including first a corridor and then a cluttered hall. BAZAR starts by following the blue cart (simulating the kit-carrying cart) that is moving along the corridor (snapshots 1–4). A first moving obstacle (human) crosses the robot path and occludes the target without affecting BAZAR's behavior (snapshot 5). The robot also succeeds in avoiding a second obstacle—while following the cart and keeping it in its field of view—until reaching the hall (snapshots 6 and 7). Then, BAZAR is in a difficult situation where an obstacle crosses its way, while the target has slightly changed direction (snapshot 8). This leads to a complete loss of target visibility. Once again, the robot manages to avoid the obstacle and to recover visibility of the cart, thanks to the target pose estimation module. Finally, BAZAR reaches the desired pose with respect to the target (snapshots 9 and 10), after having avoided all obstacles, while either keeping the target visibility or predicting its future location.

Despite these nice results, our framework has two limitations. First, vision is not generic yet (currently, the framework can only track a blue cart), but generalizing it is out of scope here and can nowadays be done with state-of-the-art deep learning algorithms such as YOLO.¹³ The second limitation concerns the initial conditions: we assume that the cart is visible at the beginning of navigation. If it is not the case, an exploratory phase can be added at initialization. This could be the object of future work.

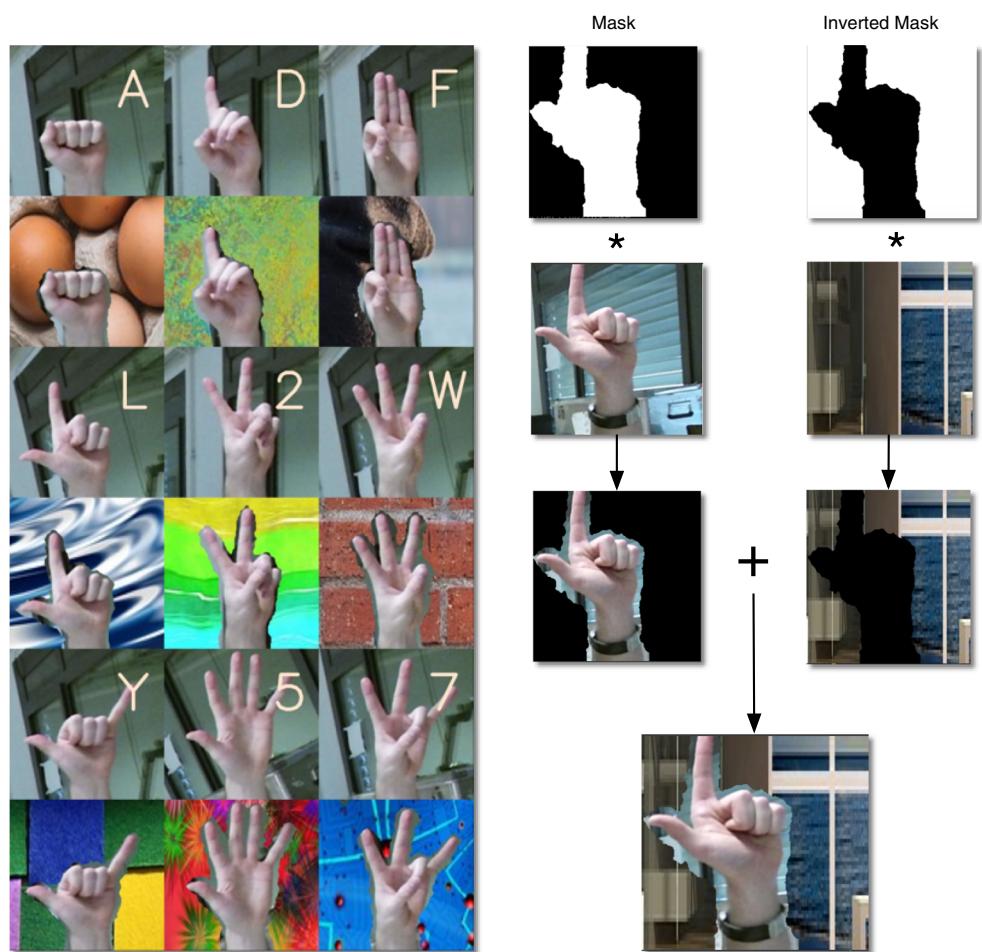
4 Interaction

In current-day factories, robots are designed and programmed to perform specialized tasks. Hence, it is difficult for an unskilled worker to reprogram the robot for a new task [49]. Traditional robot teaching methods are tedious, non-intuitive, and time consuming. Instead, gestures and touch are natural and intuitive ways to communicate/interact with the robot [48].

In the rest of this section, we explain how we enhanced BAZAR's interaction with humans. This is a crucial feature

¹³<https://pjreddie.com/darknet/yolo/>

Fig. 6 Left: the 9 gestures from the American Sign Language that we consider in our work. For each image, we show a real sample image (top) and the same image with substituted background (bottom). Right: background substitution process. From Kinect V2 depth map, we obtain a binary mask of the hand and its inverse. The first mask is multiplied by the original image to subtract the background, while the latter is multiplied by a random pattern image. Adding the resultants, we obtain a new hand image with substituted background



in the factory of the future, where non-expert users should have access to intuitive tools for easily controlling and re-programming robots on the fly. First, we present our framework for making BAZAR recognize human gestures using vision and deep learning. Then, we outline the features of our OpenPHRI (Open Physical Human Robot Interaction) software library; this is particularly useful for programming BAZAR, by using merely force and touch.

4.1 Vision-based Intention Recognition

In our recent work [40], we provide BAZAR with the capability of extracting—with its *Kinect V2*—human skeletons and hand gestures, to be used for intuitive robot control. We focus on the nine static hand gestures from the American Sign Language [4], shown in Fig. 6 (left) and consider all other as *None* gestures. Let us break the complete Vision-based Intention Recognition pipeline in the following steps:

1. data collection,
2. estimation of the human skeleton,

3. extraction of hand images,
4. image processing of hand images for training a convolutional neural network (CNN),
5. CNN training,
6. CNN validation.

In the following paragraphs, we detail each of these six steps.

Data collection In a campaign involving 10 volunteers (age 22–35, 8 males and 2 females), we have recorded—using BAZAR’s *Kinect V2*—the nine gestures. The recordings (RGB and depth images) are performed at distances of 1.5, 3, and 5 m from the *Kinect*. We have released online our dataset OpenSign,¹⁴ as a benchmark for the research community.

Estimation of the human 3D skeleton On each recorded RGB image, we extract the 2D human operator skeleton using the OpenPose library [9, 70], and enrich it with the third dimension, obtained from the *Kinect V2* depth image.

¹⁴<http://bit.do/OpenSign>

Table 1 Final results (normalized confusion matrix computed on the test set) of visual gesture recognition accuracy for the nine gestures

		Predicted labels									
		A	F	60 D	L	7	5	2	W	Y	~
Actual labels	A	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	F	0.005	0.995	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	D	0.000	0.000	0.995	0.000	0.000	0.000	0.002	0.000	0.000	0.003
	L	0.000	0.000	0.002	0.995	0.000	0.000	0.000	0.000	0.002	0.000
	7	0.000	0.000	0.000	0.000	0.998	0.000	0.002	0.000	0.000	0.000
	5	0.000	0.000	0.000	0.000	0.000	0.997	0.000	0.003	0.000	0.000
	2	0.000	0.000	0.000	0.000	0.000	0.002	0.992	0.000	0.000	0.005
	W	0.000	0.000	0.000	0.000	0.007	0.000	0.006	0.981	0.000	0.003
	Y	0.008	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.992	0.000
	~	0.019	0.005	0.005	0.004	0.005	0.005	0.011	0.000	0.005	0.943

Extraction of hand images Based on the 3D skeleton information, we crop—on each RGB and depth image—a region of interest around each hand. Specifically, the size of the region of interest is large if the hand is near (i.e., if the hand depth is small).

Processing hand images To make the detector more robust, we enrich the dataset by processing the hand images. At this step, a crucial operation is *background substitution*, shown in Fig. 6 (right). This is done with 1100 images of random patterns and indoor architectures freely available on the internet.¹⁵ Along with background substitution, we randomly apply the following techniques: addition of Gaussian or “salt and pepper” noise, histogram equalization, channel shift, zoom, shearing, axes flip, and position shift. None of these, except for background substitution, is applied to the cross-validation and test images.

CNN training For gesture classification, we exploit the state-of-the-art Convolutional Neural Network Inception V3 [66] which is pre-trained on large image data from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [59]. The total number of images for training is 20,950, and we divide them with a ratio of 3:1:1 between train, cross-validation, and test images. To make Inception V3 classify 10 gestures (instead of the 1000 for which it is designed), we replace the last softmax activation layer with a new layer of 10 neurons. We perform training in three phases by unfreezing the last layer only, then respectively the top two and four inception blocks. This gradual unfreezing prevents damaging the pre-trained weights and thus over-fitting. Each training epoch takes approximately 130 s to pass and the network achieves validation accuracy of 99.12% at the 745th epoch, after 27 h of training.

¹⁵<https://pixabay.com/>

CNN validation The cross-validation phase determines the best performing weights. Then, we run the network on the—unseen—test set, to quantify its accuracy. The normalized confusion matrix in Table 1 shows the accuracy in recognizing each gesture. The table shows that the system performs very well, with accuracies over 98% for the 9 gestures.

These results have encouraged us to deploy the system on BAZAR (see Fig. 7), to command the robot actions with the hand gestures dictionary. A limitation of our framework is that it relies on Kinect for 3D skeleton estimation. Porting to a standard camera requires 3D reconstruction, which is a non-trivial task, currently explored within our group. Furthermore, in future work, we will improve the accuracy of the *None* gesture (currently, 94.3%) by adding transitional gestures to the dataset.

4.2 Safe physical human-robot collaboration

A fundamental requirement of human-robot interaction in the *Industry 4.0* is *safety*. Furthermore, whenever interaction results in collaboration, i.e., when both agents work together for a common goal, the robot has to be *easy to program*, even by non-expert users. These two concerns have motivated us to develop OpenPHRI, an open-source software library designed to *easily program* collaborative robotic applications, with a strong emphasis on *safety*. OpenPHRI is available online¹⁶ free of charge under the GNU LGPLv3 license.¹⁷ Although we have validated OpenPHRI on BAZAR, the library can be used to control any robot meant to collaborate with humans.

Before explaining the library, let us recall some robotics fundamentals. A robot task can be expressed via joint

¹⁶<https://github.com/BenjaminNavarro/OpenPHRI>

¹⁷<https://www.gnu.org/licenses/lgpl-3.0.en.html>

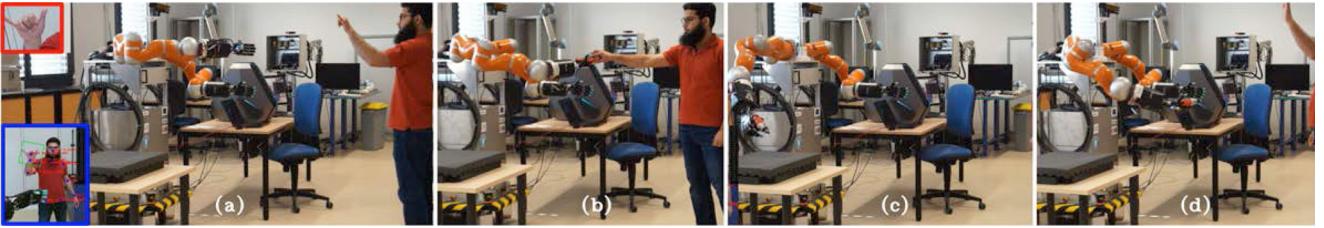


Fig. 7 Screen shots of the experiment that we performed to validate the robustness of our hand gesture detector. **a** The blue box shows the localization of human operator and the extraction of human skeleton. The red box shows the hand cropped image that is passed through the convolutional neural network for hand gesture detection. The human

velocities $\dot{\mathbf{q}}$ or via task space velocities $\dot{\mathbf{x}}$. The joint velocities are those of the robot actuators. Examples of task space velocities are for a fixed manipulator, the end-effector Cartesian linear and angular velocities with respect to a fixed frame, for a mobile base such as BAZAR's, the planar velocities, i.e., $\dot{\mathbf{x}} = [v_X, v_Y, \omega]^\top$ (see Section 3.1). A direct relation exists between these vectors through the Jacobian matrix \mathbf{J} that can be derived from the robot forward kinematic model:

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}. \quad (1)$$

OpenPHRI allows the user to provide compliance and other safety features in either the joint or task spaces, depending on the application. The framework implements a two-layer damping controller depicted in Fig. 8. The inputs can be real or virtual velocities and/or forces in joint and/or task space. The first layer (which embeds the damping controller) maps these inputs to desired joint space and task

operator in **(a)** gives a “Handover” command to the robot. In **b**, the operator hands over the tool to the robot. **c** The robot moves towards the table to drop the object. In **d**, the operator gives a “Stop” command to the robot to conclude the operation

space velocities: $\dot{\mathbf{q}}^*$ and $\dot{\mathbf{x}}^*$, respectively. The second layer accounts for the safety constraints (e.g., on velocity, power, kinetic energy) that can be arbitrarily defined by the user. This layer reduces the robot actuators' velocity $\dot{\mathbf{q}}_{\text{con}}$ if the desired velocities were to violate the constraints.

The following six equations sum up the OpenPHRI controller:

$$\dot{\mathbf{x}}^* = \mathbf{B}_t^{-1} \sum_{\mathbf{f}_i \in \mathcal{F}} \mathbf{f}_i + \sum_{\dot{\mathbf{x}}_i \in \mathcal{V}} \dot{\mathbf{x}}_i \quad (2)$$

$$\dot{\mathbf{q}}^* = \mathbf{B}_j^{-1} \sum_{\tau_i \in \Gamma} \tau_i + \sum_{\dot{\mathbf{q}}_i \in w} \dot{\mathbf{q}}_i \quad (3)$$

$$\dot{\mathbf{x}}_{\text{tot}} = \dot{\mathbf{x}}^* + \mathbf{J}\dot{\mathbf{q}}^* \quad (4)$$

$$\dot{\mathbf{q}}_{\text{tot}} = \mathbf{J}^\dagger \dot{\mathbf{x}}^* + \dot{\mathbf{q}}^* \quad (5)$$

$$\alpha = \min(1, \min(\mathcal{C})) \quad (6)$$

$$\dot{\mathbf{q}}_{\text{con}} = \alpha \dot{\mathbf{q}}_{\text{tot}}. \quad (7)$$

Damping control operates at both the task Eq. 2 and joint Eq. 3 levels, on the set of forces (\mathcal{F}, Γ) and on the set of velocities (\mathcal{V}, w), to combine multiple input sources. The user can regulate the behavior via damping gain matrices \mathbf{B}_t and \mathbf{B}_j . The resulting desired task and joint velocities ($\dot{\mathbf{x}}^*$ and $\dot{\mathbf{q}}^*$) are then combined using Eq. 1 to obtain the total velocities $\dot{\mathbf{x}}_{\text{tot}}$ and $\dot{\mathbf{q}}_{\text{tot}}$. These operations are shown in Eqs. 4 and 5, where \mathbf{J}^\dagger is the task Jacobian pseudo-inverse.¹⁸ The velocity scaling factor α in Eq. 6 is computed from the set of constraints \mathcal{C} , each constraint being a positive scalar value. The joint velocity $\dot{\mathbf{q}}_{\text{con}}$ finally sent to the robot motors is obtained with Eq. 7 by applying the velocity scaling factor α to the total joint velocity $\dot{\mathbf{q}}_{\text{tot}}$.

This approach, being quite simple, allows very fast computation of the control commands and is suitable for high-frequency real-time robot control. Besides, this simplicity facilitates the adoption of OpenPHRI by a large panel of people. More details about the method and experimental results are presented in [46]. The OpenPHRI library has been validated multiple times on BAZAR. Here,

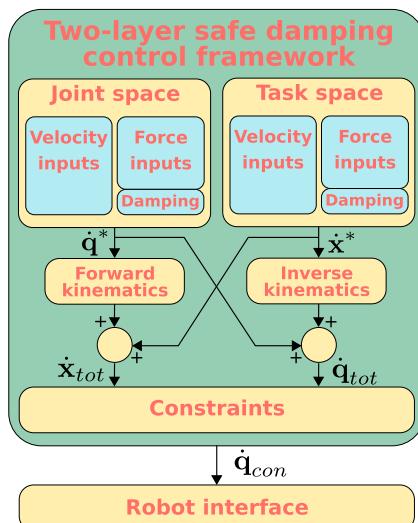


Fig. 8 Overview of the OpenPHRI framework. Variables $\dot{\mathbf{q}}^*$ and $\dot{\mathbf{x}}^*$ are the desired velocities output by the damping controllers respectively in the joint and task space. We obtain $\dot{\mathbf{x}}_{\text{tot}}$ and $\dot{\mathbf{q}}_{\text{tot}}$ by combining $\dot{\mathbf{q}}^*$ and $\dot{\mathbf{x}}^*$ respectively in the task and joint spaces. Finally, $\dot{\mathbf{q}}_{\text{con}}$ is the joint velocity sent to the robot motors after having applied the constraints

¹⁸Near singular configurations this is computed via damped least squares [17].

we outline two experiments, detailed respectively in [44] and [45].

The first (depicted on the left in Fig. 9) is a collaborative screwing experiment. In this scenario, BAZAR carries and actuates an electric screwdriver while complying with the external forces applied by the operator to reach the screwing locations. The arm motion is constrained, in order to fulfill the ISO10218 standard [1] and to keep the operator safe. Moreover, while the tool is being grasped, contact forces at the fingertip are regulated using the measures provided by the *Biotac Tactile Sensors* to ensure a firm grasp and avoid tool slippage. Since the thumb's tactile sensor is not in contact with the tool, the operator can use it as a button to shift between the different task states (i.e., grasp the tool, actuate the tool, etc). This is a simple and intuitive human-machine interface.

In the second experiment (right of Fig. 9), we apply a *whole-body* approach, by extending the arm's workspace thanks to the mobile base. As above, BAZAR is compliant to the forces applied by the operator. Yet, the goal now is to have the arm moving and the mobile base still when working locally, and the opposite (fixed arm and moving mobile base) when reaching a distant location. This is a more human-like motion than the one obtained when all robot parts are moving together. The transition between the two states (one part fixed, the other moving) is determined by the arm configuration: the base is fixed while the arm is not near a singular configuration, has a minimum of manipulability, and is away from the torso. As soon as one of these constraints is broken, the velocities are smoothly transferred to the mobile base to continue the motion. The mobile base stops when the operator pushes the arm away from these constraints.

A limitation of OpenPHRI is that currently it is being used only in our lab. Yet, as with any open-source library, we aim at attracting more users to test the library's features and limitations.

5 Dual-arm manipulation

By integrating *two* manipulators, the potential of robotics for industrial applications is significantly increased. This type of structure allows the manipulation of large and heavy objects [15] as well as the achievement of difficult tasks that are not feasible with a single arm [27, 56]. In the rest of this section, we present our work in providing BAZAR with the capabilities for manipulating rigid (Section 5.1) and deformable (Section 5.2) objects.

5.1 Dual-arm manipulation of rigid objects

Dual-arm platforms have a large number of degrees of freedom (DOF). Based on the cooperative task space representation [16], numerous bimanual operations only require the specification of arms' relative motion, while the absolute location can remain free. In this case, the number of DOF of the robot greatly exceeds the number of DOF of the task, providing high redundancy. Previous works on dual-arm control have proposed to make use of redundancy to avoid obstacles [41], increase manipulability [26], or satisfy joint constraints [33, 54].

However, these methods do not guarantee that hard constraints (e.g., physical limits of the robot) are satisfied at all times. Moreover, standard inverse kinematic controllers minimize the Euclidean norm of joint velocities, while still actuating unnecessary joints. Therefore, to perform simple operations (e.g., screwing), these controllers often generate large displacements in the workspace (as depicted in Fig. 10a).

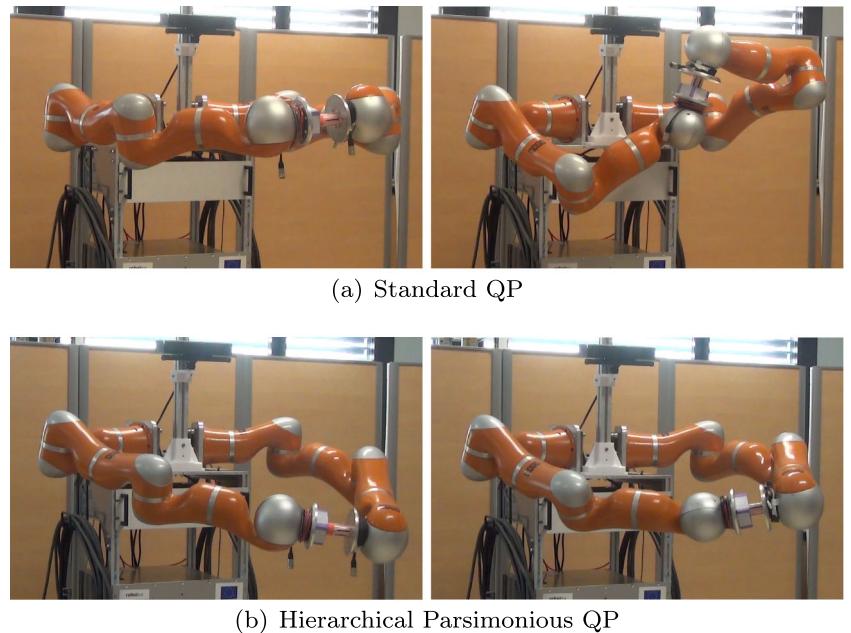
In [67], we propose a new approach to exploit redundancy during bimanual relative operations. Our strategy aims at generating a parsimonious solution, i.e., one that actuates the minimal number of joints needed to fulfill the task (see Fig. 10b). Hence, the overall robot motion is reduced and so is the risk of unintended contact



Fig. 9 Two examples of physical human-robot collaboration with BAZAR. *Left:* collaborative screwing. The robot helps the user in a screwing task. BAZAR's motion is compliant with the forces that the human applies on the hand wrist and fingertips. *Right:* intuitive

whole-body interaction with both BAZAR's arm and base. Depending on the force applied by the user, either the arm, the mobile base, or both moves

Fig. 10 Initial and final configurations reached during a screwing task with different task-solving approaches: **a** the standard QP results in the actuation of all the joints, with large motions in the workspace **(b)** conversely, with the hierarchical parsimonious QP, the robot performs the task by barely rotating its end-effectors, hence with more economic motion



in an unstructured industrial environment. Moreover, some studies have revealed that parsimonious motion provides more natural human-like movements and is therefore more predictable from the operator point of view [29, 71]. We base our approach on a hierarchy of tasks, implemented as a sequence of quadratic programs (QP) [34]. Thus, in addition to parsimonious task resolution, we define a secondary task with lower priority that is occasionally activated to move joints away from their physical limits when approaching them. Thanks to the QP architecture, hard constraints can also be handled to ensure the validity of the solution.

The objective is to find the joint velocities $\dot{\mathbf{q}}$ that guarantee a desired task space velocity $\dot{\mathbf{x}}$, according to Eq. 1. For dual-arm manipulation, $\dot{\mathbf{x}}$ is the relative motion between the end-effectors. For instance, performing the screwing operation shown in Fig. 10 corresponds to applying a rotational velocity around the axis to align the end-effectors. When the system is redundant, multiple joint configurations may realize the task, and a QP is used to obtain the desired $\dot{\mathbf{q}}$. This QP minimizes the Euclidean norm of Eq. 1 under a set of constraints to be satisfied at any time:

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}\|_2 \\ \text{s.t.} \quad & \mathbf{A}\dot{\mathbf{q}} = \mathbf{b}, \quad \mathbf{C}\dot{\mathbf{q}} \leq \mathbf{d}, \end{aligned} \quad (8)$$

with \mathbf{A} , \mathbf{C} , \mathbf{b} , and \mathbf{d} matrices and vectors predefined to set the constraints. These constraints may represent physical limits of the robot (joint position, velocity, ...) or express

safety considerations (collision avoidance). We can obtain a parsimonious solution of Eq. 1 by solving the problem:

$$\begin{aligned} \min_{\dot{\mathbf{q}}} \quad & \frac{1}{2} \|\mathbf{J}\dot{\mathbf{q}} - \dot{\mathbf{x}}\|_2^2 + \lambda \|\dot{\mathbf{q}}\|_1, \quad \lambda > 0 \\ \text{s.t.} \quad & \mathbf{A}\dot{\mathbf{q}} = \mathbf{b}, \quad \mathbf{C}\dot{\mathbf{q}} \leq \mathbf{d}. \end{aligned} \quad (9)$$

Setting $\lambda = 0^+$ will return the solution of Eq. 1 that has least l_1 -norm, corresponding to the most parsimonious solution [25].

In [67], we propose a *hierarchical parsimonious QP framework* that integrates (9) as the primary task and a joint limits avoidance strategy as lower priority task. We denote as QP1 the optimization process for solving the primary task and $\dot{\mathbf{q}}_1$ the corresponding solution:

$$\begin{aligned} \dot{\mathbf{q}}_1 \in \min_{\dot{\mathbf{q}}} \quad & \frac{1}{2} \|\mathbf{J}\dot{\mathbf{q}} - (\dot{\mathbf{x}}^* + K\mathbf{e})\|_2^2 + \lambda \|\dot{\mathbf{q}}\|_1, \quad \lambda = 0^+ \\ \text{s.t.} \quad & \underline{\dot{\mathbf{q}}} \leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{q}}}. \end{aligned} \quad (10)$$

Here, $\dot{\mathbf{x}}$ is the task velocity command (e.g., issued from trajectory generation) and $\underline{\dot{\mathbf{q}}}$ and $\bar{\dot{\mathbf{q}}}$ are lower and upper bounds for the joint velocities, respectively. A second QP (referred to as QP2) is implemented to consider joint limit avoidance as the secondary task. It generates an additional $\dot{\mathbf{q}}_2$ solution of:

$$\begin{aligned} \dot{\mathbf{q}}_2 \in \min_{\dot{\mathbf{q}}} \quad & \|\dot{\mathbf{q}} - \dot{\mathbf{q}}_r\|_2 \\ \text{s.t.} \quad & \underline{\dot{\mathbf{q}}} - \dot{\mathbf{q}}_1 \leq \dot{\mathbf{q}} \leq \bar{\dot{\mathbf{q}}} - \dot{\mathbf{q}}_1, \\ & \mathbf{J}\dot{\mathbf{q}} = \mathbf{0}. \end{aligned} \quad (11)$$

Here, $\dot{\mathbf{q}}_r$ is a joint limit repulsive vector (see [67] for details). To satisfy the primary task without degrading its

performance, we added the equality constraint $\mathbf{J}\dot{\mathbf{q}} = \mathbf{0}$. Hence, the solution space of QP2 is restricted to the null space of the primary task. Also, joint constraints have to be updated to take into account $\dot{\mathbf{q}}_1$. Finally, the joint velocity sent to the robot is:

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_2. \quad (12)$$

We have validated this approach in an experiment (shown in Fig. 10) where BAZAR's two arms perform a relative screwing motion to assemble two parts. The screwing operation that could be intuitively performed by a simple action generated a high occupancy of the workspace using the standard QP (Fig. 10a). Conversely, the parsimonious solution has resulted in local and economic movements, as demonstrated by Fig. 10b that shows similar intermediate and final configurations (except for the grippers' orientations). The secondary task has slightly modified the configurations to avoid limits for the right shoulder and for the wrists. During the whole operation, the standard QP returned solutions that actuate on average 13.4 joints vs the 6.3 actuated with the hierarchical parsimonious

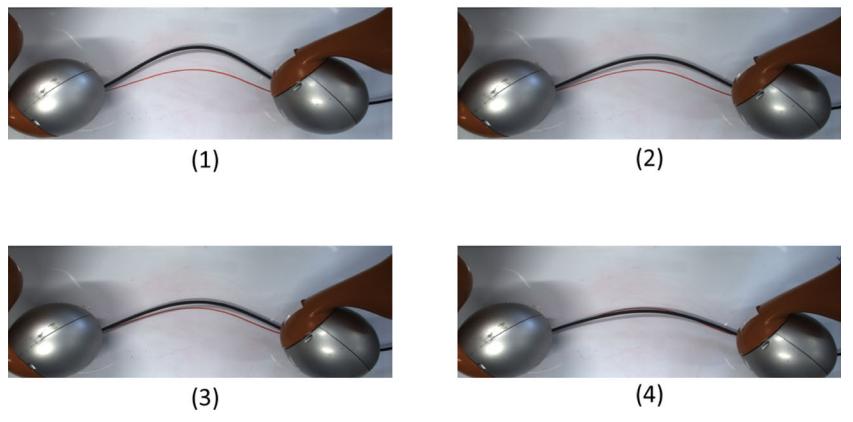
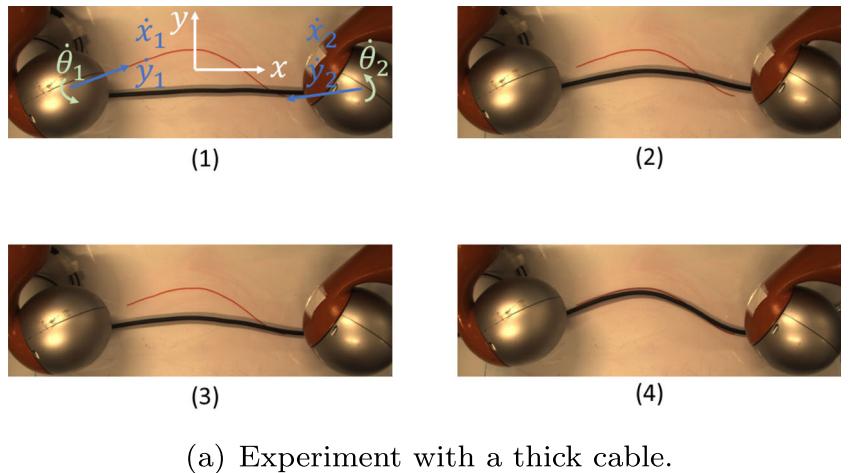
QP (12). Currently, our framework cannot manage collision avoidance (including self-collisions between the two arms). We are currently exploring this issue.

5.2 Dual-arm manipulation of deformable linear objects

Deformable objects are widely present in modern factories. The most common ones are cables/ropes, which are classified as deformable linear objects (DLOs). DLOs play an important role in many manufacturing processes. Cable displacement and insertion are typical tasks. A majority of these tasks involve shaping the DLO into a desired configuration. In this section, we present our controller for making BAZAR capable of modifying the shape of DLOs (here, cables) in the plane.

A model of linear objects considering bend, twist, and extensional deformations was developed in [69]. Nakagaki et al. extended the model in [69] and used it for cable insertion [43]. However, these approaches require a model to execute the manipulation task. This is not

Fig. 11 a, b Consecutive snapshots of BAZAR's end effectors while they manipulate a black cable. The goal is to deform the cables to the desired shape (red). To do so, we use visual feedback to control the planar velocities of the left ($\dot{x}_1, \dot{y}_1, \dot{\theta}_1$) and right ($\dot{x}_2, \dot{y}_2, \dot{\theta}_2$) end effector



practical in industrial settings, where cables often have different thicknesses and structures, so their deformation characteristics vary. Inspired by [47], we propose model-free dual-arm manipulation of DLOs using visual feedback. Further details are in [72].

We consider the DLO as a system with unknown dynamics that receives six inputs from BAZAR (see top left snapshot in Fig. 11a):

$$\mathbf{r} = [\dot{x}_1 \dot{y}_1 \dot{\theta}_1 \dot{x}_2 \dot{y}_2 \dot{\theta}_2]^T \in \mathbb{R}^6. \quad (13)$$

Each end-effector applies two translation velocities \dot{x} and \dot{y} and one angular velocity $\dot{\theta}$ in the manipulation plane. The DLO shape is observed by a static camera with optical axis perpendicular to the manipulation plane. We parameterize the DLO shape with an N order Fourier series, i.e., by the Fourier coefficient vector $\mathbf{s} \in \mathbb{R}^{4N+2}$. The goal is to design a feedback law for \mathbf{r} that drives the cable from its initial shape \mathbf{s}_0 to the desired shape \mathbf{s}^* , while estimating online the deformation model.

Small robot movements result in small variations of \mathbf{s} :

$$\delta\mathbf{s} = \mathbf{Q}\delta\mathbf{r}, \quad (14)$$

where $\mathbf{Q} \in \mathbb{R}^{(4N+2) \times 6}$ is the local deformation matrix, and $\delta\mathbf{r} = \mathbf{r}\delta t$ is the change in robot position, with δt being the sample time duration. Using vision data, we can estimate \mathbf{Q} from $\delta\mathbf{s}$. Then, to drive \mathbf{s} to \mathbf{s}^* , we apply the servo control law from [47]:

$$\delta\mathbf{r} = -\lambda(\hat{\mathbf{Q}}^\top \hat{\mathbf{Q}})^{-1} \hat{\mathbf{Q}}^\top \text{sat}(\mathbf{s} - \mathbf{s}^*), \quad (15)$$

with λ a feedback gain, $\hat{\mathbf{Q}}$ the estimate of \mathbf{Q} and $\text{sat}(\cdot)$ a vectorial saturation function.

We test our framework with different cable thicknesses, initial/desired shapes, and light conditions. Figure 11 shows the results of the experiments. The red curve indicates the desired shape. Using the proposed control framework, BAZAR successfully deforms the cables into the desired shape without knowledge of the model. Since the deformation matrix \mathbf{Q} directly maps the robot motion to the changes in the linear objects' shape, the approach requires no camera calibration. However, there are some limitations in our approach: the Fourier approximation yields an under-actuated control problem and the deformation model is only valid locally. Hence, there is no guarantee of global convergence. These limitations can be solved by path planning strategies which we are currently investigating. The goal is to plan the transitions between desired configurations so that they are “near enough” to guarantee convergence.

6 Conclusions

We have presented the design and features of BAZAR, a collaborative robot for the factory of the future that we have developed at LIRMM in Montpellier, France, since 2016. Aside the clear applicative purposes for the industry 4.0, BAZAR is an ideal platform for merging and extending state-of-the-art research results. Firstly, it can link *multiple senses* (force, vision, and distance) in the context of safe human-robot interaction and navigation. Second, it can be used for *whole-body control*, exploiting dual-arm kinematic solutions and extending them to the mobile base control. Last but not least, we have started to use BAZAR to explore the concurrent use of vision and force for *dual-arm manipulation of deformable objects*. Automatic manipulation of soft materials is a fundamental—yet open—research problem in robotics, despite its numerous applications (e.g., in the food industry). For this, we will profit from our previous works on dual-arm control [2] and on merging vision and force for manipulation [14].

Despite all these features, the algorithms that we presented in this paper (particularly BAZAR's kinematic controller described in Section 3.1) require tuning many parameters (e.g., gains, thresholds). Up to date, there is no reliable procedure to automate parameter tuning in complex robotic control tasks. In our work, tuning relies on trial-and-error and on the expertise of the researchers that learn by doing several experiments. Although this problem is being currently addressed using machine learning and non-linear optimization (e.g., particle swarms), such techniques do not easily generalize. Despite the nice results obtained in the field, the robotics community is aware that these methods work well for the situations and tasks they have been trained on, but do not generalize when the tasks change and when priorities switch among tasks. Furthermore, they require extensive use of simulators (to avoid consuming the robot hardware, as in the Google AI experiments¹⁹ where 14 robots must work in parallel for hours to learn the parameters of a simple grasping task). Yet, simulators do not always reproduce perfectly the real system variability (sensor measurements, unpredictability of human-robot interaction, etc). Although this problem concerns the optimization controllers at large and is somehow beyond the narrow scope of this paper (which focuses on the design and applications of BAZAR), it is in our opinion one of the main open unsolved problems in robotics.

Funding information This work has received funding from the European Union Horizon 2020 Research and Innovation Programme as part of the project VERSATILE under grant agreement no. 731330.

¹⁹<https://ai.google/research/teams/brain/robotics/>

References

1. ISO 10218-1:2011 (2006) Robot for industrial environments - Safety requirements - Part 1 : Robot. Tech. rep., International Organization for Standardization, Geneva, Switzerland
2. Adorno BV, Fraisse P, Druon S (2010) Dual position control strategies using the cooperative dual task-space framework. In: IEEE/RSJ international conference on robots and intelligent systems
3. Akella P, Peshkin M, Colgate E, Wannasuphprasit W, Nagesh N, Wells J, Holland S, Pearson T, Peacock B (1999) Cobots for the automobile assembly line. In: IEEE international conference on robotics and automation
4. Battison R (1978) Lexical borrowing in american sign language. ERIC
5. Betourne A, Campion G (1996) Kinematic modelling of a class of omnidirectional mobile robots. In: IEEE international conference on robotics and automation, pp 3631–3636. <https://doi.org/10.1109/ROBOT.1996.509266>
6. Bhattacharya S, Murrieta-Cid R, Hutchinson S (2007) Optimal paths for landmark-based navigation by differential-drive robots with field-of-view constraints. *IEEE Trans on Robotics* 23(1):47–59
7. Bortolini M, Faccio M, Gamberi M, Pilati F (2017) Multi-objective assembly line balancing considering component picking and ergonomic risk. *Comput Ind Eng* 112:348–367
8. Campion G, Bastin G, Novel B (1996) Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Trans on Robotics and Automation* 12(1):47–62. <https://doi.org/10.1109/70.481750>
9. Cao Z, Simon T, Wei SE, Sheikh Y (2017) Realtime multi-person 2D pose estimation using part affinity fields. In: IEEE conference on computer vision and pattern recognition
10. Cariou C, Lenain R, Thuilot B, Berducat M (2009) Automatic guidance of a four-wheel-steering mobile robot for accurate field operations. *Journal of Field Robotics* 26(6-7):504–518. <https://doi.org/10.1002/rob.20282>
11. Chamberland S, Beaudry E, Clavien L, Kabanza F, Michaud F, Lauriay M (2010) Motion planning for an omnidirectional robot with steering constraints. In: IEEE/RSJ international conference on intelligent robots and systems, pp 4305–4310. <https://doi.org/10.1109/IROS.2010.5648987>
12. Cherubini A, Colafrancesco M, Oriolo G, Freda L, Chaumette F (2009) Comparing appearance-based controllers for nonholonomic navigation from a visual memory. In: IEEE ICRA 2009 workshop on safe navigation in open and dynamic environments: application to autonomous vehicles
13. Cherubini A, Grechanichenko B, Spindler F, Chaumette F (2013) Avoiding moving obstacles during visual navigation. In: IEEE international conference on robotics and automation
14. Cherubini A, Passama R, Fraisse P, Crosnier A (2015) A unified multimodal control framework for human-robot interaction. *Robot Auton Syst* 70:106–115
15. Chiacchio P, Chiaverini S (1998) 1. In: Complex robotic systems, Springer, Berlin
16. Chiacchio P, Chiaverini S, Siciliano B (1996) Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. *J Dyn Syst Meas Control* 118(4):691–697
17. Chiaverini S, Egeland O, Kanestrom RK (1991) Achieving user-defined accuracy with damped least-squares inverse kinematics. In: 5th international conference advanced robotics, vol 1, pp 672–677. <https://doi.org/10.1109/ICAR.1991.240676>
18. Connette C, Hägele M, Verl A (2012) Singularity-free state-space representation for non-holonomic, omnidirectional undercarriages by means of coordinate switching. In: IEEE/RSJ international conference on intelligent robots and systems, pp 4959–4965. <https://doi.org/10.1109/IROS.2012.6386131>
19. Connette CP, Hofmeister S, Bubeck A, Hägele M, Verl A (2010) Model-predictive undercarriage control for a pseudo-omnidirectional, wheeled mobile robot. In: ISR 2010 (41st international symposium on robotics) and ROBOTIK 2010 (6th german conference on robotics), pp 1–6
20. Connette CP, Parritz C, Hagele M, Verl A (2009) Singularity avoidance for over-actuated, pseudo-omnidirectional, wheeled mobile robots. In: IEEE international conference on robotics and automation, pp 4124–4130. <https://doi.org/10.1109/ROBOT.2009.5152450>
21. Connette CP, Pott A, Hagele M, Verl A (2008) Control of an pseudo-omnidirectional, non-holonomic, mobile robot based on an icm representation in spherical coordinates. In: 2008 47th IEEE conference on decision and control, pp 4976–4983. <https://doi.org/10.1109/CDC.2008.4738958>
22. Connette CP, Pott A, Hägele M, Verl A (2010) Addressing input saturation and kinematic constraints of overactuated undercarriages by predictive potential fields. In: 2010 IEEE/RSJ international conference on intelligent robots and systems, pp 4775–4781. <https://doi.org/10.1109/IROS.2010.5652685>
23. Dean-Leon E, Pierce B, Bergner F, Mittendorfer P, Ramirez-Amaro K, Burger W, Cheng G (2017) TOMM: tactile omnidirectional mobile manipulator. In: IEEE international conference on robotics and automation
24. Dietrich A, Wimböck T, Albu-Schäffer A, Hirzinger G (2011) Singularity avoidance for nonholonomic, omnidirectional wheeled mobile platforms with variable footprint. In: IEEE international conference on robotics and automation, pp 6136–6142. <https://doi.org/10.1109/ICRA.2011.5979549>
25. Elad M, Bruckstein AM (2002) A generalized uncertainty principle and sparse representation in pairs of bases. *IEEE Trans Inf Theory* 48(9):2558–2567
26. Faroni M, Beschi M, Visioli A, Tosatti LM (2016) A global approach to manipulability optimisation for a dual-arm manipulator. In: IEEE international conference on emerging technologies and factory automation (ETFA)
27. Felipe J, Morales A (2014) A solution for the cap unscrewing task with a dual arm sensor-based system. In: IEEE-RAS international conference on humanoid robots
28. Folio D, Cadenat V (2008) A sensor-based controller able to treat total image loss and to guarantee non-collision during a vision-based navigation task. In: IEEE/RSJ international conference on robots and intelligent systems
29. Gauthier JP, Berret B, Jean F (2010) A biomechanical inactivation principle. *Proc Steklov Inst Math* 268(1):93–116
30. Giordano PR, Fuchs M, Albu-Schaffer A, Hirzinger G (2009) On the kinematic modeling and control of a mobile platform equipped with steering wheels and movable legs. In: IEEE international conference on robotics and automation, pp 4080–4087
31. Heyer C (2010) Human-robot interaction and future industrial robotics applications. In: IEEE/RSJ international conference on robots and intelligent systems, pp 4749–4754
32. Hoy M, Matveev AS, Savkin AV (2015) Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica* 33:463–497
33. Hu Y, Huang B, Yang GZ (2015) Task-priority redundancy resolution for co-operative control under task conflicts and joint constraints. In: IEEE/RSJ international conference on robots and intelligent systems
34. Kanoun O, Lamiaux F, Wieber PB (2011) Kinematic control of redundant manipulators: generalizing the task-priority framework to inequality task. *IEEE Trans on Robotics* 27(4):785–792

35. Khelloufi A, Achour N, Passama R, Cherubini A (2017) Tentacle-based moving obstacle avoidance for omnidirectional robots with visibility constraints. In: IEEE/RSJ international conference on robots and intelligent systems
36. Khelloufi A, Achour N, Passama R, Cherubini A (2018) Sensor-based navigation of omnidirectional wheeled robots dealing with both collisions and occlusions. *Robotica* (under revision). <https://hal.archives-ouvertes.fr/hal-01625946v4/document>
37. LaValle SM (2006) Planning algorithms. Cambridge University Press, Cambridge
38. Lemburg J, Fernandez G, Eich M, Mronga D, Kampmann P, Vogt A, Aggarwal A, Shi Y, Kirchner F (2011) AILA - design of an autonomous mobile dual-arm robot. In: IEEE/RSJ international conference on robots and intelligent systems
39. Low K, Leow Y (2005) Kinematic modeling, mobility analysis and design of wheeled mobile robots. *Ad Robotics* 19(1):73–99. <https://doi.org/10.1163/1568553053020241>
40. Mazhar O, Ramdani S, Navarro B, Passama R, Cherubini A (2018) Towards real-time physical human-robot interaction using skeleton information and hand gestures. In: IEEE/RSJ international conference on robots and intelligent systems
41. Mohri A, Yamamoto M, Hirano G (1996) Cooperative path planning for two manipulators. In: IEEE international conference on robotics and automation
42. Muir PF, Neuman CP (1987) Kinematic modeling of wheeled mobile robots. *J Robot Syst* 4(2):281–340. <https://doi.org/10.1002/rob.4620040209>
43. Nakagaki H, Kitagi K, Ogasawara T, Tsukune H (1996) Study of insertion task of a flexible wire into a hole by using visual tracking observed by stereo vision. In: IEEE international conference on robotics and automation, pp 3209–3214
44. Navarro B, Cherubini A, Fonte A, Passama R, Poisson G, Fraisse P (2016) An ISO10218-compliant adaptive damping controller for safe physical human-robot interaction. In: IEEE international conference on robotics and automation
45. Navarro B, Cherubini A, Fonte A, Poisson G, Fraisse P (2017) A framework for intuitive collaboration with a mobile manipulator. In: IEEE/RSJ international conference on robots and intelligent systems
46. Navarro B, Fonte A, Fraisse P, Poisson G (2018) Cherubini, a.: in pursuit of safety: an open-source library for physical human-robot interaction. *IEEE Robot Autom Mag* 25(2):39–50. <https://doi.org/10.1109/mra.2018.2810098>
47. Navarro-Alarcon D, Liu YH (2018) Fourier-based shape servoing: a new feedback method to actively deform soft objects into desired 2-D image contours. *IEEE Trans on Robotics* 34(1):272–279
48. Neto P, Pereira D, Pires JN, Moreira AP (2016) Gesture recognition for Human-Robot collaboration: a review. In: Proceedings of the 7th Swedish production symposium, pp 1–12
49. Neto P, Pires JN, Moreira AP (2010) High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition. *Ind Robot* 37:137–147
50. Oftadeh R, Aref MM, Ghabcheloo R, Mattila J (2013) Bounded-velocity motion control of four wheel steered mobile robots. In: 2013 IEEE/ASME international conference on advanced intelligent mechatronics, pp 255–260
51. Oftadeh R, Ghabcheloo R, Mattila J (2013) A novel time optimal path following controller with bounded velocities for mobile robots with independently steerable wheels. In: IEEE/RSJ international conference on intelligent robots and systems, pp 4845–4851
52. Oftadeh R, Ghabcheloo R, Mattila J (2014) Time optimal path following with bounded velocities and accelerations for mobile robots with independently steerable wheels. In: IEEE international conference on robotics and automation (ICRA), pp 2925–2931
53. Oftadeh R, Ghabcheloo R, Mattila J (2015) A time-optimal bounded velocity path-following controller for generic wheeled mobile robots. In: 2015 IEEE international conference on robotics and automation (ICRA), pp 676–683. <https://doi.org/10.1109/ICRA.2015.7139252>
54. Ortenzi D, Muthusamy R, Freddi A, Monteriù A, Kyrti V (2018) Dual-arm cooperative manipulation under joint limit constraints. *Robot Auton Syst* 99:110–120
55. Panagou D, Kumar V (2014) Cooperative visibility maintenance for leader-follower formations in obstacle environments. *IEEE Trans on Robotics* 30(4):831–844
56. Polverini MP, Zanchettin AM, Castello S, Rocco P (2016) Sensorless and constraint based peg-in-hole task execution with a dual-arm robot. In: IEEE international conference on robotics and automation
57. Quigley M, Conley K, Gerkey BP, Faust J, Foote T, Leibs J, Wheeler R, Ng AY (2009) ROS: an open-source robot operating system. In: ICRA workshop on open source software
58. Reister DB, Unseren MA (1993) Position and constraint force control of a vehicle with two or more steerable drive wheels. *IEEE Trans Robot Autom* 9(6):723–731. <https://doi.org/10.1109/70.265916>
59. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015) ImageNet large scale visual recognition challenge. *Int J Comput Vis (IJCV)* 115(3):211–252
60. Schwesinger U, Pradalier C, Siegwart R (2012) A novel approach for steering wheel synchronization with velocity/acceleration limits and mechanical constraints. In: IEEE/RSJ international conference on intelligent robots and systems, pp 5360–5366. <https://doi.org/10.1109/IROS.2012.6385644>
61. Selekwa MF, Nistler JR (2011) Path tracking control of four wheel independently steered ground robotic vehicles. In: 2011 50th IEEE conference on decision and control and european control conference, pp 6355–6360. <https://doi.org/10.1109/CDC.2011.6160677>
62. Sorour M, Cherubini A, Fraisse P, Passama R (2017) Motion discontinuity-robust controller for steerable mobile robots. *IEEE Robotics and Automation Letters* 2(2):452–459. <https://doi.org/10.1109/LRA.2016.2638466>
63. Sorour M, Cherubini A, Passama R, Fraisse P (2016) Kinematic modeling and singularity treatment of steerable wheeled mobile robots with joint acceleration limits. In: IEEE international conference on robotics and automation, pp 2110–2115. <https://doi.org/10.1109/ICRA.2016.7487360>
64. Stoger C, Muller A, Gattringer H (2015) Kinematic analysis and singularity robust path control of a non-holonomic mobile platform with several steerable driving wheels. In: 2015 IEEE/RSJ international conference on intelligent robots and systems, pp 4140–4145. <https://doi.org/10.1109/IROS.2015.7353962>
65. Suarez R, Palomo-Avellaneda L, Martinez J, Clos D, García N (2018) Development of a dexterous dual-arm omnidirectional mobile manipulator. In: IFAC symposium on robot control, SyRoCo
66. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2818–2826
67. Tarbouriech S, Navarro B, Fraisse P, Crosnier A, Cherubini A, Sallé D. (2018) Dual-arm relative tasks performance using sparse kinematic control. In: IEEE/RSJ international conference on robots and intelligent systems
68. Thuilot B, Novel B, Micaelli A (1996) Modeling and feedback control of mobile robots equipped with several steering wheels.

- IEEE Trans Rob Autom 12(3):375–390. <https://doi.org/10.1109/70.499820>
- 69. Wakamatsu H, Hirai S, Iwata K (1995) Modeling of linear objects considering bend, twist, and extensional deformations. In: IEEE international conference on robotics and automation
 - 70. Wei SE, Ramakrishna V, Kanade T, Sheikh Y (2016) Convolutional pose machines. In: IEEE conference on computer vision and pattern recognition
 - 71. Yazdani M (2012) Fast human movements and sparse optimal control policies. Ph.D. thesis, University of California, San Diego
 - 72. Zhu J, Navarro B, Fraisse P, Crosnier A, Cherubini A (2018) Dual-arm robotic manipulation of flexible cables. In: IEEE/RSJ international conference on robots and intelligent systems

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.