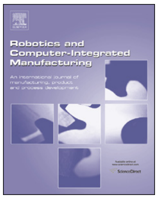
Robotics and Computer-Integrated Manufacturing 40 (2016) 1–13

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim



Full length Article

Collaborative manufacturing with physical human–robot interaction

Andrea Cherubini [a,*], Robin Passama [a], André Crosnier [a], Antoine Lasnier [b], Philippe Fraisse [a]

[a] *Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier LIRMM, Université de Montpellier - CNRS, 161 Rue Ada, 34392 Montpellier, France*

[b] *Department of Manufacturing Engineering, PSA Peugeot Citroën, Route de Gisy, 78140 Vélizy Villacoublay, France*

ARTICLE INFO

Article history:
Received 18 June 2015
Received in revised form
13 December 2015
Accepted 17 December 2015
Available online 1 January 2016

Keywords:
Cobots
Industrial robotics
Human–robot interaction
Reactive and sensor-based control

ABSTRACT

Although the concept of industrial *cobots* dates back to 1999, most present day hybrid human–machine assembly systems are merely weight compensators. Here, we present results on the development of a collaborative human–robot manufacturing cell for homokinetic joint assembly. The robot alternates active and passive behaviours during assembly, to lighten the burden on the operator in the first case, and to comply to his/her needs in the latter. Our approach can successfully manage direct physical contact between robot and human, and between robot and environment. Furthermore, it can be applied to standard position (and not torque) controlled robots, common in the industry. The approach is validated in a series of assembly experiments. The human workload is reduced, diminishing the risk of strain injuries. Besides, a complete risk analysis indicates that the proposed setup is compatible with the safety standards, and could be certified.

1. Introduction

The concept of *cobots*, i.e., robots collaborating with human workers in manufacturing assembly lines, dates back to the pioneer work [1]. In fact, cobots – designed for the assembly line worker – can reduce ergonomic concerns that arise due to on-the-job physical and cognitive loading, while improving safety, quality and productivity. This is a key issue, since according to statistics of the Occupational Safety and Health Department of the US Department of Labour,[1] more than 30% of European manufacturing workers are affected by lower back pain, leading to enormous social and economic costs. A thorough state-of-the-art on human–machine cooperation in manufacturing lines is provided in [2]. At the time of that survey (2009), the only hybrid assembly systems in manufacturing processes were weight compensators/balancers. However, the authors clearly point out the need for more advanced collaboration: although humans remain indispensable in many assembly operations, ergonomic tools assisting their duties are fundamental.

In this paper, we focus on a target application, proposed by PSA (Peugeot Citroën) in the frame of the French National Project ANR ICARO. The application is the assembly of an Rzeppa homokinetic joint, an operation that is currently done manually in the PSA line, causing muscular pain to the workers. In this work, we propose a novel, collaborative human–robot design, of this cell.

The main contributions of this work are outlined below.

- In contrast with most existing human–machine manufacturing applications, where collision avoidance is guaranteed by a minimum security distance [2], our framework successfully manages direct physical contact between robot and human, and between robot and environment.
- In our design, the robot alternates active and passive behaviours during assembly, to lighten the burden on the operator in the first case, and to comply to his/her needs in the latter.
- In contrast with most similar works, our approach can be applied to standard position (and not torque) controlled robots, common in the industry.

From the end user (PSA) viewpoint, two aspects are particularly noteworthy. First, since the operator load is reduced by approximately 60%, the proposed assembly cell can be reclassified in the PSA ergonomics scale. Second, a complete risk analysis by PSA indicates that the proposed setup is compatible with the safety standards, and could be certified.

The article is organized as follows. Section 2 summarizes the state-of-the-art in collaborative manufacturing, and highlights our contributions in the context of current, related research. In Section 3, we present the targeted application: collaborative assembly of a homokinetic joint. The proposed framework is outlined in Section 4. The framework components (nominal trajectory generation, admittance control, and safety monitoring) are then detailed in the

* Corresponding author.
E-mail addresses: andrea.cherubini@lirmm.fr (A. Cherubini), robin.passama@lirmm.fr (R. Passama), andre.crosnier@lirmm.fr (A. Crosnier), antoine.lasnier@mpsa.com (A. Lasnier), philippe.fraisse@lirmm.fr (P. Fraisse).

[1] www.osha.eu.int

http://dx.doi.org/10.1016/j.rcim.2015.12.007
0736-5845/© 2015 Elsevier Ltd. All rights reserved.

following sections (respectively, Sections 5–7). Experimental results are reported in Section 8, and finally summarized in the Conclusion.

2. Related work

This section summarizes the current state-of-the-art in collaborative manufacturing. We first review the more application-oriented research on human–machine cooperation (Section 2.1), and then the academic research on physical human–robot interaction (Section 2.2).

2.1. Research on human–machine cooperation in the industry

The authors of [3] provide a very rich overview of the emerging technologies in automotive assembly, including the supporting systems (mainly the information technologies). They show that mass customization requires high technological flexibility, and propose various designs to cope with this, by integrating both automated and human-based assembly. A similar perspective is that of the recent EU project ROBO-PARTNER [4], that aims at integrating assembly systems, and human capabilities. The main enablers, according to the authors, are: intuitive interfaces, safe strategies and equipment, proper methods for planning and execution, and the use of mobile robots, and of distributed computing. More recently, the U.S. Consortium for Automotive Research conducted a study on feasibility of fenceless robotic cells for automotive applications [5]. The study defines the levels of human–robot collaboration according to the cell complexity, to drive the probabilities of successful implementation. But as in the previously cited survey [2], the paper exposes the absence of high level human–robot collaboration, apart from "Intelligent Lift Assistants".

Although some automotive manufacturers are gradually introducing robots in their human production line [6,7], a crucial question persists: how should a collaborative robotic cell be designed? Various researchers have looked into this. Papakostas et al. [8] discuss the key features of cooperating robotic cells in automotive assembly, and provide simulated comparisons of two scenarios: a conventional welding robotic cell, and one with cooperating robots. The authors of [9] assess five alternative safety designs, covering both hardware and control design, of a human–robot collaboration prototype cell for cable harness assembly. In [10], a new cell production assembly system, with human–robot cooperation is developed. The system consists of three key technologies; parts feeding by double manipulators on a mobile base, production process information support for the operator, and safety management for cooperation between operator and robot. The main target of [11] is safety of the shared work cell, in the absence of physical fences between human and robot. Since safety options provided by basic infrared sensors are limited, the authors design a network architecture of these sensors, for tracking user positions, while avoiding collisions. The authors of [12] propose a method for optimizing task distribution among workers and robots. The method is validated, using an ABB Dual Arm Concept Robot, in a PLC Input/Output module assembly scenario.

2.2. Research on physical human–robot collaboration

Recent robotics research focuses on the study and characterization of physical human–robot interaction (pHRI [13,14]). The goal is to enable close collaboration between human and robot, in all service and industrial tasks, that require the adaptability of humans to be merged with the high performance of robots in terms of precision, speed and payload [15]. In this context, it becomes indispensable to define safety and dependability metrics [16–19]. These can contribute to the definition of standards, such as the recent ISO 10218-1:2011 "Safety requirements for industrial robots".[2]

In this line of research, many solutions for realizing safe collaborative tasks have been explored in recent years. Although these solutions have not yet been transferred to the industry, we hereby list some of the most relevant theoretical works. In [20], a deformation-tracking impedance control strategy is designed to enable robot interaction with environments of unknown geometrical and mechanical properties. For successful interaction with unknown environments and operators, the robot should behave in a human-like manner. This is the target of the research in [21,22]: a human-like learning controller is designed, to minimize motion error and effort, during interaction tasks. Simulations show that this controller is a good model of human–motor adaptation, even in the absence of direct force sensing. A robust controller for a collaborative robot in the automotive industry, is extended in [23], to manage not only the interaction between an industrial robot and a stiff environment, but also human–robot–environment and human–robot–human–environment interactions.

Other researchers have focused more on industrial applications. For example, an industrial robot controller, incorporating compliance of the joints with the environment, is presented in [24]. The desired pose of the tool center point is computed from the force error. Parallel control considers a reference trajectory while allowing feedforward in force controlled directions. Although the method is designed for industrial assembly tasks, it does not take into account the presence of a human in the loop. In contrast, Erden and colleagues [25–27] have thoroughly studied an industrial task that directly involves a human operator, i.e., manual welding. In [25], a physically interactive controller is developed for a manipulator robot arm: the human applies forces on the robot, to make it behave as he/she likes. Then, a manual welding assistant robot is presented in [26]: as the human controls the welding direction and speed, the robot suppresses involuntary vibrations (e.g., caused by novice welders). The results show a considerable improvement in the welders performance when they are assisted. Finally, [27] presents a study of end-point impedance measurement at human hand, with professional and novice welders. The results support the hypothesis that impedance measurements could be used as a skill level indicator, to differentiate the welding performance levels. Although the welding assistance application targeted by these works also falls in the shared workplace paradigm evoked in [2], it differs from the one treated here, since the robot motion is driven by the human worker. Instead, in our work, the robot is active and autonomous during various phases of the assembly cycle. For the same reason, robot programming by demonstration/teaching is also out of scope here.

Other works similar to ours, but targeting manually guided robot operation, are presented in [28,29]. In [28], an operator teaches tasks to a robotic manipulator, by manually guiding its end effector. For this, the authors design a virtual tool, whose dynamics the operator should feel when interacting with the robot. An admittance controller driven by the measurements of a force/torque sensor is designed to ensure the desired virtual dynamic behaviour. The second paper addresses the problem of controlling a robot arm, executing a cooperative task with a human, who guides the robot through direct physical interaction. This problem is tackled by allowing the end effector to comply according to an impedance control law [30] defined in the Cartesian space. Redundancy ensures the stability of the coupled human–robot system, through inertial decoupling at the end effector. However, in

[2] www.iso.org/iso/catalogue_detail.htm?csnumber=51330

contrast with our work, the robot is torque (not position) controlled, and an impedance (instead of admittance) controller is used.

As outlined in most of the cited works, an efficient cobot must interact easily with the worker, even if s/he is non-trained. To this end, many pHRI works rely on the physical contact (touch) between robot and operator [31]. More recently, to guarantee interaction even in the absence of direct contact, researchers have proposed the use of pointing gestures [32], as well as the integration of vision with force [33–35]. Also, in our work, interaction includes both vision and force. In fact, vision stops/triggers robot operation, in case of danger, while the operator contact forces are used to start and end assembly, and to deform the robot trajectory, for collision avoidance.

3. Application: collaborative assembly of a homokinetic joint

The application that is targeted in this work is the collaborative human–robot assembly of a car homokinetic joint. Homokinetic joints allow a drive shaft to transmit power to a second shaft through a variable angle, at constant rotational speed, while limiting friction and play. They are used both in front wheel drive cars, and in rear wheel drive cars with independent rear suspension.

Specifically, the homokinetic joint that we use in this work is the Rzeppa joint, which was invented in 1926 by Alfred Hans Rzeppa (1885–1965). The joint works in a manner similar to a bevel gear, with steel balls in place of teeth. The working principle and exploded diagram of this joint are shown in Fig. 1.

The joint consists of a spherical star-shaped *inner race* with six grooves in it, and a similar enveloping outer shell (*housing*). Each groove guides one steel spherical ball. The half shaft fits in the centre of the inner race, which itself nests inside a circular *cage*. The cage is spherical but with open ends, and with six openings around the perimeter, one for each ball. Finally, the housing is attached to the second shaft.

In this work, we propose a collaborative strategy for the assembly of the Rzeppa joint. In fact, after thorough discussions with the PSA ergonomists, and analysis of the most difficult cells in the PSA process, it appears that this use-case generates more Musculoskeletal Disorders (MSDs) than other assembly cells. This is what motivated the choice of this cell, as case study. In particular, we focus on the insertion of the six balls in the joint. This task is currently done manually, using an *insertion tool* similar to the one that is shown, along with all the joint components, in the left of Fig. 2. The ball insertion operation that is currently done in the PSA production process is outlined on the right of the same figure. The ball order is predefined, since, to limit the mechanical constraints, the first three balls must be inserted, so that there is an empty groove between any two of them.

Let us hereby describe the steps of the manual assembly operation. Prior to ball insertion (i.e., in the previous work cell), the inner race, cage, and housing have been assembled, and a first ball has been inserted. This ball links the cage and inner race, leaving some backlash, which will diminish as the other balls are placed. Hence, for simplicity, in the rest of this paragraph, the term cage will be used to refer to the linked grouping of cage and inner race. At first, the operator receives the partially assembled Rzeppa joint, and places it in an ad-hoc support. Then, by tapping on the cage with the insertion tool (see Fig. 2(1)), he slightly inclines it, until the second ball can be inserted (Fig. 2(2)). Since the backlash diminishes, for inserting the third and following balls, tapping is not sufficient anymore, and the operator must use the insertion tool to increase the lever arm needed to incline the cage (Fig. 2(3)). The insertion tool is designed so that its lower part fits precisely in the cage. The cage inclination (or *cage opening*), followed by ball insertion is repeated for the remaining balls (Figs. 2(4)– 2(8)). Once all six balls have been placed, the cage is closed (i.e., reset to the flat position, Fig. 2(9)), so that the Rzeppa joint can be sent to the following cell in the production chain.

An experienced operator executes this entire operation very quickly, to guarantee the required production rate of 70 joints per hour. However, the cage opening and closing steps can provoke MSDs. These injuries are due to the forces required for cage positioning, opening, and closing, and to the repetition of such actions. Ergonomists estimate that over an eight hour shift, the operator lifts a total weight of approximately 5 tons, and realizes 18 000 upper limb movements [36]. Therefore, this part of the operation should be automated to alleviate the operator. On the other hand, the action of ball insertion by itself requires very high precision and adaptability skills, which are not attainable by present-day industrial robots. Thus, the characteristics of the Rzeppa joint ball insertion work cell make it an ideal scenario for cobotics research.

In practice, one should automate the cell according to the following specifications:

(1) The human operator must position the five balls in the corresponding cage openings, with little effort and motion, so that the fatigue and chances of injury are minimized.
(2) The physical interaction between human and environment (specifically, between the human hand and the steel parts) must be controlled, to guarantee *safety*.
(3) The physical interaction between insertion tool and environment (specifically, the Rzeppa joint) must be controlled, to avoid blockage.
(4) The cobot velocity must guarantee safety, i.e., it must comply with the ISO safety standard cited in Section 2.2 (tool center point velocity limited to 0.25 ms^{-1}).

To fulfil the above requirements, we have redesigned the manufacturing cell, as shown in Fig. 3 (top). The lower part of the pre-assembled Rzeppa joint (composed of the housing, cage, inner race, and first ball), is held by a gripper placed on the end effector of a manipulator robot. The insertion tool, instead, is fixed to a support that is rigidly constrained to the robot base. In contrast with the manual insertion operation shown in Fig. 2, most of the required movements will be carried out by the robot, with the human intervening only to position the balls. The scenario is

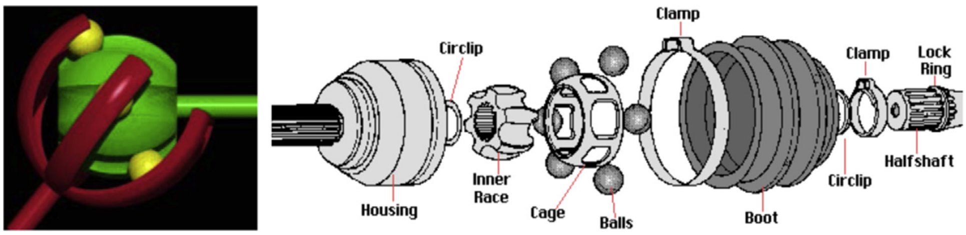


Fig. 1. Working principle (left) [https://en.wikipedia.org/wiki/Constant-velocity_joint] and exploded diagram (right) [www.aa1car.com] of the Rzeppa homokinetic joint.

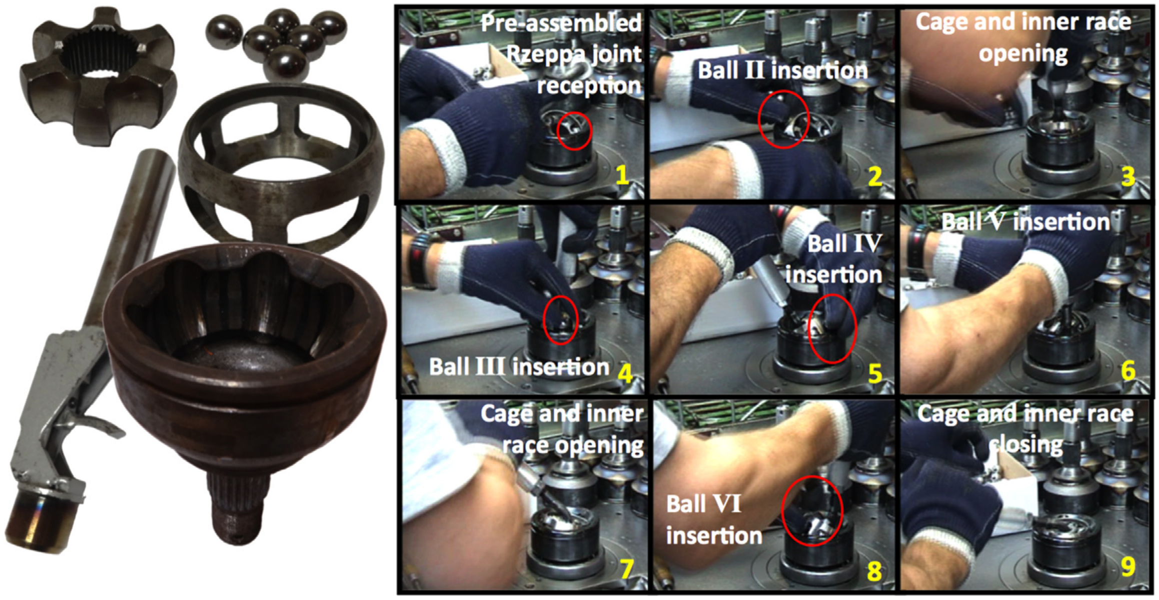

Pre-assembled Rzeppa joint reception
1
Ball II insertion
2
Cage and inner race opening
3
Ball III insertion
4
Ball IV insertion
5
Ball V insertion
6
Cage and inner race opening
7
Ball VI insertion
8
Cage and inner race closing
9

Fig. 2. Manual insertion. *Left*: concerned components (clockwise from top left: inner race, balls, cage, housing and insertion tool). *Right*: nine steps of the PSA manual insertion operation; ball 1 does not appear, since, along with the inner race, cage, and housing, it has been assembled before this operation.

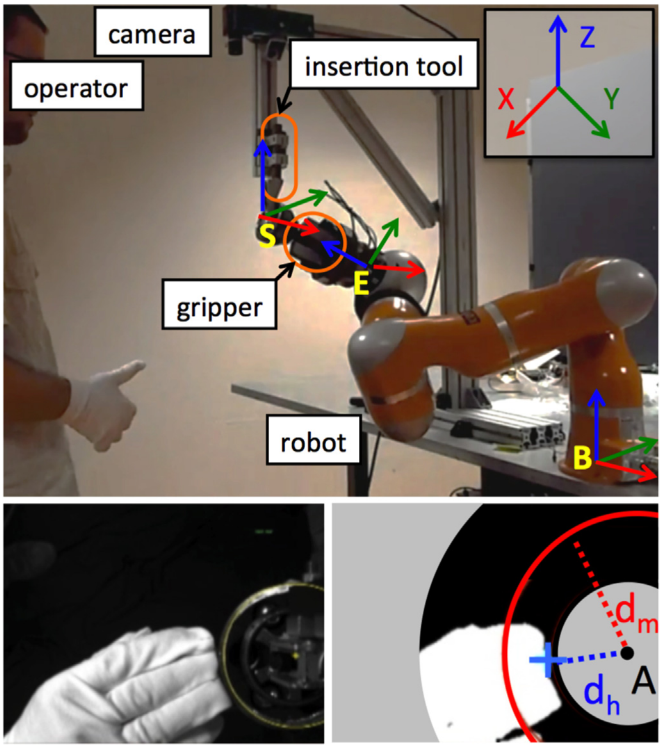


Fig. 3. *Top*: experimental setup for collaborative Rzeppa joint assembly with: insertion tool support (S), end effector (E), and robot base (B) reference frames. *Bottom left*: raw image from the camera. *Bottom right*: processed image with: locus of the tool axis in the image (A), distance between hand and tool (d_h), and minimum distance tolerated for safety (d_m). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

perfectly reversed, to make it *human-centered*: now the Rzeppa joint is displaced around the operator, instead of the opposite. To this end, we control the robot so that it opens/closes the cage – for ball insertion – by properly inclining it within the housing, and then pivots it around the vertical axis so that the operator is always facing the next desired opening. At the start (end) of the operation, the robot end effector raises (descends) to connect (disconnect) insertion tool and inner race. To realize all these motions (tool connection/disconnection, cage opening/closing and pivoting), the robot follows nominal, pre-taught trajectories. These nominal trajectories are deformed to comply with the external force/torques from the environment and operator, using an admittance controller [30]. To further enhance safety of operation, the robot is stopped in the presence of strong force/torques, and a fixed camera, mounted on the insertion tool support, monitors the ball insertion. The camera is used both to detect ball insertion, and to stop the robot if it may endanger the operator hands. Since there is no risk of clamping fingers in the back of the joint (solid steel of the housing, without openings), the camera only monitors the front of the cage. Images from the camera are shown at the bottom of Fig. 3 (raw image on the left, and processed image on the right).

In the following, we provide more details on the control framework developed for collaborative assembly of the Rzeppa joint.

4. Definitions and characteristics of the control framework

4.1. Definitions and assumptions

The reference frames used to describe the assembly task are shown in Fig. 3. In this, and in all other figures of the paper, the RGB convention is used: axes *X*, *Y* and *Z* are respectively red, green and blue. The frames used in this work are: the robot base ($\mathcal{F}_B$), insertion tool support ($\mathcal{F}_S$), and end effector ($\mathcal{F}_E$) frames, with origins respectively *B*, *S* and *E*. Reference frames $\mathcal{F}_B$ and $\mathcal{F}_S$ are fixed in the world, whereas $\mathcal{F}_E$ moves with the robot. The pose of $\mathcal{F}_S$ is determined via an estimation procedure explained in Section 5. The pose of a generic frame *A* in frame $\mathcal{F}_B$ is defined as: ${}^B\mathbf{p}_A = [{}^B\mathbf{t}_A\,{}^B\theta\mathbf{u}_A,]^\top \in \mathbb{SE}(3)$, with ${}^B\mathbf{t}_A$ the translation vector from *B* to *A*, and ${}^B\theta\mathbf{u}_A$ the angle/axis vector [37].

In this work, we consider a manipulator with $j \geq 6$ degrees of freedom (dof), and note $\mathbf{q} \in \mathbb{R}^j$ the robot joint values.

We also assume that it is possible to estimate (either through direct, or through joint torque measurement) the wrench (force ${}^E\mathbf{f}_E$ and torque ${}^E\mathbf{m}_E$) applied to the end effector, and expressed in the end effector frame: ${}^E\mathbf{h}_E = [{}^E\mathbf{f}_E,\ {}^E\mathbf{m}_E]^\top$. This information will be fed to an admittance controller [30], to adapt the robot motion, and avoid blockage between the Rzeppa inner race and the insertion tool.

4.2. Inverse kinematics controller

In our framework, an inverse kinematics controller is used to map end effector poses to the joint space. This is necessary for initially generating the nominal trajectory, and then for deforming it, according to the admittance controller. We hereby recall the classic inverse kinematics formulation according to [38].

The evolution of the end effector pose can be expressed as:

$$ {}^{B}\dot{\mathbf{p}}_E = \mathbf{J}\dot{\mathbf{q}}, \tag{1}$$

with $\mathbf{J} = \frac{\partial {}^{B}\mathbf{p}_E}{\partial \mathbf{q}}$ the *task Jacobian*, of size $6 \times j$, that is derived from the robot measured configuration $\hat{\mathbf{q}}$, and $\dot{\mathbf{q}}$ the robot joint velocities.

The end effector pose can be regulated to a desired value ${}^{B}\mathbf{p}_E^{*}$ and, if $j > 6$, redundancy [39] exists. Then, a secondary task can be realized, by exploiting the $j - 6$ extra degrees of freedom (since only 6 are required by the primary task, ${}^{B}\mathbf{p}_E^{*}$). The joint velocity $\dot{\mathbf{q}}$ for driving ${}^{B}\mathbf{p}_E$ to ${}^{B}\mathbf{p}_E^{*}$, is generated via:

$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}\mathbf{\Lambda}({}^{B}\mathbf{p}_E^{*} - {}^{B}\mathbf{p}_E) + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\nabla\mathbf{g}. \tag{2}$$

In the above equation:

- $\mathbf{J}^{\dagger}$ is the $j \times 6$ right pseudoinverse of $\mathbf{J}$, such that $\mathbf{J}\mathbf{J}^{\dagger}\mathbf{J} = \mathbf{J}$. We assume that $\mathbf{J}$ is full row rank during operation, so that this pseudoinverse can be calculated. This was the case throughout the experiments and is a common assumptions in inverse kinematics control [40].
- $\mathbf{\Lambda}$ is a positive definite 6-dimensional square diagonal matrix that determines the convergence rate of ${}^{B}\mathbf{p}_E$ to ${}^{B}\mathbf{p}_E^{*}$.
- The term $\nabla\mathbf{g} = \frac{\partial g}{\partial \mathbf{q}}$ is introduced to minimize the scalar cost function $g(\mathbf{q})$ to perform the secondary task.

It is well-known that system (1), controlled by (2), is globally asymptotically stable with respect to the pose regulation task. Indeed, plugging (2) into (1) yields:

$$ {}^{B}\dot{\mathbf{p}}_E = \mathbf{\Lambda}({}^{B}\mathbf{p}_E^{*} - {}^{B}\mathbf{p}_E), \tag{3}$$

and, since $\mathbf{\Lambda}$ is a positive definite diagonal matrix, ${}^{B}\mathbf{p}_E = {}^{B}\mathbf{p}_E^{*}$ is a stable equilibrium for the closed-loop system. Also, note that minimization of g has no effect on the task dynamics.

4.3. Control framework

Our framework for collaborative assembly is shown in Fig. 4. The framework is made up of two steps (shown respectively at the top and bottom of the figure): the *offline* nominal trajectory generation step, and the *online* controller for collaborative assembly. We hereby outline the framework, and will then detail its components in Sections 5–7.

A *nominal trajectory* $\mathcal{T}$, corresponding to the entire assembly operation, is input to the controller. This is defined by a series of N

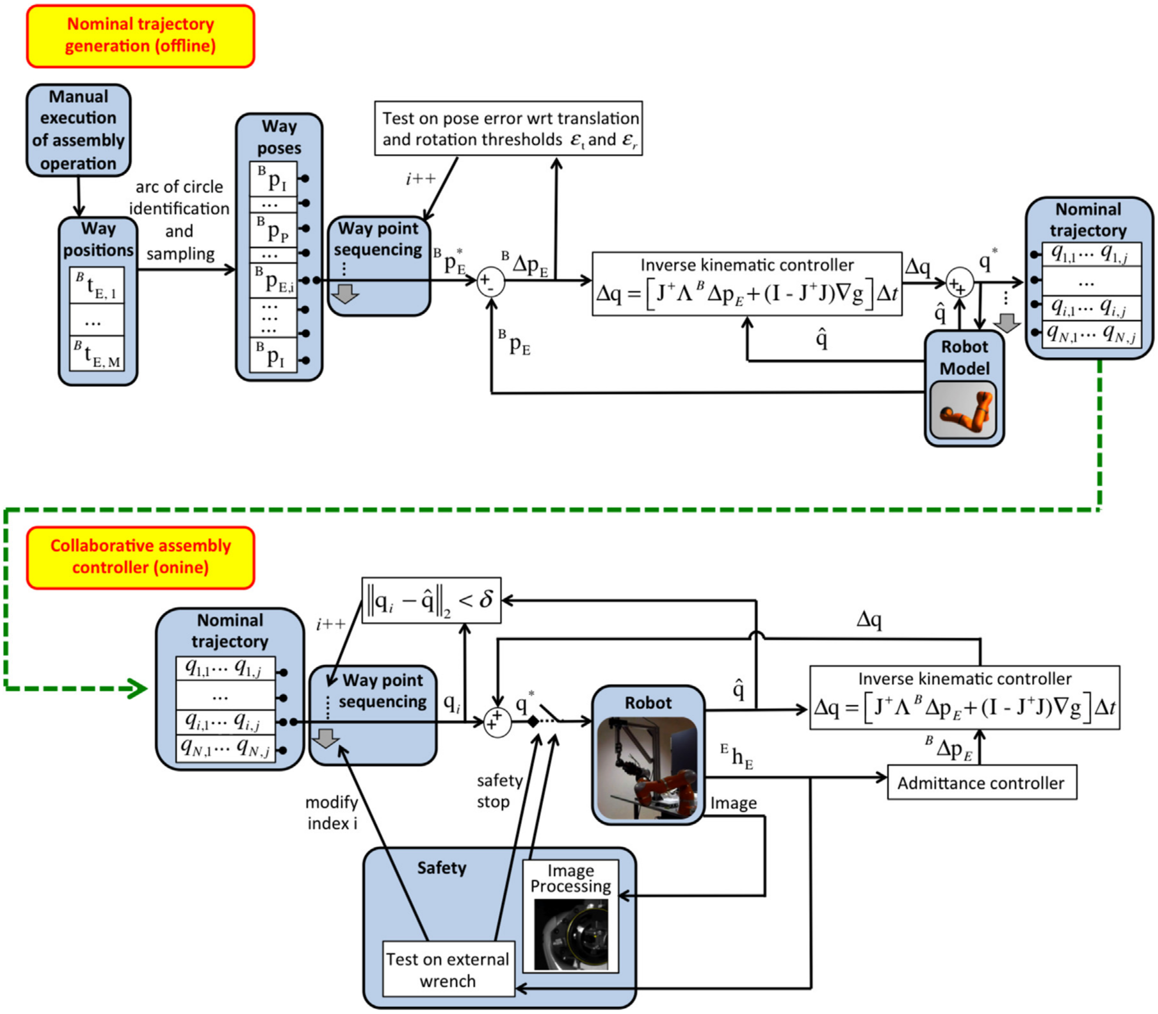


Fig. 4. Control framework for collaborative assembly. *Top*: nominal trajectory generation. *Bottom*: collaborative assembly controller.

way points in the joint space, that are to be followed by the robot:

$$\mathcal{T} = \{\mathbf{q}_1, \ldots, \mathbf{q}_N\}$$
$$\mathbf{q}_i = \left[q_{i,1} \ldots q_{i,j}\right]^\top \in \mathbb{R}^j. \tag{4}$$

In (4), $q_{i,1}, \ldots, q_{i,j}$ are the nominal joint angular values to be realized at iteration *i*. These are mapped from the operational space by the inverse kinematics controller, presented just above. Nominal trajectory (4) is identified offline through a procedure outlined at the top of Fig. 4, and detailed in Section 5.

Then, at run time, $\mathcal{T}$ is adapted online to account for the physical interaction with the environment and operator. To this end, an *admittance controller* (see Section 6) shifts the end effector desired pose by ${}^{B}\Delta\mathbf{p}_E = {}^{B}\mathbf{p}_E^* - {}^{B}\mathbf{p}_E$, according to the external wrench (force and torque) on the end effector, expressed in the support frame, that we denote with ${}^{S}\mathbf{h}_E$. This pose variation is mapped, again by the inverse kinematics controller, to a joint correction $\Delta\mathbf{q}$, then applied to the current waypoint of $\mathcal{T}$, to obtain the actual joint reference $\mathbf{q}^*$ for the robot motors. Index *i* is increased as soon as the reference trajectory way point $\mathbf{q}_i$ has been reached, i.e., when $\|\mathbf{q}_i - \hat{\mathbf{q}}\|_2 < \delta$, with $\hat{\mathbf{q}}$ the joint measure, and δ an arbitrary scalar threshold (see 'way point sequencing' block in the bottom of Fig. 4).

Prior to this, *safety* monitoring is applied. This relies on two verifications. The first consists in checking the magnitude of the external wrench, whereas the second uses processed image data to monitor the operator hand. The safety module is detailed in Section 7.

5. Nominal trajectory generation

This section presents the generation of the nominal trajectory $\mathcal{T}$ used by the control framework.

5.1. Characterization of the trajectory

Without loss of generality, we consider that the robot final joint value q_j is defined along the axis aligned with the end effector frame *z*-axis (positive q_j corresponds to a counter clockwise rotation around the *z*-axis). This assumption, which is consistent with the kinematics of most current day collaborative robots [41], facilitates the pivoting of the Rzeppa cage after each ball insertion. This pivoting will orient the joint, so that the groove where the next ball is to be inserted, will face the operator. The assembly operation can then be broken into five similar trajectories (one per ball), differing *only* by the orientation of the Rzeppa joint, i.e., of the end effector frame, with respect to the second-to-last joint frame. These five orientations (shown at the top of Fig. 5) have been designed to respect the ball insertion order shown on the top left of the same figure.

The collaborative Rzeppa joint assembly operation that is presented in this paper is then composed of the following steps (shown in the bottom of Fig. 5).

1. The end effector *E* connects the inner race to the insertion tool. For this, *E* tracks a straight line segment in the operational space.
2. The robot moves the Rzeppa joint to enable insertion of each ball – the following steps are cycled 5 times (once for each ball).
   (a) The joint is inclined so that the cage opens in front of the operator, and then the robot stays still until ball insertion (detected by the camera). For this, *E* tracks an arc of circle in the operational space (since the position of *S* and the distance between *E* and *S* are both constant).
   (b) The joint is inclined so that the cage is closed. Again, *E* tracks an arc of circle in the operational space.
   (c) The joint is brought back to the starting position of step 2, by having *E* track again an arc of circle in operational space.
   (d) The last joint q_j is actuated to pivot the Rzeppa joint, so that the next cage opening will face the operator. Specifically (see Fig. 5, top), after the insertion of each ball II…VI, joint q_j must rotate, relatively to the previous value by:

$$\Delta q_j^{\mathrm{II}} = -2\pi/3 \quad \Delta q_j^{\mathrm{III}} = -\pi/3$$
$$\Delta q_j^{\mathrm{IV}} = 2\pi/3 \quad \Delta q_j^{\mathrm{V}} = 2\pi/3 \quad \Delta q_j^{\mathrm{VI}} = -\pi/3. \tag{5}$$

   The last rotation shift is necessary to realign the robot end effector with the initial configuration, to prepare the next assembly operation.
3. The end effector disconnects the inner race from the insertion tool. For this, *E* tracks a straight line segment in the operational space.

Each of these steps is obtained by executing a trajectory, and all are concatenated to obtain the complete nominal trajectory $\mathcal{T}$ in (4). Defining these trajectories in the joint, rather than in the operational space, guarantees their feasibility (typically w.r.t. singularities and joint limits), and avoids errors due to inaccuracies in the encoder measurement and in the inverse kinematics computation. However, although technically defined in the joint space, the trajectories correspond to desired behaviours of the end effector in the operational space. The only exception is the trajectory in 2(d), which is best defined in the joint space, since it consists in controlling only the last joint value q_j to pivot the end effector ($q_{1,k} = q_{2,k} = \ldots = q_{N,k}$, $k = 1, \ldots, j-1$). For the other steps, instead, the task is best defined in the operational space.

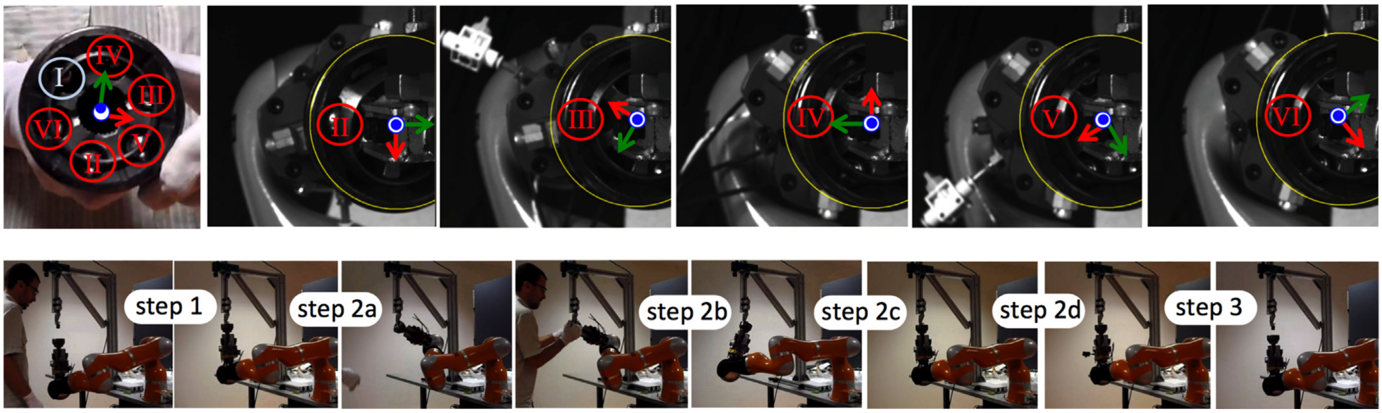


Fig. 5. *Top*: ball insertion order (left), and end effector frame configurations for inserting balls II…VI (right). *Bottom*: snapshots of the consecutive steps of the assembly operation (steps 2(a) to 2(d) are cycled 5 times, once for each ball).

5.2. Manual execution and identification in the operational space

The most delicate issue is the identification of the circular trajectory to be followed during steps 2(a)–2(c). This is crucial, since the Rzeppa joint is physically constrained to the insertion tool, during robot motion, and an inaccurate trajectory may lead to blockage. The trajectories for the other steps (1, 2(d), and 3) are derived from these circular trajectories. In fact: the trajectories for steps 1 and 3 are simple rectilinear connections to and from the arc of circle, without physical contact between the joint and the insertion tool, and the trajectory for 2(d) is constrained to the circle, with only orientation changes of the end effector. However, the equation of the circle in the robot base frame is related to the dimensions, configurations and poses of the insertion tool and of its support, which are very difficult to measure accurately. Thus, we have devised an identification procedure based on manually executing the nominal trajectory to the robot.

This procedure relies on the assumption that the robot is backdrivable (either via software or hardware). This is the case in most current day collaborative robots [41]. At first, the robot end effector is manually driven to the starting pose of step 2, noted P_{teach}. This corresponds to the insertion tool being connected to the inner race, with the Z axis of the end effector frame aligned with the insertion tool. Then, the operator moves the robot end effector back and forth a few times along the circular trajectory that is to be followed for ball insertion. The Cartesian positions of the end effector in the robot base frame are recorded throughout this operation:

$$\{{}^{B}\mathbf{t}_{E_1},\ \dots,\ {}^{B}\mathbf{t}_{E_M}\}$$
$$
{}^{B}\mathbf{t}_{E_i} = \left[{}^{B}X_{E_i}\,{}^{B}Y_{E_i}\,{}^{B}Z_{E_i}\right]^{\top} \in \mathbb{R}^3. \tag{6}
$$

This set of positions in the operational space, which is clearly noisy due to the shaky motion of the operator, is fed to an identification algorithm, coded in *Matlab*®.

The results of the identification are shown on the left of Fig. 6. First, the sphere (radius R and center position ${}^{B}\mathbf{t}_S$ in the base frame) that is closest to the taught trajectory, is found, by solving with `lsqnonlin` the nonlinear least-squares problem:

$$
\min_{R,{}^{B}\mathbf{t}_S} \sum_i \left[\left({}^{B}X_{E_i} - {}^{B}X_S\right)^2 + \left({}^{B}Y_{E_i} - {}^{B}Y_S\right)^2 + \left({}^{B}Z_{E_i} - {}^{B}Z_S\right)^2 - R^2 \right]^2. \tag{7}
$$

The center of this sphere is set as the origin of the support frame, S. Second, the plane passing through S, and closest to the taught points is found, by solving, with `lsqlin`, the constrained linear least-squares problem:

$$
\min_{a,b,c} \sum_i \left(a^{B}X_{E_i} + b^{B}Y_{E_i} + c^{B}Z_{E_i} - 1 \right)^2 \quad \text{subject to } a^{B}X_S + b^{B}Y_S + c^{B}Z_S = 1. \tag{8}
$$

Here, a, b and c are the parameters of the plane equation $a^{B}X + b^{B}Y + c^{B}Z = 1$. Third, we define the start point of step 2, noted P for pivoting cage point. Point P is obtained by projecting P_{teach} first on the sphere (7), and then on the plane (8). This way, it is possible to define the $\mathcal{F}_S$ frame axes: Z is parallel to line (SP) and pointing towards S, X is parallel to plane normal $[a\ b\ c]^{\top}$ and pointing to the right of the operator, and Y completes the frame. Then, naming Q the intersection between the sphere and the Y axis of $\mathcal{F}_S$, the equation of the arc of circle in $\mathcal{F}_B$ is:

$$
\begin{cases}
{}^{B}X(\psi) = {}^{B}X_S + {}^{S}X_P \cos\psi + {}^{S}X_Q \sin\psi \\
{}^{B}Y(\psi) = {}^{B}Y_S + {}^{S}Y_P \cos\psi + {}^{S}Y_Q \sin\psi \\
{}^{B}Z(\psi) = {}^{B}Z_S + {}^{S}Z_P \cos\psi + {}^{S}Z_Q \sin\psi,
\end{cases} \tag{9}
$$

with $\psi \in [\psi_m, \psi_M] \subset]-\pi, \pi]$. The two trajectory extrema ψ_m and ψ_M are manually tuned according to the setup. These correspond to points O (for open cage), and C (for closed cage). Finally, translating P away from the arc of circle, in the direction of decreasing ${}^{S}Z$, by the arbitrary length of the connecting trajectory, we derive the end effector initial pose, noted I. Having set a desired sampling interval on ψ, the four points I, P, O, and C, with equation (9), define the way poses of the end effector in the base frame for steps 1-3 for inserting *only* the first ball:

$$
\{{}^{B}\mathbf{p}_I, \dots, {}^{B}\mathbf{p}_P, \dots, {}^{B}\mathbf{p}_O, \dots, {}^{B}\mathbf{p}_P, \dots, {}^{B}\mathbf{p}_C, \dots, {}^{B}\mathbf{p}_P, \dots, {}^{B}\mathbf{p}_I\}. \tag{10}
$$

For each pose, the end effector orientation ${}^{B}\theta\mathbf{u}_E$ is calculated so that the end effector frame Z axis passes through S, and the Y axis is tangent to the circle, so that the ball 2 cage opening is in front of the operator (as in the second snapshot of Fig. 5). The result of the described identification procedure is shown in the left of Fig. 6, where we show: in red the recorded taught positions ${}^{B}\mathbf{t}_{E_i}$, in gray the insertion tool, in green the identified plane (with the top vertex corresponding to S), and in black the sampled trajectory (10).

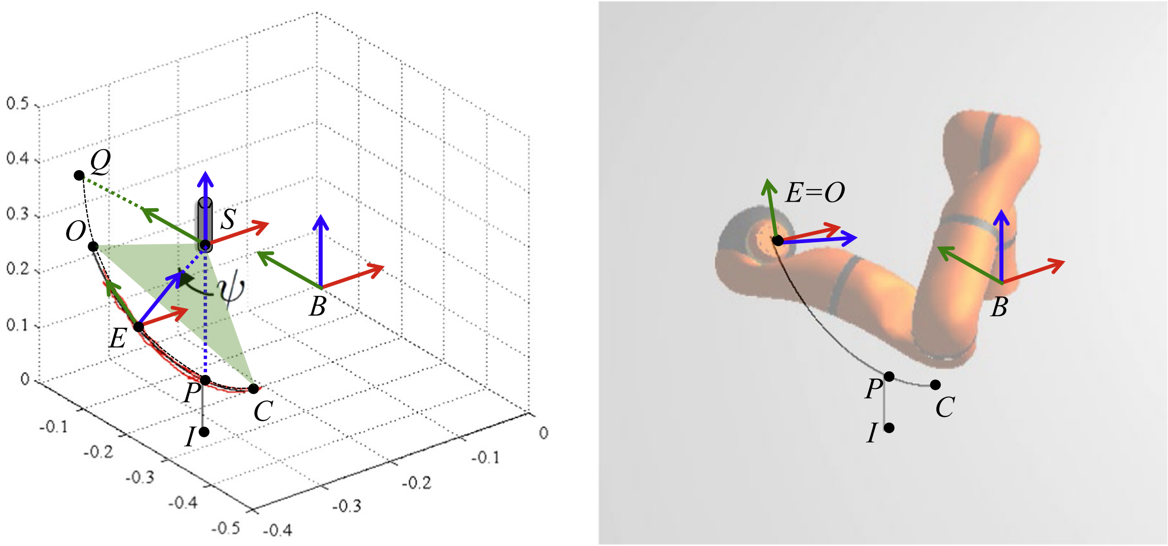


Fig. 6. *Left*: results of the trajectory identification with relevant way points I (initial), P (pivoting), O (open cage), and C (closed cage), and insertion tool (grey). *Right*: Simulations for replaying the identified path (black) using an inverse kinematics controller. In both figures, the base and end effector frames are also drawn. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

5.3. From the operational to the joint space trajectory

The following step consists in converting operational space trajectory (10) to the corresponding joint space trajectory, expressed as (4). For a general redundant robot, the inverse geometric model cannot be derived in closed form. Thus, we use the inverse kinematic pose controller (2) to track trajectory (10), as explained just below. First, we position the robot in the initial pose ${}^{B}\mathbf{p}_{I}$. Then, renaming ${}^{B}\mathbf{p}_{E_i}$ the ordered sample poses in (10), trajectory tracking consists in driving the end effector pose ${}^{B}\mathbf{p}_{E}$ to the next desired pose in the trajectory (${}^{B}\mathbf{p}^{*}_{E} = {}^{B}\mathbf{p}_{E_i}$), and incrementing i as soon as ${}^{B}\mathbf{p}^{*}_{E}$ is reached (i.e., as soon as $\|{}^{B}\mathbf{t}_{E} - {}^{B}\mathbf{t}^{*}_{E}\|_2 < \epsilon_t$ and $\|{}^{E}\mathbf{u}^{*}_{E} \sin {}^{E}\theta^{*}_{E}\|_2 < \epsilon_r$, with ϵ_t and ϵ_r two arbitrary scalar thresholds). The trajectory terminates when pose ${}^{B}\mathbf{p}^{*}_{E} = {}^{B}\mathbf{p}_{C}$ is reached, as in (10).

The joint values recorded while tracking (10), i.e.,

$$\{\mathbf{q}_I, \ldots, \mathbf{q}_P, \ldots, \mathbf{q}_O, \ldots, \mathbf{q}_P, \ldots, \mathbf{q}_C, \ldots, \mathbf{q}_P, \ldots, \mathbf{q}_I\}, \tag{11}$$

are then split in the following five primitive joint trajectories:

$$\begin{aligned}
&\mathcal{T}_1 = \{\mathbf{q}_I, \ldots, \mathbf{q}_P\},\\
&\mathcal{T}^{\mathrm{I}}_{2a} = \{\mathbf{q}_P, \ldots, \mathbf{q}_O\},\\
&\mathcal{T}^{\mathrm{I}}_{2b} = \{\mathbf{q}_O, \ldots, \mathbf{q}_P, \ldots, \mathbf{q}_C\},\\
&\mathcal{T}^{\mathrm{I}}_{2c} = \{\mathbf{q}_C, \ldots, \mathbf{q}_P\},\\
&\mathcal{T}_3 = \{\mathbf{q}_P, \ldots, \mathbf{q}_I\}.
\end{aligned} \tag{12}$$

The index of each trajectory coincides with the corresponding step, as listed at the beginning of this Section. For the five ball insertions, the last joint value q_j of the trajectories (12) must be shifted by the $\varDelta_q$ values in (5). This corresponds to applying the following iterative rule:

$$\begin{cases}
\mathcal{T}^{\mathrm{BALL}+1}_{\mathrm{STEP}} = \mathcal{T}^{\mathrm{BALL}}_{\mathrm{STEP}} & \\
\qquad\qquad + \{\mathbf{q}^{\mathrm{BALL}}, \ldots, \mathbf{q}^{\mathrm{BALL}}\} & \\
\mathcal{T}^{\mathrm{BALL}+1}_{2d} = \{\mathbf{0}, \ldots, \mathbf{q}^{\mathrm{BALL}+1}\} & \mathrm{STEP} = \{2a, 2b, 2c\}\\
\mathbf{q}^{\mathrm{BALL}} = [0\ldots 0\Delta q^{\mathrm{BALL}}_{j}] & \mathrm{BALL} = \{\mathrm{I}, \mathrm{II}, \mathrm{III}, \mathrm{IV}, \mathrm{V}\},
\end{cases} \tag{13}$$

with, in addition, $\Delta q^{\mathrm{I}}_{j}$ set to 0. In practice, the three trajectories of step 2 are identical for all five balls, except for the value of the last joint angle, which is shifted by a constant value at each new ball insertion. To guarantee smooth transitions between the trajectories, the Rzeppa joint is pivoted during step 2(d) with the linear interpolation indicated by the second equation in (13).

To summarize, the complete nominal trajectory for assembling an Rzeppa joint, is given by the following concatenation of trajectories:

$$\begin{aligned}
\mathcal{T} = \mathcal{T}_1 &\cup \mathcal{T}^{\mathrm{II}}_{2a} \cup \mathcal{T}^{\mathrm{II}}_{2b} \cup \mathcal{T}^{\mathrm{II}}_{2c} \cup \mathcal{T}^{\mathrm{II}}_{2d} \cup \mathcal{T}^{\mathrm{III}}_{2a} \cup \mathcal{T}^{\mathrm{III}}_{2b} \cup \mathcal{T}^{\mathrm{III}}_{2c} \cup \mathcal{T}^{\mathrm{III}}_{2d} \cup\\
&\cup \mathcal{T}^{\mathrm{IV}}_{2a} \cup \mathcal{T}^{\mathrm{IV}}_{2b} \cup \mathcal{T}^{\mathrm{IV}}_{2c} \cup \mathcal{T}^{\mathrm{IV}}_{2d} \cup \mathcal{T}^{\mathrm{V}}_{2a} \cup \mathcal{T}^{\mathrm{V}}_{2b} \cup \mathcal{T}^{\mathrm{V}}_{2c} \cup \mathcal{T}^{\mathrm{V}}_{2d} \cup\\
&\cup \mathcal{T}^{\mathrm{VI}}_{2a} \cup \mathcal{T}^{\mathrm{VI}}_{2b} \cup \mathcal{T}^{\mathrm{VI}}_{2c} \cup \mathcal{T}^{\mathrm{VI}}_{2d} \cup \mathcal{T}_3.
\end{aligned} \tag{14}$$

In the next section, we explain how this nominal trajectory $\mathcal{T}$ is used to control the robot, during the Rzeppa joint assembly operation, when the interaction forces with the environment and operator are also to be taken into account.

6. Admittance control

Admittance and impedance are reciprocal concepts [30]: while impedance control produces forces/torques in response to velocities, admittance control produces velocities in response to forces and torques, at a given interaction port. Here, we apply an admittance controller, to map the contact wrench to a desired deviation of the end effector motion in the base frame, ${}^{B}\Delta\mathbf{p}_{E}$, through an equivalent spring-damper system. This guarantees motion safety, since the end effector trajectory is 'naturally' deformed to avoid blockage. It also makes the robot compliant during step 1, so that the operator can manually guide the inner race in the insertion tool.

The contact between robot and environment is located at S, and the frame $\mathcal{F}_S$ is immobile during operation. For these reasons, the admittance controller will rely on the wrench applied on the end effector and expressed in the support frame, ${}^{S}\mathbf{h}_{E}$. It is straightforward to obtain this from ${}^{E}\mathbf{h}_{E}$, through:

$${}^{S}\mathbf{h}_{E} = {}^{S}\mathbf{W}_{E}{}^{E}\mathbf{h}_{E}, \tag{15}$$

with:

$${}^{S}\mathbf{W}_{E} = \begin{bmatrix} {}^{S}\mathbf{R}_{E} & \mathbf{0} \\ \left[{}^{S}\mathbf{t}_{E}\right]_{\times}{}^{S}\mathbf{R}_{E} & {}^{S}\mathbf{R}_{E} \end{bmatrix}, \tag{16}$$

and ${}^{S}\mathbf{R}_{E}$ the rotation matrix corresponding to angle/axis vector ${}^{S}\theta\mathbf{u}_{E}$. The relative pose of the effector in the support frame required in (16), can be derived at each iteration from the robot end effector pose ${}^{B}\mathbf{p}_{E}(\mathbf{q})$, and from the pose of the support in the base frame, ${}^{B}\mathbf{p}_{S}$ (computed as explained in 5).

To ensure that the admittance controller is activated only for sufficiently high values, and filter out measurement noise, each component of ${}^{S}\mathbf{h}_{E}$ is passed through the following deadband filter:

$$h_{adm,i} = \begin{cases} {}^{S}h_{E,i} + h_{m,i} & \text{if } {}^{S}h_{E,i} < -h_{m,i} \\ 0 & \text{if } \left|{}^{S}h_{E,i}\right| < h_{m,i} \\ {}^{S}h_{E,i} - h_{m,i} & \text{if } {}^{S}h_{E,i} > h_{m,i} \;\; i = 1\ldots6, \end{cases} \tag{17}$$

to generate the components of wrench vector $\mathbf{h}_{adm}$, that is the one actually input to the admittance controller. The sizes of the six deadband regions, $h_{m,i}$, are manually tuned, as will be explained just below.

We consider a spring-damper system, with null mass, and adjustable parameters (stiffness matrix $\mathbf{K}$, and damping matrix $\mathbf{B}$, both positive definite and diagonal):

$$\mathbf{h}_{adm} = \mathbf{K}{}^{S}\Delta\mathbf{p}_{E} + \mathbf{B}{}^{S}\Delta\dot{\mathbf{p}}_{E}. \tag{18}$$

Numerically differentiating on a sampling period Δt, and rearranging terms, yields the expression of ${}^{S}\Delta\mathbf{p}_{E}$ at time t:

$${}^{S}\Delta\mathbf{p}_{E}(t) = \Delta t\,(\Delta t\mathbf{K} + \mathbf{B})^{-1}\mathbf{h}_{adm}(t) + (\Delta t\mathbf{K} + \mathbf{B})^{-1}\mathbf{B}{}^{S}\Delta\mathbf{p}_{E}(t - \Delta t). \tag{19}$$

This controller relies on its output at the previous iteration ${}^{S}\Delta\mathbf{p}_{E}(t - \Delta t)$. Finally, a spatial motion transform matrix is used to convert ${}^{S}\Delta\mathbf{p}_{E}$ to ${}^{B}\Delta\mathbf{p}_{E}$, as required by the inverse kinematics controller (2).

The whole admittance control pipeline is outlined in Fig. 7. The procedure for tuning all the parameters ($h_{m,i}$, and the diagonal components of $\mathbf{K}$ and $\mathbf{B}$, noted respectively $k_{1\ldots6}$ and $b_{1\ldots6}$) is the following.

(1) The robot is manipulated away from the insertion tool support, with the end effector not in contact with the environment.
(2) The damping is set to zero in (19), so the ${}^{B}\Delta\mathbf{p}_{E}$, to be realized by the robot is a linear function of $\mathbf{h}_{adm}$ (pure spring model).
(3) The operator applies external torques/forces to the end effector along each of the six dof, and tunes the corresponding values of the deadband range $h_{m,1\ldots6}$ and of the stiffness $k_{1\ldots6}$, to obtain the desired deadband tolerance and static gain.
(4) $\mathbf{B}$ is introduced to remove the oscillations: the operator applies an external wrench, but this time the robot is controlled

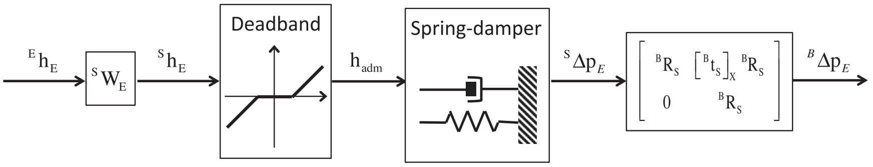


Fig. 7. Complete pipeline of the admittance controller, from measured external wrench $^E\mathbf{h}_E$ to desired pose variation $^B\Delta\mathbf{p}_E$.

with (19), and the $b_{1\ldots6}$ are tuned to obtain the desired response time for each component.

The values used in the experiments are given in Section 8.

7. Safety

This section presents the *safety* module. This relies on two tests. The first consists in checking whether any component of $^S\mathbf{h}_E$ is greater than the corresponding one in a threshold vector. Then, according to the current phase of the experiment, the robot either stops (‘safety stop’) or changes waypoint, using a strategy that will be detailed in Section 7.1. Second, the external camera is used to monitor safety of the insertion. If the operator hand is detected in the scene during robot motion, a *safety stop* is triggered. Details on the algorithm used for this are given in Section 7.2.

7.1. Dealing with strong external wrench

The external wrench on the end effector is used not only as input to the admittance controller presented just above, but also to monitor state changes or blocking situations during the assembly operation. Here also, we use its expression in the support frame, $^S\mathbf{h}_E$, derived with (15). The components of $^S\mathbf{h}_E$ are monitored either to check if a step has been completed, or to stop the motion in case of blockage. These two cases are indicated respectively by the left and right arrows exiting block *Test on external wrench* in Fig. 4.

In practice, steps 1 and 2(b) are considered completed if any component of $^S\mathbf{h}_E$ is greater than the corresponding one in threshold vectors τ_1 and τ_{2b}. If this is the case, even if the end effector has not yet reached the end point (respectively, P and C), the nominal reference is shifted to the start of the next trajectory (respectively, $\mathcal{T}_{2a}$ and $\mathcal{T}_{2c}$). During the connecting trajectory $\mathcal{T}_1$, we only monitor component $^S f_{E,Z}$:

$$\tau_1 = [+\infty \;\; +\infty \;\; \tau_{1,z} \;\; +\infty \;\; +\infty \;\; +\infty]. \tag{20}$$

Since this force is due to the first contact between the Rzeppa joint and the insertion tool, if it passes the threshold, we consider point P to be reached, and start the first inclination step, $\mathcal{T}^1_{2a}$. During the cage closing phase, $\mathcal{T}_{2b}$, a strong component of $^S\mathbf{h}_E$ will indicate that the closure is complete, and that the ball has been properly inserted in the inner race. Then, we consider that point C has been reached, and reverse the motion to return towards P, by following $\mathcal{T}_{2c}$.

Throughout operation, a safety stop is triggered if any component of $^S\mathbf{h}_E$ is greater than the corresponding one in threshold vector τ_s. The robot is blocked to avoid damaging the parts (or the robot itself), and the operator must manually unblock it, to resume operation. The image processing module (explained just below) ensures that motion is not resumed while the operator hand is in contact with the end effector.

7.2. Image processing

To safely interact with the human and with the environment, the designed controller relies on various sensed data. Along with the force/torque measures that are monitored as explained in the previous section, we utilise a camera to observe the scene around the insertion tool. Specifically, the camera should detect the operator hand, to trigger a safety stop, if the hand is near the robot, and restart motion, once a ball has been inserted.

The camera is placed with its optical axis parallel to the insertion tool, to maximize scene saliency. The type of image seen from the camera is shown in the bottom left of Fig. 3. Only the image greyscale is considered, and we assume that the operator hand is the lighter gray area in the scene. To facilitate detection, the operator wears white gloves, and the ground floor, below the end effector, is black. These are reasonable constraints, since the industrial cell can be partly structured, and since image processing is not the main contribution of this work. More sophisticated approaches, such as [42,43], could be applied to relax these constraints.

A simple image processing algorithm is used to derive the distance d_h between the hand and the insertion tool axis (see bottom right of Fig. 3). For this, two geometric parameters are predefined according to the relative pose of the camera and tool. These are: the locus of the tool axis (point A), and an annulus surrounding the tool, which is considered relevant for safety. Only the pixels in this annulus are taken into account (the grey pixels in the figure are excluded, and in the following paragraph, we refer to this annulus as the *image*).

The algorithm then consists in the following steps:

1. A two-dimensional Gaussian filter is applied to the image to improve the signal-to-noise ratio.
2. The image is binarized: pixels are turned black or white, by comparing their luminosity with a fixed threshold.
3. One erosion and two dilations are applied to the image, to connect regions of similar pixels (blobs).
4. The size of each white blob in the image is evaluated by counting the neighbour pixels.
5. The greatest white blob is selected; if it is sufficiently large, it is assumed to be the hand, and d_h is calculated. Otherwise, d_h is undefined.

If d_h is defined, and if it is below a minimum tolerated distance for safety, denoted d_m, a safety stop is triggered (see Fig. 4). The stop is only active while $d_h < d_m$. For example, if the operator has to manually unblock the robot as explained in Section 7.1, the robot will resume motion only when: it has been unblocked (all components of $^S\mathbf{h}_E$ are smaller than those of τ_s) *and* the operator hand is in safety (not in the camera field of view, or with $d_h > d_m$). This method is also used to close the Rzeppa cage: as soon as the hand exits the dangerous area, the ball is considered inserted, and the robot starts the closing phase (step 2(b)).

8. Experiments

8.1. Setup and nominal trajectory generation

To validate our framework, we have run a series of experiments with a 7 dof lightweight KUKA LWR IV[3] robot in the scenario

[3] www.kuka-labs.com/en

illustrated in Fig. 3 (top). This robot is well known for its flexibility, which has fostered its use in pHRI applications [44]. The controller sampling time is set to $\Delta t = 20$ ms (a specification imposed by the ICARO project software architecture). The $\mathbf{q}^*$ computed by the controller (see Fig. 4) are smoothed by the Reflexxes online trajectory generation library[4] before being sent to the robot joints. To get the interaction wrench ${}^E\mathbf{h}_E$, instead of mounting a force sensor on the end effector, we average, over a window of 4 samples, the wrench that is estimated by the robot controller through the FRI Interface[5] every 5 ms. This signal, which is derived from the applied and measured joint torques, proved to be accurate enough for our application. The camera used to monitor the human hand is a B & W Stingray F201B from Allied Vision Technologies, with resolution 1024 × 768 pixels. As secondary task in (2), we impose joint limit avoidance, via the following scalar cost function:

$$g(\mathbf{q}) = \frac{1}{2}\sum_{k=1}^{7}\left(\frac{q_k - q_{k,mid}}{q_{k,M} - q_{k,m}}\right)^2, \tag{21}$$

with $[q_{k,m}, q_{k,M}]$ the available range for joint k and $q_{k,mid} = (q_{k,M} + q_{k,m})/2$ its midpoint.

Two compliant controllers (in the operational and joint space) are already embedded in the KUKA LWR. However, we decided to use our own admittance controller (described in Section 6) for three reasons. First, we must control the robot in the joint space (to verify its configuration w.r.t. the environment and w.r.t. self-collisions), whilst defining the admittance characteristics (wrench references, damping and stiffness) in the operational space, and this is not feasible with the KUKA controller. Second, the admittance parameters must be varied online at the various assembly steps, and the KUKA controller can only operate with prefixed values. Finally, our method can be applied to any position-based industrial robot with wrench measurement, making the framework generic, and not limited to the LWR.

The manually taught trajectory is identified (see Fig. 6, left), as an arc of circle of radius $R = 0.241$ m, center position ${}^B\mathbf{t}_S = [-0.432 \;\; -0.421 \;\; 0.448]^\top$ m, and amplitude $\psi_M - \psi_m = 1.70$ radians. Then, to generate the corresponding articular joint trajectories, we apply (2) to track operational space trajectory (10), on a KUKA LWR IV, simulated in Webots[6] (see Fig. 6, right). The simulations are used to record the joint values (11), without endangering the real robot. We set $\Lambda = \mathbf{I}_6$, $\epsilon_t = 0.02$ and $\epsilon_r = 0.005$, to obtain the five primitive joint trajectories (12) for each of the seven joints, plotted in Fig. 8.

8.2. Control parameters

These trajectories are deformed online in case of external contacts, using the admittance controller described in Section 6. For the deadband filter, we apply the following threshold: $\mathbf{h}_m = [8 \; 8 \; 8 \; 3 \; 3 \; 3]^\top$, (i.e., force components below 8 N and torque components below 3 Nm are not taken into account). The stiffness and damping matrix components are set to:

$$\begin{aligned} &k_{1,2} = 250 \text{ N/mm} \quad k_3 = 500 \text{ N/mm} \quad k_{4,5,6} = 170 \text{ Nm/mrad} \\ &b_{1,2} = 0.005 \text{ Ns/m} \quad b_3 = 0.01 \text{ Ns/m} \quad b_{4,5,6} = 0.003 \text{ Nms/rad}. \end{aligned} \tag{22}$$

The threshold vectors that are used to check if a step has been completed, or to stop the motion (see Section 7.1), are as follows (forces expressed in N, moments in Nm).

- To verify that P has been reached (i.e., that the tool has been inserted), we only monitor ${}^S\mathbf{f}_{E,Z}$, according to (20): $\tau_{1,z} = 40$ N.

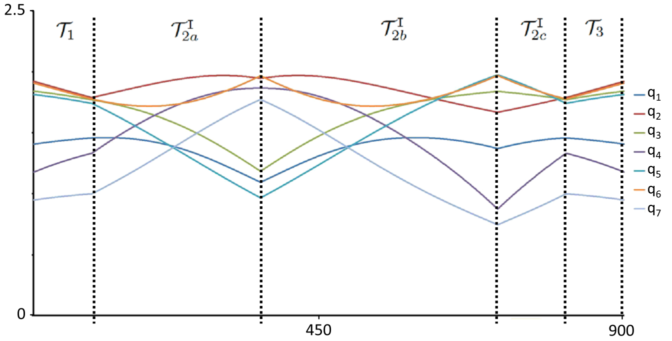


Fig. 8. The five primitive trajectories in joint space, obtained by inverse kinematics in Webots (abscissa: iterations, ordinate: joint angles in rad).

- To verify that the closure is complete, we compare the wrench to the threshold:

$$\boldsymbol{\tau}_{2b} = [70 \; 40 \; 70 \; 15 \; 15 \; 15]^\top. \tag{23}$$

The reason for setting the threshold on ${}^S\mathbf{f}_{E,Y}$ smaller than ${}^S\mathbf{f}_{E,X}$ and ${}^S\mathbf{f}_{E,Z}$ is that, in our setup, when the closure is completed, the most relevant forces appear along the support frame Y-axis (see Fig. 3).

- To block the robot:

$$\boldsymbol{\tau}_s = [70 \; 70 \; 70 \; 25 \; 25 \; 25]^\top. \tag{24}$$

In this case, all directions are considered equally relevant.

Although we used the ViSP library [45] for visualization, the image processing algorithms for detecting and tracking the operator hand were developed from scratch, as explained in Section 7.2. The threshold for considering the hand dangerous is set to $d_m = 300$ pixels, while the relevant annulus minimum and maximum radii are respectively 180 and 750 pixels. The whole image processing pipeline takes only 5 ms, well below both the controller (20 ms) and camera framegrabber (33 ms) sampling times.

8.3. Results and discussion

The experiments consisted in having various users insert the five balls in the Rzeppa joint. We hereby discuss the results of these experiments, which are shown in Fig. 9, and in the video available at https://youtu.be/3KWduKKSyy8.

On the top of Fig. 9, the relevant steps of a complete successful experiment are shown. These include: tool connection/disconnection (1, 8), ball insertion (2, 3, 7), joint pivoting (4, 5), vision triggered emergency stop (6), and joint manual unmounting (9, 10).

The joint trajectories obtained during this experiment are plotted in Fig. 10. In this figure, the emergency stops activated by vision are highlighted in yellow, and the insertion phases for each ball are indicated with the black rectangles. Correspondingly, the robot motion is stopped. As the curves show, the values of $q_{1\ldots6}$ are the same for each of the five balls, whereas q_7 changes, to pivot the cage before the start of the next ball insertion (just before the vertical dashed lines). In the current version of the software, the end of the insertion phase is notified by keyboard pressing, to resume robot motion. The reason is that it is very difficult, and unnecessary, to determine this state automatically. In future work, this notification could be given via a pedal, to free the operator hands.

The admittance controller is indispensable to avoid blockage between robot and tool, which systematically occurs when it is deactivated. Its effect can be seen in Fig. 11, where we have plotted the components of the external wrench ${}^S\mathbf{h}_E$, along with the

[4] www.reflexxes.com

[5] http://cs.stanford.edu/people/tkr/fri/html/

[6] www.cyberbotics.com

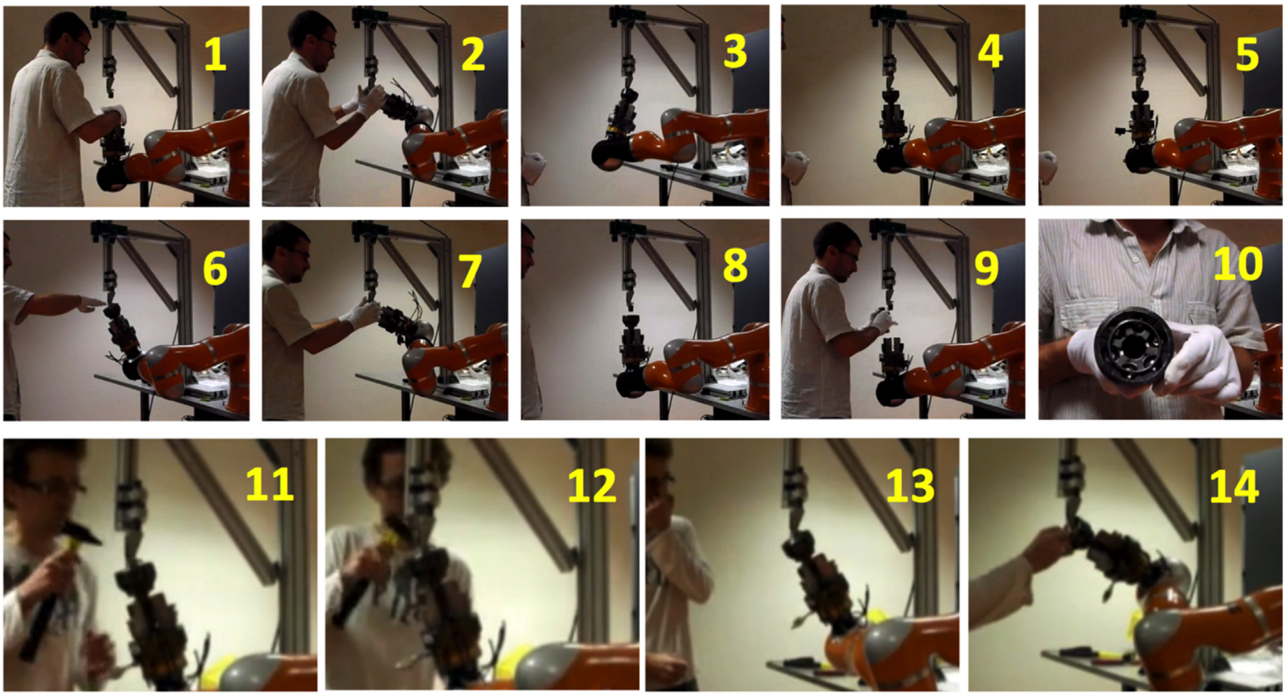


Fig. 9. Top: first experiment. (1) Tool connection. (2, 3) Second ball insertion. (4, 5) Joint pivoting. (6) Vision triggered emergency stop. (7) Sixth ball insertion. (8) Tool disconnection. (9, 10) Unmounting the assembled Rzeppa joint. *Bottom*: second experiment. (11, 12). The operator unblocks the joint using a hammer. (13, 14) Motion resumes and a new ball can be inserted.

corresponding variation of the articular trajectories. Apart from the peaks on ${}^{S}\mathbf{f}_{E,Y}$ (red curves in the top figure), which occur at each of the five insertion phases, it can be seen that throughout the assembly, the admittance controller induces variations of the order of magnitude of 0.02 rad (1 degree) to the articular joint values.

Although the whole assembly takes approximately 120 s, i.e., almost 4 times more than the manual one, the following aspects must be considered.

- Part of the delays are due to the current cell configuration. In particular, when the Rzeppa joint opens, the cage is horizontal. This complicates the ball insertion task, and makes, in this experiment, the operator drop the second ball to the ground, causing a 25 s insertion phase (first black rectangle in Fig. 10). To solve this problem, in future work, we plan to incline the insertion tool w.r.t. the ground, so that the housing is horizontal during ball insertion.
- We have deliberately decided to operate at 40% of the maximum speed allowed in physical Human–Robot Interaction (0.25 ms^{-1}). In fact, throughout the experiments, the tool center point velocity is always below 0.1 ms^{-1}. However, given the low image processing sampling time (5 ms), we are confident that increasing the robot speed towards the 0.25 ms^{-1} limit would not pose problems.
- Note that, even if the collaborative assembly was executed at the maximum speed allowed in physical Human–Robot Interaction (0.25 ms^{-1}), it would still be slower than the manual one. However, since the operator activity over one cycle time is smaller, it could be possible to have more robots working with a single operator, in order to leverage his/her production time.
- Another gain can be obtained by pivoting the Rzeppa joint during step 2(c). This would suppress step 2(d), which currently takes approximately 10% of the entire cycle time (2.5 s per ball).

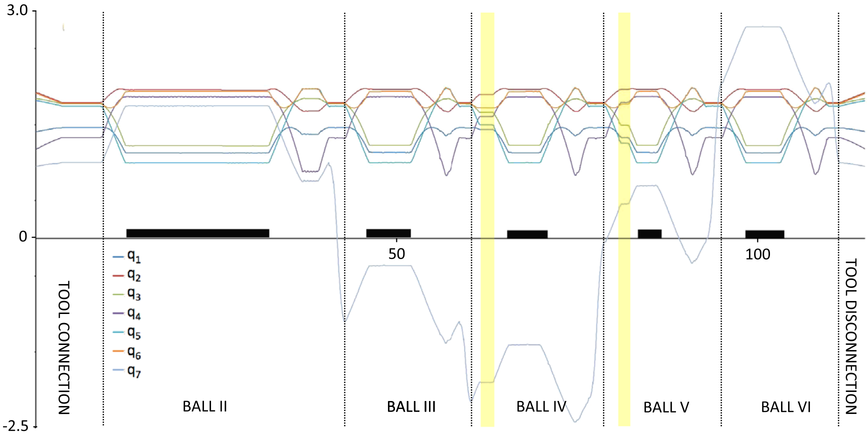


Fig. 10. The joint trajectories, during the first experiment (abscissa: time in s, ordinate: joint angles in rad). The emergency stops activated by vision are highlighted in yellow, and the insertion phases for each ball are indicated with the black rectangles. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

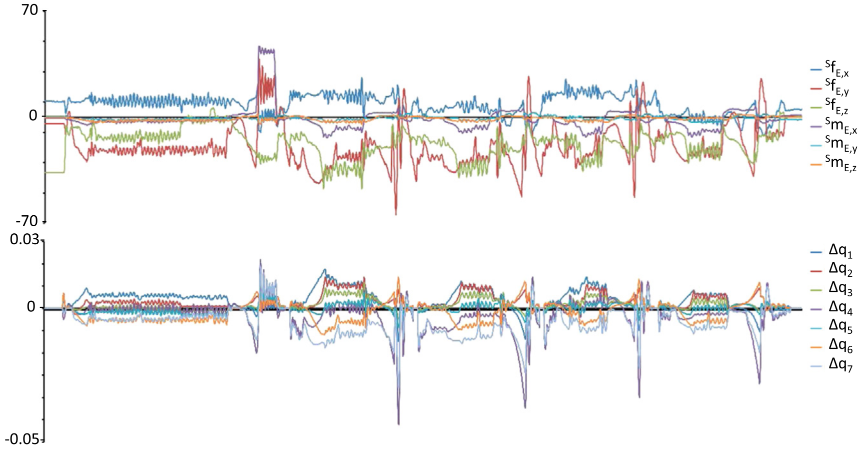


Fig. 11. Six components of the external wrench ${}^{S}\mathbf{h}_E$ (top, in N and Nm), and corresponding variation of the joint trajectories, $\Delta\mathbf{q}$, produced by the admittance controller (bottom, in rad). The abscissa timescale is identical to that of Fig. 10. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

- The collaborative robot assembly substantially reduces the risk of strain injuries with respect to the manual assembly, since most of the physical effort is realized by the robot. In fact, the operator load is reduced by approximately 60%, and, following this work, the Rzeppa assembly cell was reclassified in the PSA ergonomics scale (from 'red' to 'medium' level).

The experiments have been repeated by two users, to show the simplicity and robustness of the system. The video shows four experiments, where the entire assembly is completed, twice by each user. The last experiment (shown in the bottom of Fig. 9) is of particular interest, since it shows a case where the robot motion is blocked due to excessive external wrench (above threshold (24)). In this case, the user has to manually unblock the cage, using a hammer, until motion resumes (when the wrench returns within the safety range). It is noteworthy to point out that this also often occurs during manual assembly by experienced users. Typically, a slight misplacement of the third ball can jam the inner race and housing, requiring intervention with a hammer. The advantage of our framework is that these situations are automatically managed at a high level, avoiding permanent blockage of the robot.

9. Conclusions

This paper presents preliminary results on the development of a collaborative human–robot manufacturing cell for Rzeppa joint assembling. In contrast with most human–machine manufacturing applications, this one requires direct physical contact between robot and human, as well as between robot and environment.

The proposed framework integrates many state-of-the-art robotics components, seldom applied in real industrial scenarios. These include: trajectory optimization, admittance control, and image processing. The approach is validated in a series of experiments, with different users. With the proposed visual gesture monitoring and intrinsic collision detection, the robot setup is compatible with the safety standards, and could be certified. In addition, PSA has conducted a complete risk analysis of the application, to ensure that there is no obstacle in the deployment of such technology. Although the cycle time is lower than that of manual assembly, the collaborative cell lightens the operator burden, and substantially reduces the risk of strain injuries. Furthermore, the causes of the assembly delays have been identified, and we are confident that slight adjustments in this sense would radically reduce the cycle time, and spur the transfer of the proposed technology.

If successfully deployed in automotive factories, the proposed robotic assistant would have a very quick return on investment. Indeed, in spite of the high price of the hardware, the expected reduction of MSDs and associated costs, is tremendous. It has been estimated, by PSA, that the savings in terms of activity limitations and disability could cover the hardware cost of the setup in approximately a year.

Following this work, the two groups (LIRMM and PSA) are pursuing research in this field, in the context of other French National Projects. Future work will consist in deploying the proposed methodologies on mobile manipulator robots, which represent key technologies in flexible manufacturing.

Acknowledgements

This work has been supported by the ANR (French National Agency) ICARO project, grant number ANR-10-CORD-0025.

Appendix A. Supplementary material

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.rcim.2015.12.007.

References

[1] P. Akella, M. Peshkin, E. Colgate, W. Wannasuphoprasit, N. Nagesh, J. Wells, S. Holland, T. Pearson, B. Peacock, Cobots for the automobile assembly line, in: IEEE International Conference on Robotics and Automation, ICRA, 1999.

[2] J. Kruger, T.K. Lien, A. Verl, Cooperation of human and machines in assembly lines, CIRP Ann. – Manuf. Technol. 58 (2009) 628–646.

[3] G. Michalos, S. Makris, N. Papakostas, D. Mourtzis, G. Chryssolouris, Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach, CIRP J. Manuf. Sci. Technol. 2 (2) (2010) 81–91.

[4] G. Michalos, S. Makris, J. Spiliotopoulos, I. Misios, P. Tsarouchi, G. Chryssolouris, ROBO-PARTNER: Seamless Human–Robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future, in: The Fifth CIRP Conference on Assembly Technologies and Systems (CATS), 2014.

[5] J. Shi, G. Jimmerson, T. Pearson, R. Menassa, Levels of human and robot collaboration for automotive manufacturing, in: Workshop on Performance Metrics for Intelligent Systems, PerMIS, 2012, pp. 95–100.

[6] W. Knight, How Human–Robot teamwork will upend manufacturing - MIT Technology Review, ⟨www.technologyreview.com⟩.

[7] New human–robot cooperation in Audi's production processes - Feb 16, 2015, ⟨www.audi.com⟩.

[8] N. Papakostas, G. Michalos, S. Makris, D. Zouzias, G. Chryssolouris, Industrial applications with cooperating robots for the flexible assembly, Int. J. Comput. Integr. Manuf. 24 (7) (2011) 650–660.

[9] J. Tan, D. Feng, Y. Zhang, K. Watanabe, R. Kato, T. Arai, Human–robot

collaboration in cellular manufacturing: Design and development, in: IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, 2009.

[10] M. Morioka, S. Sakakibara, A new cell production assembly system with human–robot cooperation, CIRP J. Manuf. Sci. Technol. 59 (2010) 9–12.

[11] N. Pedrocchi, F. Vicentini, M. Malosio, L.M. Tosatti, Safe human–robot cooperation in an industrial environment, Int. J. Adv. Robot. Syst., IJARS 10 (27) (2012).

[12] H. Ding, M. Schipper, M. Bjoern, Optimized task distribution for industrial assembly in mixed human–robot environments – case study on IO module assembly, in: IEEE International Conference on Automation Science and Engineering, CASE, 2014.

[13] A. Bicchi, M. Peshkin, J. Colgate, Safety for physical human–robot interaction, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer, 2008, pp. 1335–1348.

[14] A. De Luca, F. Flacco, Integrated control for phri: collision avoidance, detection, reaction and collaboration, in: IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, BIOROB, 2012.

[15] C. Heyer, Human–robot interaction and future industrial robotics applications, in: IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, 2010, pp. 4749–4754.

[16] A. De Santis, B. Siciliano, A. De Luca, A. Bicchi, An atlas of physical human–robot interaction, Mech. Mach. Theory 43 (3) (2007) 253–270.

[17] S. Haddadin, A. Albu-Schäffer, G. Hirzinger, Requirements for safe robots: measurements, analysis and new insights, Int. J. Robot. Res., IJRR, 2009.

[18] F. Flacco, A. De Luca, Safe physical human–robot collaboration, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2013.

[19] J. Fryman, M. Bjoern, Safety of industrial robots: From conventional to collaborative applications, in: The Seventh German Conference on Robotics ROBOTIK, 2012.

[20] L. Roveda, F. Vicentini, L.M. Tosatti, Deformation-tracking impedance control in interaction with uncertain environments, in: IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, 2013.

[21] G. Ganesh, A. Albu-Schäffer, M. Haruno, M. Kawato, E. Burdet, Biomimetic motor behavior for simultaneous adaptation of force, impedance and trajectory in interaction tasks, in: IEEE International Conference on Robotics and Automation, ICRA, 2010.

[22] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schäffer, E. Burdet, Human-like adaptation of force and impedance in stable and unstable interactions, IEEE Trans. Robot. (2011). http://dx.doi.org/10.1109/TRO.2011.2158251.

[23] D. Surdilovic, G. Schreck, U. Schmidt, Development of collaborative robots (cobots) for flexible human-integrated assembly automation, in: 41st International Symposium on Robotics, ISR, 2010.

[24] F. Lange, W. Bertleff, M. Suppa, Force and trajectory control of industrial robots in stiff contact, in: IEEE International Conference on Robotics and Automation, ICRA, 2013.

[25] M. Suphi Edren, T. Tomiyama, Human intent detection and physically interactive control of a robot without force sensors, IEEE Trans. Robot. 26 (2) (2010) 370–382.

[26] M. Suphi Edren, B. Maric, Assisting manual welding with robot, Robot. Comput. Integr. Manuf. 27 (2011) 818–828.

[27] M. Suphi Edren, A. Billard, End-point impedance measurements at human hand during interactive manual welding with robot, in: IEEE International Conference on Robotics and Automation, ICRA, 2014.

[28] G. Ferretti, G. Magnani, P. Rocco, Assigning virtual tool dynamics to an industrial robot through an admittance controller, in: International Conference on Advanced Robotics, ICAR, 2009.

[29] F. Ficuciello, A. Romano, L. Villani, B. Siciliano, Cartesian impedance control of redundant manipulators for human–robot co-manipulation, in IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, 2013.

[30] N. Hogan, Impedance control: an approach to manipulation: parts I–III, ASME J. Dyn. Syst. Meas. Control 107 (1985) 1–24.

[31] G. Grunwald, A. Albu-Schäffer, G. Hirzinger, Touch: the direct type of human interaction with a redundant service robot, in: IEEE International Workshop on Robot and Human Interactive Communication, ROMAN, 2001.

[32] C. Perez Quintero, R. Tatsambon Fomena, A. Shademan, N. Wolleb, T. Dick, M. Jagersand, Sepo: Selecting by pointing as an intuitive human–robot command interface, in: IEEE/RSJ International Conference on Intelligent Robots and Systems IROS, 2013.

[33] V. Lippiello, B. Siciliano, L. Villani, Interaction control of robot manipulators using force and vision, Int. J. Optomechatron. 2 (3) (2008) 257–274.

[34] A. De Santis, V. Lippiello, B. Siciliano, L. Villani, Human–robot interaction control using force and vision, Adv. Control Theory Appl. 353 (2007) 51–70.

[35] A. Cherubini, R. Passama, A. Meline, A. Crosnier, P. Fraisse, Multimodal control for human–robot cooperation, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2013.

[36] A. Lasnier, ANR ICARO D2.1 - Description détaillée des besoins, PSA, ANR ICARO Project Confidential Deliverable (in French), 2012.

[37] K. Waldron, J. Schmiedeler, Kinematics, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer, 2008, pp. 9–33.

[38] B. Siciliano, L. Sciavicco, L. Villani, G. Oriolo, Robotics: Modelling, Planning and Control, Springer, 2009.

[39] A. Liegeois, Automatic supervisory control of configurations and behavior of multibody mechanisms, IEEE Trans. Syst. Man Cybern. 7 (6) (1977) 868–871.

[40] N. Mansard, O. Khatib, A. Kheddar, A unified approach to integrate unilateral constraints in the stack of tasks, IEEE Trans. Robot. 25 (3) (2009).

[41] Robotiq. Collaborative robot e-book – ⟨http://blog.robotiq.com⟩.

[42] I. Oikonomidis, N. Kyriazis, A.A. Argyros, Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints, in: International Conference on Computer Vision ICCV, 2011.

[43] S.S. Rautaray, A. Agrawal, A real time hand tracking system for interactive applications, Int. J. Comput. Appl. 18 (March (6)) (2011) 28–33.

[44] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, G. Hirzinger, The KUKA-DLR Lightweight Robot Arm – A new reference platform for robotics research and manufacturing, in: the 41st International Symposium on Robotics (ISR) and the Sixth German Conference on Robotics (ROBOTIK), 2010.

[45] E. Marchand, F. Spindler, F. Chaumette, Visp for visual servoing: a generic software platform with a wide class of robot control skills, IEEE Robot. Autom. Mag., Special Issue Softw. Packages Vision-Based Control Motion 12 (4) (2005) 40–52.