Robotics and Autonomous Systems 136 (2021) 103711

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

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

Human–robot collaboration in sensorless assembly task learning enhanced by uncertainties adaptation via Bayesian Optimization

Loris Roveda [a,*], Mauro Magni [b], Martina Cantoni [b], Dario Piga [a], Giuseppe Bucca [b]

[a] Istituto Dalle Molle di studi sull'Intelligenza Artificiale (IDSIA), Scuola Universitaria Professionale della Svizzera Italiana (SUPSI), Università della Svizzera Italiana (USI) IDSIA-SUPSI, 6928 Manno, Switzerland
[b] Department of Mechanical Engineering, Politecnico di Milano, 23900 Lecco, Italy

ARTICLE INFO

Article history:
Received 13 July 2020
Received in revised form 14 October 2020
Accepted 30 November 2020
Available online 13 December 2020

ABSTRACT

Robots are increasingly exploited in production plants. Within the Industry 4.0 paradigm, the robot complements the human's capabilities, learning new tasks and adapting itself to compensate for uncertainties. With this aim, the presented paper focuses on the investigation of machine learning techniques to make a sensorless robot able to learn and optimize an industrial assembly task. Relying on sensorless Cartesian impedance control, two main contributions are defined: (1) a task-trajectory learning algorithm based on a few human's demonstrations (exploiting Hidden Markov Model approach), and (2) an autonomous optimization procedure of the task execution (exploiting Bayesian Optimization). To validate the proposed methodology, an assembly task has been selected as a reference application. The task consists of mounting a gear into its square-section shaft on a fixed base to simulate the assembly of a gearbox. A Franka EMIKA Panda manipulator has been used as a test platform, implementing the proposed methodology. The experiments, carried out on a population of 15 subjects, show the effectiveness of the proposed strategy, making the robot able to learn and optimize its behavior to accomplish the assembly task, even in the presence of task uncertainties.

© 2020 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

1.1. Context

Industry 4.0 paradigm [1] is proposing an enhanced production environment in which robots are increasingly required to be intelligent systems [2]. In fact, robots (and in particular collaborative robots) bring many advantages to the production environment, reducing safety risks [3,4] and being flexible solutions in terms of usage and programming [5,6]. To improve industrial production, the robot should be able to carefully monitor its working scene [7], safely interacting with humans and with the environment [8,9], learning new skills/tasks and adapting its behavior to the specific context [10,11]. Assembly tasks are one of the most required industrial applications to be executed by a manipulator, autonomously or in collaboration with the human operator [12,13]. Assembly tasks are still nowadays not trivial to be performed by a robot, requiring a deep knowledge of the assembly procedures, materials, geometry of the parts, etc. [6]. Therefore, *ad hoc* programming of such kind of tasks is commonly difficult and time-consuming, resulting in rigid applications that cannot be adapted to different situations. Thus, to implement a flexible and intelligent robotic cell, the robot has to be able to learn, optimize and adapt its behavior to the specific working scene with as low as possible programming effort. In such a scenario, it is of fundamental importance the possibility to easily transfer the human's task-related knowledge to the robotic system [14,15]. Human's demonstrations-based approaches are one of the most powerful methodologies for such a purpose [16,17]. Exploiting such approaches it is in fact possible to intuitively teach a target task to the robot. Once the task is learned, autonomous task optimization and adaptation methodologies have also to be implemented to give to the robot a fully autonomous and flexible behavior [18,19]. In the following, the state of the art related to learning, optimization and adaptation of robotized tasks is analyzed.

1.2. Related works

Machine learning offers to robotics a framework and a set of tools for the design of sophisticated and hard-to-engineer behaviors and applications; conversely, the challenges of robotic problems provide both inspiration, impact, and validation for developments in robot learning [20,21]. Machine learning techniques have been widely applied to robotics in different fields, such as motor skill learning for human–robot collaboration purposes [22], robot model learning [23], control design and tuning [24], grasping capabilities [25], etc.

* Corresponding author.
E-mail address: loris.roveda@idsia.ch (L. Roveda).

https://doi.org/10.1016/j.robot.2020.103711
0921-8890/© 2020 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Machine learning has found a huge application in robotized task learning [26]. In particular, (i) autonomous task learning methodologies and (ii) collaborative task learning methodologies can be highlighted from the state of the art analysis. Considering (i), the main objective of such approaches is to make the robot able to learn a specific task without any interaction with the human. The following contributions can be identified related to this kind of methodology. In [27] the iterative learning and reinforcement learning procedures are applied to an automotive industrial assembly task, to automatize the compliance controllers parameters tuning. In [28] a policy search method is used to learn a range of dynamics manipulation behaviors without using known models or example demonstrations. [29] proposes a method that combines motion planning with reinforcement learning policy search for efficient learning of assembly tasks. In [30] a team of heterogeneous robots assembles autonomously a furniture kit, following a generated assembly sequence. [31] applies a reinforcement learning method based on motor primitives to hit a baseball with an anthropomorphic robotic arm. A recent deep reinforcement learning algorithm based on off-policy training shows that a robot can learn a variety of 3D manipulation skills without any prior demonstrations or manually designed representations [32]. Considering (ii), the main objective of such approaches is to make the robot able to learn a specific task on the basis of human's demonstrations. The following contributions can be identified related to this kind of methodology. [33] developed a method for learning and reproduction of complex trajectories for robot programming by demonstration applied to a painting process. In [34] the robot learns a reward function from a demonstration and a task model from repeated attempts to perform the task, thanks to the application of reinforcement learning. In [35] a *Programming by Demonstrations* approach allows the end-user to *program* the robot simply by showing it how to perform the task: in fact, the teacher does several demonstrations of the task of juicing an orange, by changing the location of each item to allow the robot to generalize correctly. The same approach is used in [36], where a robot learns how to make a chess move by generalizing across different demonstrations. In [37] demonstrations recorded through human tele-operation are used in a robotic arm assembly task. In [38] the compliance of the robot controller is adjusted allowing a human teacher to indicate compliance variations by physically interacting with the robot during the task execution.

Assembly tasks are one of the most investigated applications considering task learning methodologies [39]. In fact, it is not trivial to define *a priori* the sequence of all the subtasks, the control parameters, etc., making the robot able to perform the specific task. In [40] Reinforcement Learning (RL) is applied to compute a geometric motion plan for the robot executing the assembly. Such computation is based on the available parts CAD data. In [41] a strategy to guide the user during the assembly task phases is proposed, on the basis of the assembly CAD data. In [42] an optimization algorithm is presented in combination with impedance control strategy to optimize the robotic dual peg-in hole assembly task, reducing the assembly time while smoothing the contact forces during assembly process. In [43] the learning of the required assembly forces is performed on the basis of human's demonstrations. In [44] a self-supervised learning methodology is presented to perform multimodal representations of contact-rich tasks, and it has been demonstrated in a peg-in-hole task.

Once the reference task has been learned, its (autonomous) optimization (*i.e.*, optimization of control variables to maximize the task performance) and (eventually) adaptation can be performed in order to maximize the task performance while compensating for task uncertainties. Sources of uncertainties are related to the working scenario layout (*i.e.*, positioning of the manipulated parts), assembly properties (*i.e.*, assembly tolerances and parts materials), and noise/uncertainties on the learned task trajectories (*i.e.*, uncertainties on the reference task teached by a human, *e.g.*, containing measurements noise). In fact, considering real industrial production plants, positioning of the parts and their geometry are not always the same, affecting the task execution performance and possibly resulting in task failures. Therefore, task optimization and adaptation capabilities are fundamental in order to enhance the robotic cell with an intelligent behavior. However, the state of the art approaches related to assembly task learning are only considering fixed and well-defined scenarios, not including such sources of uncertainties. Once the task learning is completed, there is no possibility to adapt its execution to a (partially) new scenario. In addition, the learning, optimization and adaptation of an assembly task with a standard sensorless (*i.e.*, no force/torque sensor is used to reduce hardware costs and setup time) industrial robot is even more difficult [45, 46]. In fact, without the measurements of the interaction forces the assembly procedure has to rely only on the robot Cartesian pose information.

1.3. Paper contribution

The here presented paper proposes a methodology to make a sensorless robot able to learn and optimize an industrial assembly task. In fact, standard industrial robots are not equipped with force/torque sensors to measure the interaction force, requiring additional costs and efforts in order to integrate such devices in the robot. Relying on sensorless Cartesian impedance control (extendible to position control-based sensorless impedance control for the robots not equipped with torque control), two main contributions are defined: (1) a task-trajectory learning algorithm, and (2) an optimization procedure of the task execution. Considering (1), a few human's demonstrations (from a single operator) of the assembly task execution are performed. The robot (in gravity compensation control mode, *i.e.*, not requiring a force sensor for such collaborative procedure) is manually guided by the human to perform the task, while recording the Cartesian end-effector pose. The recorded data (considering each translational Cartesian degree of freedom and each angular velocity component separately) are then processed by a *Hidden Markov Model* (HMM) approach. The HHM approach is able to select the nominal reference assembly task trajectory to be given to the sensorless Cartesian impedance control as the target set-point. The HMM approach aims to the following contributions:

- teach the task to the robot on the basis of a single operator demonstrations;
- teach the task to the robot exploiting a few (between 3 and 5) repetitions of the target application in order to eliminate/reduce teaching uncertainties. In fact, a single demonstration might contain uncertainties in robot guidance, while the proposed HMM method is capable with few demonstrations to select the most reliable one.

Considering (2), a *Bayesian Optimization* (BO)-based algorithm has been designed to autonomously optimize (*i.e.*, optimization of control variables to maximize the task performance) the robot sensorless Cartesian impedance control parameters. The objective of the BO algorithm is to maximize the task performance (*i.e.*, avoiding task failures while reducing the interaction force), making the robot able to compensate for task uncertainties. For this aim, the learned nominal reference assembly task trajectory (*i.e.*, the sensorless Cartesian impedance control set-point) is optimized, together with the control parameters (*i.e.*, stiffness and damping sensorless Cartesian impedance control matrices). The BO approach aims to the following contributions:

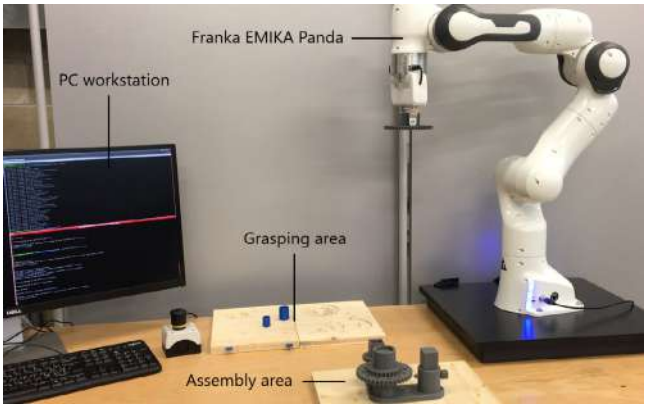


Fig. 1. Reference assembly task of a gear component performed by a Franka EMIKA panda robot.

- compensate for task uncertainties (such as parts positioning);
- autonomous (re)-learning of the task in new conditions (such as new positioning of the parts or new geometries to be manipulated);
- autonomous tuning of the control parameters.

The task-trajectory learning algorithm and the optimization procedure are executed separately for each subtask of the assembly on the basis of a predefined sequence of the subtasks. Such predefined sequence of subtasks can be automatically computed [47] or identified from the part assembly-datasheet.

With respect to Dynamic Movement Primitives [48] and deep learning [49] approaches for task learning applications, the here presented HMM methodology allows to achieve a fast teaching of the specific task with few demonstrations, being easily implementable and applicable. In addition, the proposed optimization procedure allows to avoid the classical "record and play" approaches to reproduce a teached task. In fact, in the case of teaching and/or task uncertainties, such approaches result in a task failure. The employed BO approach allows in addition to limit the number of required experiments in order to optimize the task.

To the best knowledge of the authors, the state of the art approaches devoting to assembly task learning are considering fixed and well-defined scenarios. Therefore, such approaches do not allow to compensate for task uncertainties (as described in Section 1.2). The here presented paper, instead, is capable to both learn a nominal task and (re)optimize it if required, in order to face the above mentioned task uncertainties. This objective is the core novelty of the proposed paper.

To validate the proposed methodology, an assembly task has been selected as a reference application. The task consists of mounting a gear into its square-section shaft on a fixed base to simulate the assembly of a gearbox (Fig. 1). A Franka EMIKA Panda manipulator has been used as a test platform, implementing the proposed methodology. 15 subjects have been involved in the experimental validation. Each subject is treated separately, teaching its own assembly task independently from the other subjects. Each subject performed 5 task demonstrations. The HHM approach has been applied to learn (independently for each subject) the nominal reference assembly task trajectory. After the nominal reference assembly task trajectory is learned, the assembly location has been modified in order to introduce position uncertainties into the working scene (*i.e.*, nominal demonstrations conditions are different from optimization and testing conditions for each subject). The BO algorithm has been then applied (independently to each learned nominal assembly task trajectory, one for each subject) to optimize the task execution, adapting the robot behavior to the new working scenario. Experimental results show the effectiveness of the proposed strategy, making the robot able to learn and optimize its behavior to accomplish the assembly task, even in the presence of task uncertainties. The proposed methodology has been compared with the non-optimized assembly task (*i.e.*, only the nominal assembly task trajectory learning is performed for each subject independently) showing the importance of the BO-based methodology for task optimization and adaptation purposes (assembly task success rate equal to 93% for the HMM+BO approach, assembly task success rate equal to 19% for the HMM approach).

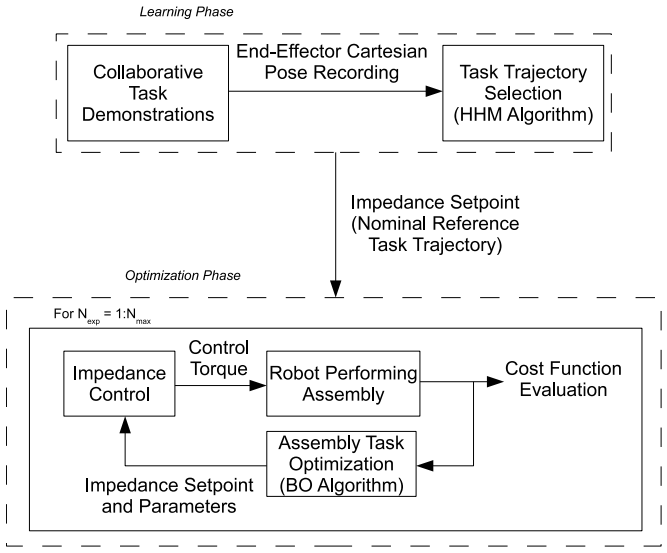


Fig. 2. Human–robot collaborative framework for assembly task learning and optimization. The sensorless Cartesian impedance controller is exploited to perform the task execution. The collaborative task learning procedure (based on the HMM approach) is exploited to define the nominal task reference trajectory. The iterative (from experiment $N_{exp} = 1 : N_{max}$, where N_{max} is the maximum number of experiments) optimization procedure (based on the BO approach) is exploited to tune the control parameters to maximize the task performance.

1.4. Paper outline

The paper is structured as follows. Section 2 defines the proposed architecture for collaborative task learning and optimization purposes, highlighting the main components. Section 3 describes the implemented robot controller. Section 4 presents the proposed approach for collaborative learning of nominal task trajectory based on the Hidden Markov Model approach. Section 5 presents the proposed approach for task optimization based on the Bayesian optimization approach. Section 6 describes the validation scenario and the achieved results. Conclusions and directions for future works are given in Section 7.

2. Methodology

Relying on sensorless Cartesian impedance control, the proposed paper defines a human–robot collaborative framework for assembly task learning and optimization. On the one hand, the assembly task learning is performed exploiting a few human's demonstrations. The robot (in gravity compensation control mode, *i.e.*, do not requiring a force/torque sensor for the human–robot collaboration) is manually guided by the human performing the task. The robot end-effector Cartesian positions and angular velocities are recorded during the task executions. The recorded data (considering each Cartesian degree of freedom – DoF – separately) for each demonstration are then processed by

a Hidden Markov Model (HMM) approach, selecting the nominal reference assembly task trajectory from the set of demonstrations to be given to the sensorless Cartesian impedance control as the target set-point. The main contribution of the HMM approach is to limit the teaching uncertainties inherently included into the human's demonstrations. On the other hand, a Bayesian Optimization (BO)-based algorithm has been designed to autonomously optimize the robot sensorless Cartesian impedance control parameters. The BO algorithm maximizes the task performance on the basis of a defined cost function, evaluated for each experimental iteration: task failures have to be avoided while reducing the interaction force. The proposed BO methodology optimizes the reference assembly task trajectory (*i.e.*, the sensorless Cartesian impedance control set-point computed by the HMM algorithm), together with the sensorless Cartesian impedance control parameters (*i.e.*, stiffness and damping parameters). The main contribution of the BO is to compensate for task uncertainties (*e.g.*, parts positioning). The proposed framework is shown in Fig. 2.

The nominal reference task trajectory learning algorithm and the optimization procedure are executed separately for each subtask of the assembly on the basis of a predefined sequence. Considering a peg-in-hole task [50] or an assembly task [6] with insertion direction along the vertical Cartesian DoF z, three main subtasks can be identified:

- *approaching subtask* (subtask #1): the robot has to positioning the part to be assembled in the proximity of the assembly location;
- *insertion subtask* (subtask #2): the robot has to perform the part insertion (*e.g.*, a peg positioned into its hole, or a gear positioned into its shaft);
- *pushing subtask* (subtask #3): the final pushing is performed in order to complete the assembly (*e.g.*, a pushing force is required to complete a mechanical fixing of the part on a reference surface).

Such three subtasks will be considered for the task-trajectory learning algorithm and for the optimization procedure. In general, the predefined sequence of subtasks related to a specific assembly can be automatically computed [47] or identified from the part assembly-datasheet.

It is important to underline the motivation behind the selection of the proposed algorithms for nominal task learning and optimization (*i.e.*, HMM and BO, respectively). Considering nominal task learning phase, the main objective has been related to select a methodology allowing a fast teaching procedure (*i.e.*, with limited demonstrations) of the specific task. As detailed in Section 1.3, w.r.t. other state-of-the-art techniques, HMM approach shows the advantage to be able to select the nominal task from a limited set of demonstrated ones [51]. Considering optimization phase, the main objective has been related to select a methodology capable to reduce the required experimental trials. W.r.t. other state-of-the-art techniques, Bayesian Optimization allows, in fact, to minimize the required experiments for optimization purposes, balancing exploration and exploitation behaviors [52].

In the following Sections, the sensorless Cartesian impedance controller, the nominal reference task trajectory learning methodology and the task optimization methodology are described.

Remark 1. In this paper, an assembly task with insertion direction along the vertical Cartesian DoF z is considered. The extension of the approach to a general assembly location pose is straightforward.

Remark 2. The optimization procedure might consists in an optimization from scratch, or in a re-optimization in the case that specific tasks conditions change, introducing new sources of uncertainties (*e.g.*, re-location of the parts or assembly location in the robot workspace).

3. Sensorless Cartesian impedance control

In this Section, the sensorless Cartesian impedance control is described. Such control methodology allows to avoid any use of force/torque sensor, implementing a compliant behavior for the manipulator. In such a way, the target assembly task can be performed, having the robot safely interacting with the surrounding environment.

The sensorless Cartesian impedance controller can be designed on the basis of the following manipulator dynamics [53]:

$$\mathbf{B}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) = \boldsymbol{\tau} - \mathbf{J}(\mathbf{q})^T \mathbf{h}_{ext}, \tag{1}$$

where $\mathbf{B}(\mathbf{q})$ is the robot inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the robot Coriolis vector, $\mathbf{g}(\mathbf{q})$ is the robot gravitational vector, $\boldsymbol{\tau}_f(\dot{\mathbf{q}})$ is the robot joint friction vector, $\mathbf{q}$ is the robot joint position vector, $\mathbf{J}(\mathbf{q})$ is the robot Jacobian matrix, and $\mathbf{h}_{ext}$ is the robot external force/torque vector, $\boldsymbol{\tau}$ is the robot joint torque vector.
Based on (1), it is possible to design the sensorless Cartesian impedance controller with dynamics compensation [53], defining the robot joint torque vector $\boldsymbol{\tau}$ as:

$$\boldsymbol{\tau} = \mathbf{B}(\mathbf{q})\boldsymbol{\gamma} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}), \tag{2}$$

where $\boldsymbol{\gamma}$ is the sensorless Cartesian impedance control law. Translational $\ddot{\mathbf{p}}$ and rotational $\ddot{\boldsymbol{\varphi}}_{cd}$ (described by the intrinsic Euler angles representation) acceleration components of the sensorless Cartesian impedance controller $\boldsymbol{\gamma}$ can be written as:

$$\begin{aligned} \ddot{\mathbf{p}} &= \mathbf{M}_t^{-1} \left(-\mathbf{D}_t \dot{\mathbf{p}} - \mathbf{K}_t \, \Delta\mathbf{p}\right), \\ \ddot{\boldsymbol{\varphi}}_{cd} &= \mathbf{M}_r^{-1} \left(-\mathbf{D}_r \, \dot{\boldsymbol{\varphi}}_{cd} - \mathbf{K}_r \, \boldsymbol{\varphi}_{cd}\right). \end{aligned} \tag{3}$$

Considering the translational part of the sensorless Cartesian impedance control, $\mathbf{M}_t$ is the mass matrix, $\mathbf{D}_t$ is the damping matrix, and $\mathbf{K}_t$ is the stiffness matrix. $\mathbf{p}$ is the actual Cartesian positions vector, while $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}^d$, where $\mathbf{p}^d$ is the target position vector. Concerning the rotational part of the sensorless Cartesian impedance control, $\mathbf{M}_r$ is the inertia matrix, $\mathbf{D}_r$ is the damping matrix, $\mathbf{K}_r$ is the stiffness matrix. $\boldsymbol{\varphi}_{cd}$ is the set of Euler angles extracted from $\mathbf{R}_c^d = \mathbf{R}_d^T \mathbf{R}_c$, describing the mutual orientation between the compliant frame $\mathbf{R}_c$ (at the end-effector) and the target frame $\mathbf{R}_d$.

Angular accelerations $\dot{\boldsymbol{\omega}}_{cd}$ can be computed considering the rotational part of the sensorless Cartesian:

$$\dot{\boldsymbol{\omega}}_{cd} = \mathbf{T}(\boldsymbol{\varphi}_{cd}) \left(\mathbf{M}_r^{-1} \left(-\mathbf{D}_r \, \dot{\boldsymbol{\varphi}}_{cd} - \mathbf{K}_r \, \boldsymbol{\varphi}_{cd}\right)\right) + \dot{\mathbf{T}}(\boldsymbol{\varphi}_{cd}) \dot{\boldsymbol{\varphi}}_{cd}, \tag{4}$$

where matrix $\mathbf{T}(\boldsymbol{\varphi}_{cd})$ defines the transformation from Euler angles derivatives to angular velocities $\boldsymbol{\omega}_{cd} = \mathbf{T}(\boldsymbol{\varphi}_{cd}) \dot{\boldsymbol{\varphi}}_{cd}$, and $\boldsymbol{\omega} = \mathbf{R}_{ee} \boldsymbol{\omega}_{cd}$ (with $\mathbf{R}_{ee}$ the rotation matrix from the robot base to its end-effector) [53]. By defining $\widetilde{\mathbf{M}}_r = \left(\mathbf{R}_{ee} \mathbf{T}(\boldsymbol{\varphi}_{cd})\right)^{-1} \mathbf{M}_r$ and $\widetilde{\mathbf{D}}_r = \mathbf{D}_r - \widetilde{\mathbf{M}}_r \mathbf{R}_{ee} \dot{\mathbf{T}}(\boldsymbol{\varphi}_{cd})$, (4) can be written as:

$$\dot{\boldsymbol{\omega}} = \widetilde{\mathbf{M}}_r^{-1} \left(-\widetilde{\mathbf{D}}_r \, \dot{\boldsymbol{\varphi}}_{cd} - \mathbf{K}_r \, \boldsymbol{\varphi}_{cd}\right). \tag{5}$$

The formulation resulting from (5), (4), and (3) can be written in a compact form as follows:

$$\ddot{\mathbf{x}}^{imp} = -\mathbf{M}^{-1} \left(\mathbf{D} \, \dot{\mathbf{x}} + \mathbf{K} \, \Delta\mathbf{x}\right), \tag{6}$$

where the target acceleration computed by the sensorless Cartesian impedance control is $\ddot{\mathbf{x}}^{imp} = [\ddot{\mathbf{x}}_t; \ddot{\mathbf{x}}_r] = [\ddot{\mathbf{p}}; \dot{\boldsymbol{\omega}}]$. $\mathbf{M} = [\mathbf{M}_t \, \mathbf{0}; \mathbf{0} \, \widetilde{\mathbf{M}}_r]$, $\mathbf{D} = [\mathbf{D}_t \, \mathbf{0}; \mathbf{0} \, \widetilde{\mathbf{D}}_r]$, $\mathbf{K} = [\mathbf{K}_t \, \mathbf{0}; \mathbf{0} \, \mathbf{K}_r]$ are the sensorless Cartesian impedance mass, damping and stiffness matrices composed by both the translational and rotational parts, and $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}^d = [\Delta\mathbf{p}; \boldsymbol{\varphi}_{cd}]$.

The sensorless Cartesian impedance control law $\boldsymbol{\gamma}$ can then be written as follows:

$$\boldsymbol{\gamma} = \mathbf{J}(\mathbf{q})^{-1}\left(\ddot{\mathbf{x}}^{imp} - \dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}\right). \tag{7}$$

In general, matrix $\mathbf{J}(\mathbf{q})^{-1}$ can be substituted with the pseudo-inverse of the Jacobian matrix $\mathbf{J}(\mathbf{q})^{\#}$ [54].

Substituting (2) in (1), under the hypothesis that the manipulator dynamics is known (such identification can be performed with state-of-the-art techniques [55]), the controlled robot dynamics results in:

$$\ddot{\mathbf{q}} = \boldsymbol{\gamma} - \mathbf{B}(\mathbf{q})^{-1}\mathbf{J}(\mathbf{q})^{T}\mathbf{h}_{ext}, \tag{8}$$

where $\mathbf{h}_{ext} = [\mathbf{f}, \mathbf{T}^{T}(\boldsymbol{\varphi}_{cd})\,\boldsymbol{\mu}^{d}]$ (considering the external forces $\mathbf{f}$ and the external torques $\boldsymbol{\mu}^{d}$ - referred to the target frame $\mathbf{R}_{d}$ - acting on the robot related to the interaction with the surrounding environment). Substitution of (7) into (8) leads to:

$$\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} = \ddot{\mathbf{x}} = \ddot{\mathbf{x}}^{imp} - \mathbf{J}(\mathbf{q})\mathbf{B}(\mathbf{q})^{-1}\mathbf{J}(\mathbf{q})^{T}\mathbf{h}_{ext}, \tag{9}$$

having $\ddot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ the resulting Cartesian acceleration of the robot end-effector resulting from the implementation of the proposed sensorless Cartesian impedance controller.
Finally, substituting (6) into (9), the controlled robot dynamics resulting from the design of the sensorless Cartesian impedance control is described by the following equation:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \mathbf{K}\Delta\mathbf{x} = -\overline{\mathbf{H}}(\mathbf{q})\mathbf{h}_{ext}, \tag{10}$$

where $\overline{\mathbf{H}}(\mathbf{q}) = \mathbf{M}\mathbf{J}(\mathbf{q})\mathbf{B}(\mathbf{q})^{-1}\mathbf{J}(\mathbf{q})^{T}$.

Remark 3. The resulting controlled robot dynamics (as defined by (10)) is therefore coupled in the DoFs by the matrix $\overline{\mathbf{H}}(\mathbf{q})$. This will affect the task execution. Such coupled behavior has to be compensated in order to correctly perform the target task.

Remark 4. It has to be underlined that the only difference between sensorless Cartesian impedance control and sensor-based impedance control results from (10), where the coupling of the DoFs is not present for the sensor-based controller.

Remark 5. It has to be underlined that the proposed learning approach described in the following can be applied in the same way to a sensorless position control-based impedance control for the robots not equipped with torque control.

3.1. Redundancy management

The Franka EMIKA panda manipulator has been used as a test platform for experimental validation. Such a robot is redundant, requiring to manage its null-space configuration while performing the main task. In this paper, a pure damping behavior is proposed for the null-space configuration control, aiming to damp the null-space motion:

$$\boldsymbol{\tau}_{R} = \mathbf{B}(\mathbf{q})\left(\left(\mathbf{I} - \mathbf{J}(\mathbf{q})^{\#}\mathbf{J}(\mathbf{q})\right)(-\mathbf{D}_{n}\dot{\mathbf{q}})\right), \tag{11}$$

where $\boldsymbol{\tau}_{R}$ is the null-space control torque, $\mathbf{I}$ is the identity matrix, $\mathbf{J}(\mathbf{q})^{\#}$ is the pseudo-inverse of the Jacobian matrix, and $\mathbf{D}_{n}$ is the null-space damping diagonal matrix. The term $\left(\mathbf{I} - \mathbf{J}(\mathbf{q})^{\#}\mathbf{J}(\mathbf{q})\right)$ is the null-space projection matrix. The term $-\mathbf{D}_{n}\dot{\mathbf{q}}$ allows to damp the null-space motion.

The control law (2) is, therefore, modified as follows:

$$\boldsymbol{\tau} = \mathbf{B}(\mathbf{q})\boldsymbol{\gamma} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}_{f}(\dot{\mathbf{q}}) + \boldsymbol{\tau}_{R}. \tag{12}$$

The control torque $\boldsymbol{\tau}_{R}$ acts in the null-space of the manipulator, *i.e.*, not affecting the Cartesian motion of the robot. Indeed, the Cartesian controlled robot behavior in (10) is not affected by this term.

4. Nominal reference task trajectory learning

In this Section, the nominal reference task trajectory learning methodology is described. In this phase, the robot is controlled by means of the gravity compensation mode (*i.e.*, allowing the human to move the robot in the free space). A few human's demonstrations of the target task are performed, recording the robot end-effector Cartesian positions $\mathbf{x}$ and angular velocities $\boldsymbol{\omega}$ for each subtask separately. While the positions $\mathbf{x}_{t}$ are directly measured and recorded, the end-effector Cartesian rotation matrix is recorded. The Euler angles are then extracted in order to compute the related derivatives $\dot{\boldsymbol{\varphi}}$. The angular velocities are finally computed on the basis of the formulation $\boldsymbol{\omega} = \mathbf{T}(\boldsymbol{\varphi})\dot{\boldsymbol{\varphi}}$ [53]. The records are processed (for each demonstration) by a HMM algorithm to select most consistent demonstrated task trajectory (Fig. 2). In fact, each demonstration includes teaching uncertainties and the proposed HMM approach allows to select the most reliable one to limit the presence of teaching uncertainties. The HMM algorithm is applied to each Cartesian DoF separately in order to limit the influence of the uncertainties of a specific DoF on the other ones. The complete trajectory is then recomposed to be executed by the manipulator. Exploiting the proposed approach, it is possible to transfer the human's task knowledge to the manipulator, making it able to learn the target application. In fact, in addition to the task kinematics, the recorded data include important task knowledge, such as workspace constraints, obstacle avoidance requirements, workpiece approaching strategies, and subtask definition. The proposed approach is shown in Fig. 3, showing the implemented data processing procedure and algorithms.

Remark 6. It has to be underlined that the recording and processing of the robot end-effector Cartesian position $\mathbf{x}_{t}$ and angular velocities $\boldsymbol{\omega}$ has been proposed instead of the robot joint trajectories $\mathbf{q}$. In fact, processing each joint trajectory independently results in a deviation of the Cartesian end-effector trajectory from the human's demonstrated one in the final motion reconstruction. Taking as a reference the Cartesian DoFs allows to preserve the required task information.

4.1. Hidden Markov model-based methodology

Following the proposed methodology detailed in Fig. 3, in order to apply the HMM approach for the selection of the reference task trajectory on the basis of a single human's demonstrations, some preliminary data (*i.e.*, the robot end-effector Cartesian positions $\mathbf{x}_{t}$ and angular velocities $\boldsymbol{\omega}$) recording and pre-processing steps are required:

- human's demonstrations data recording [56] (*Observed trajectories* block in Fig. 3): the human guides the robot (in gravity compensation control mode) to perform the task. The robot end-effector Cartesian positions $\mathbf{x}_{t}(t)$ and angular velocities $\boldsymbol{\omega}(t)$ are recorded for each time-instant t. The recording time is fixed, according to the task execution time requirements. Each trajectory is then divided in 32 sub-intervals with equal duration for the next processing steps of the methodology;
- Short Time Fourier transformation [57] of the recorded data (*Short Time Fourier Transformation* block in Fig. 3): each end-effector Cartesian position $\mathbf{x}_{t}$ and angular velocity $\boldsymbol{\omega}$ shifts into the frequency domain for each sub-interval;
- LBG clustering [58] (*LBG clustering* block in Fig. 3): it is used to partition a set of input data into a number of clusters, generating a codebook, exploiting the frequency domain data.

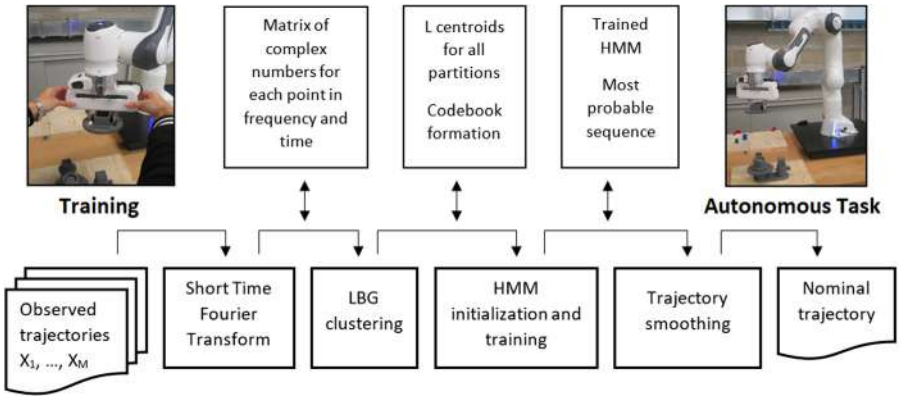


Fig. 3. Reference task trajectory learning methodology: on the basis of a few human's demonstrations of the target task, the most consistent demonstrated task trajectory is selected for autonomous execution and optimization. The procedure start with the training of phase, where the task is teached by the human, and it ends with the robot capable to reproduced the learned task.

After the recording and pre-processing of the data, the HMM Bakis Left-Right algorithm [51] is implemented to select the nominal reference task trajectory over the demonstrated ones. The proposed HMM algorithm (*HMM initialization and training* block in Fig. 3) is characterized by the following mathematical model:

$$\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}), \tag{13}$$

where $\mathbf{A}$ is the state transition probability matrix, $\mathbf{B}$ is the observation probability matrix, and $\boldsymbol{\pi}$ is the initial state distribution. Especially, the state transition probability matrix and the observation probability matrix are randomly initialized. Furthermore, the initial state distribution $\boldsymbol{\pi}$ is initialized as follows:

$$\pi_j = \begin{cases} 0, & j \neq 1; \\ 1, & j = 1. \end{cases} \tag{14}$$

where $j = 1 : N$ is the observation index incrementally increased, and N is the total number of observations.

The model in (13) is then trained on the set $\mathbf{O}$, which is formed by the observation sequences, exploiting the Expectation–Modification (EM) method [59]. Afterwards, the HMM algorithm is applied to select the most consistent demonstrated trajectory, evaluating the probability of a sub-interval m to be in a HMM state x^p_{HMM}, while the next sub-interval $m + 1$ is in a state x^j_{HMM} (where j and p denote the generic HMM state, *i.e.* the generic frequency cluster). Each robot end-effector Cartesian DoF is elaborated separately. The trajectory selection is performed on the basis of the likelihood index calculation [60] for each demonstrated one. The trajectory showing the maximum likelihood index is considered to be the most consistent one, and it will be selected as the nominal reference trajectory.

Once the reference task trajectory has been selected over the demonstrated ones by the proposed HMM, a post-processing step (*Trajectory smoothing* block in Fig. 3) is additionally implemented in order to improve the autonomous trajectory execution performance. The purpose of this post-processing step is to find a good fit of the given path, limiting the trajectory accelerations (*i.e.*, avoiding dangerous discontinuities and satisfying limits). The selected criteria is the *cubic smoothing spline*. A complete description is provided by [61,62]. The method defines the smoothing of the selected trajectories according to a user-defined parameter s. For the purpose of this work, $s = 0.5$ has been imposed, so that a balance between the preserving the trajectories information and their smoothing is achieved. In fact, in the case $s = 0$, the characteristic shape of the curve is lost, while in the case $s = 1$, no smoothing of the trajectory is performed.

Finally, the complete nominal reference task trajectory is recomposed, generating the robot Cartesian motion by considering all the 6 Cartesian DoFs above elaborated by the proposed methodology (translational and rotational DoFs). Such nominal learned trajectory defines the sensorless Cartesian impedance setpoint $\mathbf{x}^d(t)$ in (6) for the assembly task execution.

Remark 7. It has to be underlined that the proposed methodology is applied to each subtask separately. Therefore, subtasks trajectories are independently recorded, elaborated and recomposed into the sensorless Cartesian impedance setpoint $\mathbf{x}^{d,\#k}(t)$, where $\#k$ denotes the considered subtask in a defined sequence of the assembly task.

Remark 8. It is important to note that, since the gravity compensation mode is exploited to control the robot in this phase, there is no need of external forces/torques measurements. The proposed HMM approach can be applied to both sensor-based or sensorless robots as presented in this Section.

5. Task optimization

In this Section, the optimization algorithm for task-performance maximization and uncertainties compensation is described. The optimization is performed for each subtask separately, by defining a cost function $J^{\#k}_{BO}$ (where $\#k$ identifies the specific subtask) maximizing the task performance (*i.e.*, avoiding task failures, limiting interaction forces and guaranteeing the execution time). Relying on the Bayesian Optimization (BO) [63], the proposed methodology allows to optimize both the nominal learned reference trajectory (*i.e.*, the sensorless Cartesian impedance setpoint $\mathbf{x}^{d,\#k}$, which is computed as described in Section 4) and the sensorless Cartesian impedance control parameters (*i.e.*, the stiffness and damping parameters $\mathbf{K}^{\#k}$ and $\mathbf{D}^{\#k}$). The cost function is evaluated for each experimental iteration of the approach in order to perform the optimization (Fig. 2). In addition, task uncertainties (*e.g.*, parts positioning, demonstrations uncertainties - *i.e.*, residual noise in the nominal reference trajectory) are compensated in such optimization approach. One of the most important effect to be compensated is the coupled behavior resulting from the sensorless Cartesian impedance controller (10), affecting the execution of the nominal learned reference trajectory.

5.1. Cost function definition

On the basis of the subtasks definition in Section 2, a specific optimization can be performed for each subtask (*i.e.*, specifically defining a cost function for each subtask). Considering the *approaching subtask*, it is considered that an optimization is not

required since such subtask only aims to pre-positioning the manipulated part in the correspondence of the assembly location. In addition, no information are available to perform an optimization since no external sensors (*e.g.*, vision systems) are involved in the proposed application. Considering the *insertion subtask* and the *pushing subtask*, an optimization has instead to be performed in order to maximize task performance and compensate for task uncertainties (*e.g.*, position of the parts/assembly location w.r.t. the robot reference frame).

5.1.1. Insertion subtask optimization

The optimization of the *insertion subtask* consists in being able to perform the initial insertion of the part in the reference assembly location. Such assembly subtask is the most critical one, since a failure in performing the first insertion of the part will result in a task failure. In addition, such subtask is characterized by the highest position uncertainties in the parts/assembly location w.r.t. the robot reference frame. Therefore, it is of fundamental importance to being able to optimize such subtask while compensating for assembly uncertainties.

In order to perform the optimization of the insertion subtask, the following cost function $J_{BO}^{\#2}$ (to be maximize) can be defined:

$$J_{BO}^{\#2} = -\left(G_{xy}^{\#2} e_{xy}^{\#2} + G_z^{\#2} e_z^{\#2} + G_\psi^{\#2} e_\psi^{\#2} + G_f^{\#2} e_f^{\#2} + G_L^{\#2} L^{\#2}\right). \tag{15}$$

The term $e_{xy}^{\#2}$ in (15) is computed as the maximum value of the positioning error in the x–y plane:

$$e_{xy}^{\#2} = \max\left(\sqrt{\left(x_x^{d,\#2}(t) - x_x(t)\right)^2 + \left(x_y^{d,\#2}(t) - x_y(t)\right)^2}\right). \tag{16}$$

$x_x^{d,\#2}(t)$ and $x_y^{d,\#2}(t)$ are the sensorless Cartesian impedance control setpoint x and y coordinates. The Cartesian robot end-effector positions $x_x(t)$ and $x_y(t)$ are measured during the experimental task execution. Such term allows to compensate for misalignments related to parts/assembly location position uncertainties and to demonstrations uncertainties related to the approaching subtask.

The term $e_z^{\#2}$ in (15) is computed as the maximum value of the positioning error along the z DoF:

$$e_z^{\#2} = \max\left(\mid x_z^{d,\#2}(t) - x_z(t) \mid\right). \tag{17}$$

$x_z^{d,\#2}(t)$ is the sensorless Cartesian impedance control setpoint z coordinate. The Cartesian robot end-effector position $x_z(t)$ is measured during the experimental task execution. Such term allows to compensate for misalignments related to parts/assembly location position uncertainties and to interference effects of the parts.

The term $e_\psi^{\#2}$ in (15) is computed as the maximum value of the rotational error along the z axis:

$$e_\psi^{\#2} = max\left(\mid x_\psi^{d,\#2}(t) - x_\psi(t) \mid\right). \tag{18}$$

$x_\psi^{d,\#2}(t)$ is the sensorless Cartesian impedance control setpoint for the rotational DoF ψ (rotation about the z axis). The Cartesian robot end-effector position $x_\psi(t)$ is measured during the experimental task execution. Such term allows to compensate for misalignments related to parts/assembly location orientation uncertainties.

The term $e_f^{\#2}$ in (15) is computed as the maximum value of the estimated interaction force:

$$e_f^{\#2} = \max\left(\sqrt{f_x(t)^2 + f_y(t)^2 + f_z(t)^2}\right). \tag{19}$$

The ith estimated interaction force is computed as:

$$f_i(t) = K_i^{\#2}\left(x_i^{d,\#2}(t) - x_i(t)\right). \tag{20}$$

Such term allows to avoid too strong interaction between the robot and the environment, avoiding for safety emergency stop and/or task failures.

The term $L^{\#2}$ in (15) is introduced to penalize subtask failures. Such failures are related to safety emergency stop of the robot (such as extra interaction force/torque identified by the robot controller). In addition, failures are related to the impossibility in finding the insertion location. Such failure is identified by the monitoring of the term $e_z^{\#2}$. If such error increases over a specified threshold $\bar{e}_z^{\#2}$, the subtask is considered as failed. In fact, this means that the manipulated part has not be properly inserted in the target location. In the case of a failure, the optimization is stopped and penalized, the robot is re-positioned to the starting subtask position and the next optimization iteration is started. The penalty $L^{\#2}$ is imposed as:

$$L^{\#2} = e^{-t/T^{\#2}} \quad \text{if} \quad e_z^{\#2} > \bar{e}_z^{\#2}, \tag{21}$$

where $T^{\#2}$ is the target subtask execution time.

The terms $G_{xy}^{\#2}$, $G_z^{\#2}$, $G_\psi^{\#2}$, $G_f^{\#2}$, and $G_L^{\#2}$ are gains related to the specific cost function term. Such gains can be tuned in order to weight the different contributions in the cost function. In this paper, these gains have been experimentally tuned.

Remark 9. In order to achieve a more accurate estimation of the interaction force in (20), a sensorless methodology, as proposed in [64,65], can be implemented.

5.1.2. Pushing subtask optimization

The optimization of the *pushing subtask* consists in being able to finalize the assembly task (*e.g.*, a pushing force is required to complete a mechanical fixing of the part on a reference surface).

In order to perform the optimization of the pushing subtask, the following cost function $J_{BO}^{\#3}$ (to be maximize) can be defined:

$$J_{BO}^{\#3} = -\left(G_z^{\#3} e_z^{\#3} + G_\psi^{\#3} e_\psi^{\#3} + G_f^{\#3} e_f^{\#3} + G_L^{\#3} L^{\#3}\right). \tag{22}$$

The term $e_z^{\#3}$ in (22) is computed similarly to $e_z^{\#2}$ in (17). Such term allows to avoid any stacking behavior during the subtask execution.

The term $e_\psi^{\#3}$ in (22) is computed similarly to $e_\psi^{\#2}$ in (18). Such term allows to compensate for misalignments related to parts/assembly location orientation uncertainties and/or for parts engagement.

The term $e_f^{\#3}$ in (22) is computed as the maximum value of the force error along the vertical direction:

$$e_f^{\#3} = \max\left(\mid f_z^d - f_z(t)^2 \mid\right). \tag{23}$$

where the estimated interaction force $f_z(t)$ is computed as in (20) and f_z^d is the assembly reference force. Such term allows to finalize the assembly, making the robot applying a reference force.

The term $L^{\#3}$ in (22) is computed similarly to $L^{\#2}$ in (21), considering the subtask execution time $T^{\#3}$. Such terms allows to penalize task failures.

The terms $G_z^{\#3}$, $G_\psi^{\#3}$, $G_f^{\#3}$, and $G_L^{\#3}$ are gains related to the specific cost function term. Such gains can be tuned in order to weight the different contributions in the cost function. In this paper, these gains have been experimentally tuned.

Remark 10. It has to be underlined that cost functions (15) and (22) are user-defined. Therefore, such definitions can be modified in order to include different terms and metrics for the assembly optimization.

5.2. Bayesian optimization

On the basis of the controller structure described in Section 3, the cost functions $J_{BO}^{\#k}$ in Section 5.1 allow to tune the target controller parameters (for the specific subtask k), *i.e.*, the sensorless Cartesian impedance control setpoint $\mathbf{x}^{d,\#k}$ and the stiffness and damping parameters of $\mathbf{K}^{\#k}$ and $\mathbf{D}^{\#k}$.

Let us consider the generic cost function J and its related design parameters. By collecting all these design parameters in a vector $\boldsymbol{\theta}$, the tuning task reduces to the minimization of the cost $J(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$, within a space of admissible values $\boldsymbol{\Theta}$. However, a closed-form expression of the cost J as a function of the design parameter vector $\boldsymbol{\theta}$ is not available. Furthermore, this cost cannot be evaluated through numerical simulations as the robot dynamics are assumed to be partially unknown. Instead, it is possible to perform experiments on the robot and measure the cost J_i achieved for a given controller parameter vector $\boldsymbol{\theta}_i$, and thus run an optimization algorithm driven by measurements of J. Nonetheless, the peculiar nature of the optimization problem at hand restricts the class of applicable optimization algorithms. Indeed,

(*i*) the measured cost J_i consists in a noisy observations of the "true" cost function, namely $J_i = J(\boldsymbol{\theta}_i) + n_i$, with n_i denoting measurement noise and possibly intrinsic process variability;

(*ii*) no derivative information is available;

(*iii*) there is no guarantee that the function $J(\boldsymbol{\theta})$ is convex;

(*iv*) function evaluations may require possibly costly and time-consuming experiments on the robot.

Features (*i*), (*ii*) and (*iii*) rule out classical gradient-based algorithms and restrict us to the class of gradient-free, global optimization algorithms. Within this class of algorithms, *Bayesian optimization* (BO) is generally the most efficient in terms of number of function evaluations [63,66] and it is thus the most promising approach to deal with (*iv*).

In BO, the cost J is simultaneously learnt and optimized by sequentially performing experiments on the robot. Specifically, at each iteration i of the algorithm, an experiment is performed for a given controller parameter $\boldsymbol{\theta}_i$ and the corresponding cost J_i is measured. Then, all the past parameter-cost observations $\mathcal{D}_i = \{(\boldsymbol{\theta}_1, J_1), (\boldsymbol{\theta}_2, J_2), \ldots, (\boldsymbol{\theta}_i, J_i)\}$ are processed and a new parameter $\boldsymbol{\theta}_{i+1}$ to be tested at the next experiment is computed according to the approach discussed in the following. Additional details related to *surrogate model*, *acquisition function*, and *algorithm outline* used for Bayesian Optimization can be found in [52].

Remark 11. It has to be underlined that, since the assembly scenario is considered (partially) unknown (*i.e.*, uncertainties on assembly location and parts geometry/positioning), the proposed (re)optimization procedure will be the same for sensor-based robots. The only difference between sensorless and sensor-based optimization is in the possibility to use forces/torques measurements in (19) and in (23), instead of the estimation provided by (20). In fact, without accurate information on the parts geometry and positioning, even sensor-based controllers cannot perform assembly tasks in a totally blind way [27]. Such limitations can be overcome by means of vision systems and machine vision algorithms. Such a solution, however, results in increased costs, setup time, and computational time, in addition requiring an *ad hoc* (re) design of the robotic cell to provide proper light-conditions and avoidance of occlusions between the vision system and the assembly location.

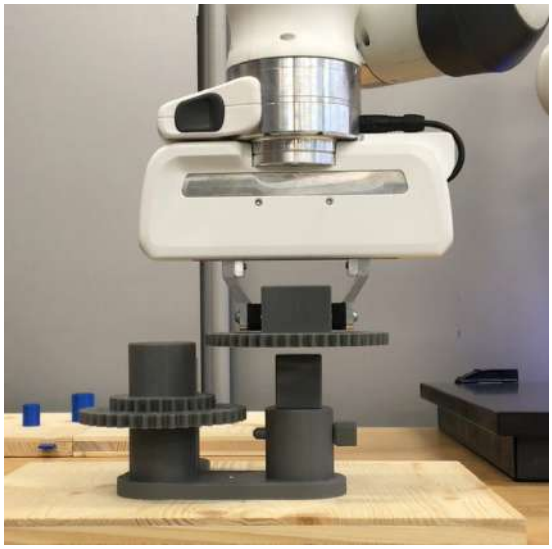


Fig. 4. Target assembly task. The Franka EMIKA panda robot grasps and assembles the target gear into its square shaft.

6. Experimental results

In this Section, the experimental validation of the proposed methodology is performed. The proposed reference assembly task is described. Achieved results applying the proposed HMM algorithm and the proposed BO algorithm are given separately, in order to better highlight specific performance. The proposed HMM+BO procedure has been compared with the HMM procedure without optimization, in order to highlight the improved performance provided by the BO algorithm, capable to compensate for task uncertainties.

Remark 12. A video of the proposed methodology (including the HMM algorithm for task learning and the BO algorithm for task optimization) is available at https://www.youtube.com/watch?v=ZCT56C2Gb_8&t=108s.

6.1. Reference assembly task description

In order to validate the proposed methodology, an assembly of a gear into its square shaft has been considered (Fig. 4). Parts are positioned in the robot working area with a tolerance of ± 7.5 mm. Tight assembly tolerance H7/h6 is shown for the installation of the gear in its shaft. The task is considered successful if the gear inserted in its shaft, making it engaging with the already installed gear. As detailed in Section 2, three main subtasks can be identified for the target assembly: an approaching subtasks, an insertion subtask, and a pushing subtask. The learning and optimization description and results of such phases are detailed in the next Sections.

The robotic platform involved in the experiments is a Franka EMIKA panda manipulator. Exploiting its model-based torque control mode with gravity compensation (control frequency 1 kHz), the sensorless Cartesian impedance controller in Section 3 has been implemented. Friction compensation has been implemented as in [67].

The HMM algorithm in Section 4 has been performed exploiting the Hidden Markov Model (HMM) Matlab Toolbox [60].

The BO algorithm in Section 5 has been performed exploiting the c++ *limbo* [68] and the *NLopt* [69] libraries.

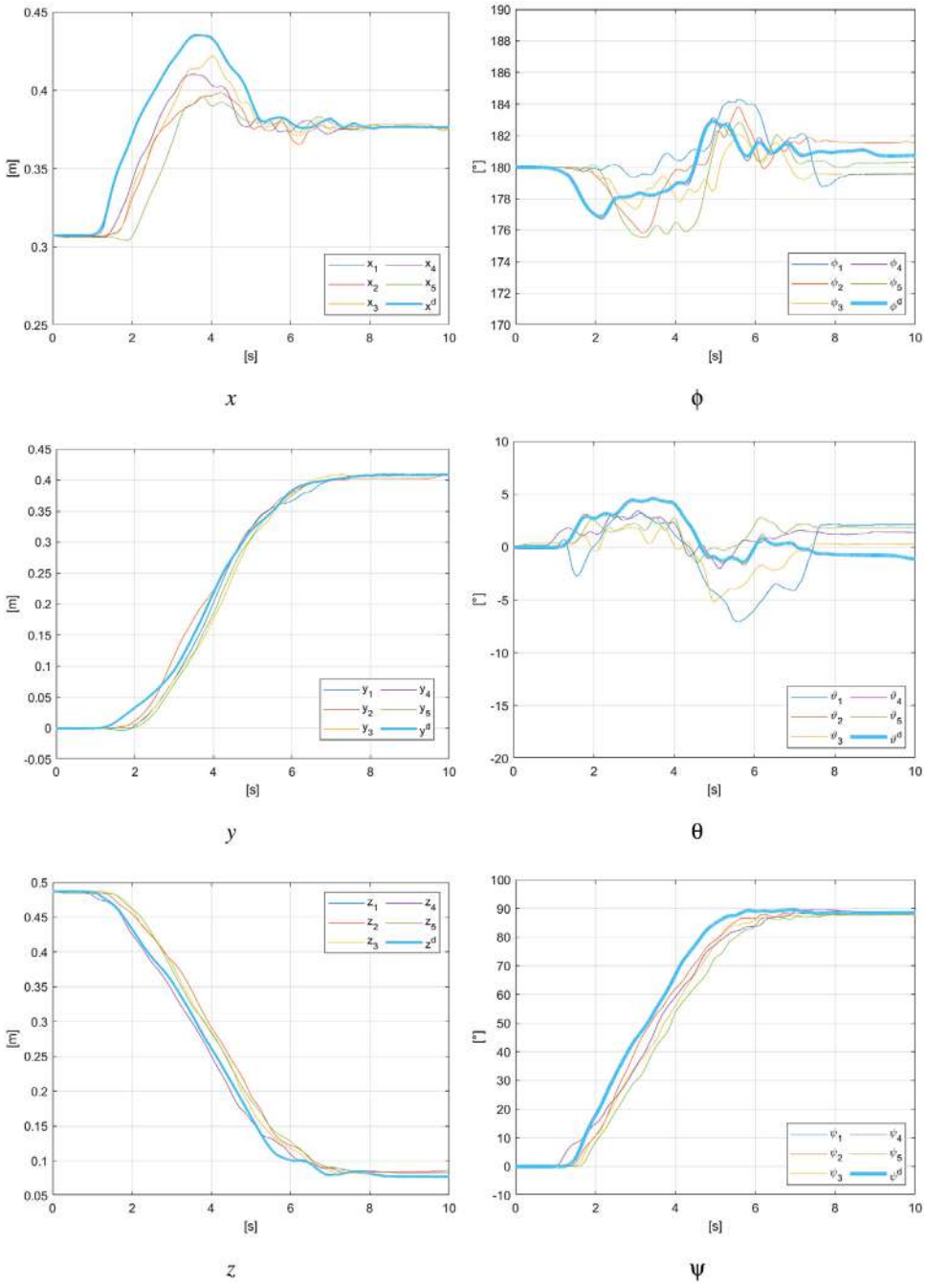


Fig. 5. The 5 subjects' demonstrations are shown for the Cartesian translational DoFs x, y, z and for the Cartesian rotational DoFs ϕ, θ, ψ. The computed nominal reference trajectory learned exploiting the proposed HMM methodology is highlighted for all the DoFs.

6.2. Nominal reference task trajectory learning validation description and results

In order to evaluate the proposed nominal reference task trajectory learning methodology (described in Section 4), 15 subjects have been involved in the experimental validation. Prior to testing, all subjects have been informed about the evaluation scenario and the testing procedure. In particular, each subject had to perform each subtask (described in Section 2) 5 times in order to record the robot end-effector Cartesian pose trajectory (sampling frequency 1 kHz). The subject manually guided the robot (controlled in gravity compensation mode) to perform each subtask. After the recording is performed, each subtask has been processed by the proposed methodology (described in Section 4) in order to select the most consistent subtask trajectory.

Remark 13. It has to be underlined that each subject is treated independently w.r.t. the other subjects, meaning that 15 different teached task trajectories will be available, one for each subject.

By applying the learning HMM methodology described in Section 4, the trajectory shown in Fig. 5 is selected. Results are shown for one subject, considering the approaching subtask. Results related to other subjects and other subtasks (*i.e.*, insertion and pushing subtasks) are similar. The proposed HMM algorithm is capable to select the most consistent trajectory for each Cartesian DoF independently, allowing to select the most reliable demonstrated task trajectory. In particular, the selected trajectory is

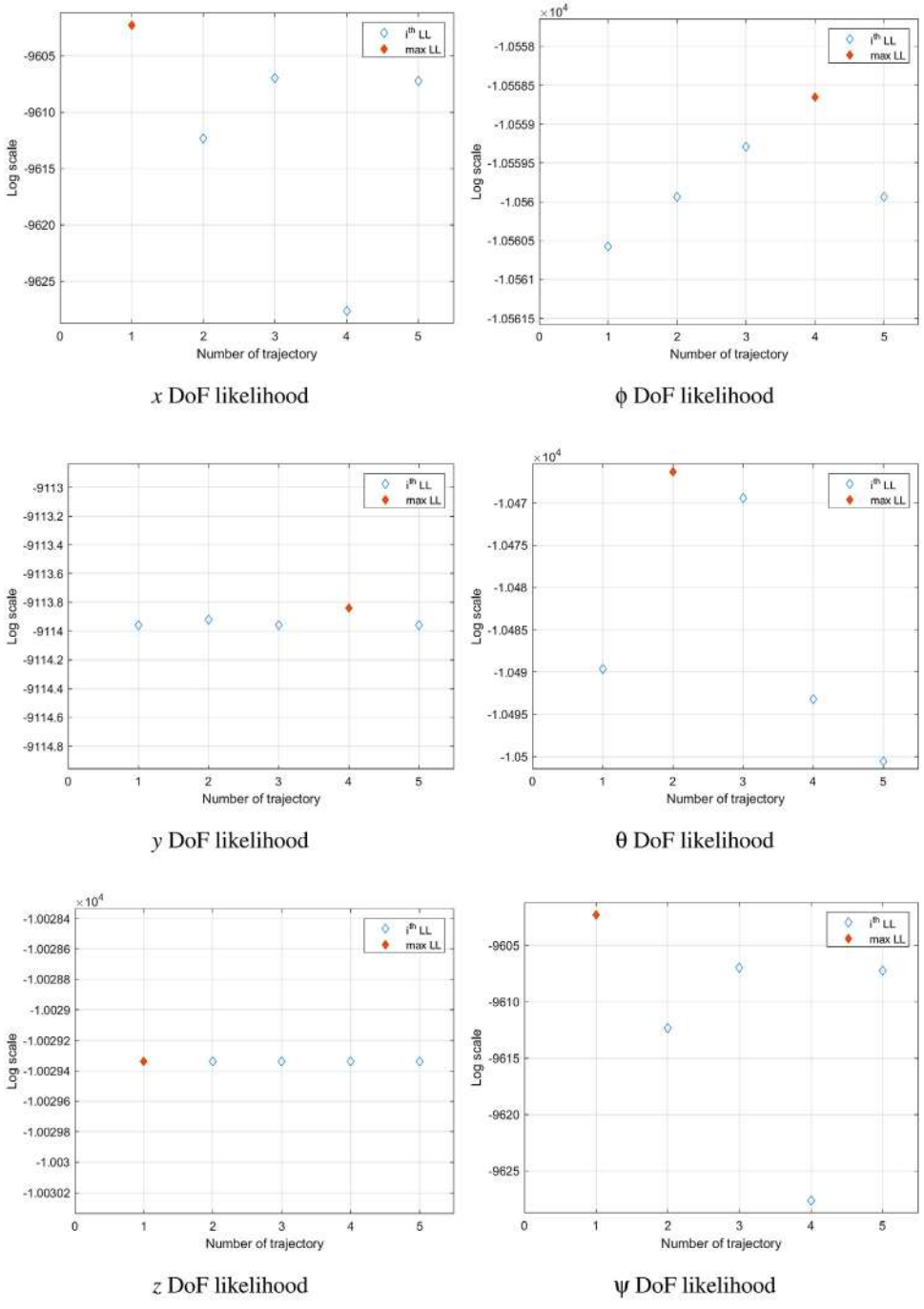


Fig. 6. Likelihood values for the demonstrated trajectories (for each DoF) considering the approaching subtask.

Fig. 7. HMM algorithm applied to the learning of the three subtasks: approaching subtask #1, insertion subtask #2, pushing subtask #3.

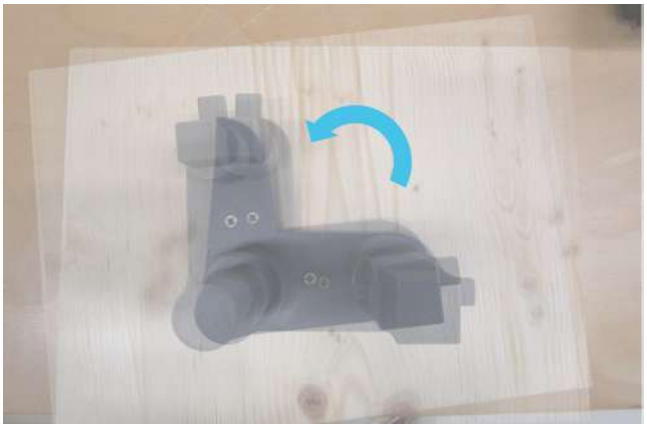


Fig. 8. Simulated task assembly location pose uncertainties for the evaluation of the BO algorithm capabilities.

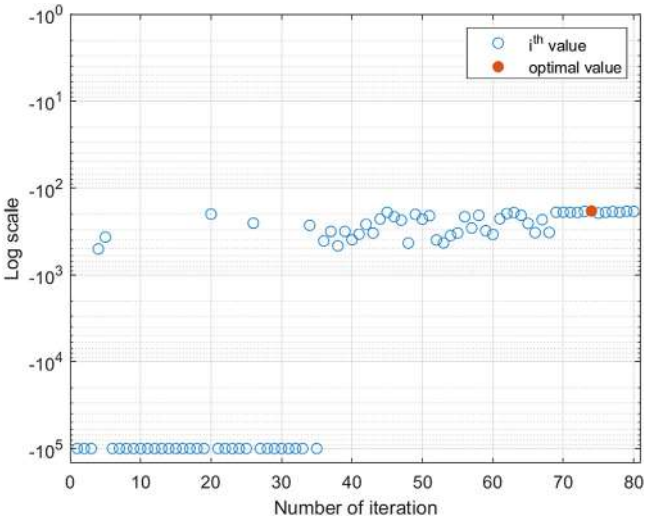


Fig. 9. Cost function $J_{BO}^{\#2}$ values over the experimental iterations for the first sub-optimization of the insertion subtask. The best iteration is marked by the red circle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the one which shows the maximum likelihood index computed by the HMM methodology, and it is highlighted in the plots. Likelihood values are shown in Fig. 6 for the approaching subtask. Similar results are obtained for the insertion and pushing subtasks. The complete robot motion is finally recomposed for each subtask.

Considering the approaching subtask, the selected nominal reference trajectory $\mathbf{x}^{d,\#1}(t)$ is executed to positioning the robot in the correspondence of the assembly location. Considering the insertion subtask, the final pose $\mathbf{x}_f^{d,\#2}$ is extracted from the selected nominal reference trajectory to be used in the BO algorithm for the optimization of this subtask. $\mathbf{x}_f^{d,\#2}$ will be, in fact, used to define the optimization variables ranges to perform the insertion subtask optimization. Considering the pushing subtask, the final pose $\mathbf{x}_f^{d,\#3}$ is extracted from the selected nominal reference trajectory to be used in the BO algorithm for the optimization of this subtask. $\mathbf{x}_f^{d,\#3}$ will be, in fact, used to define the optimization variables ranges to perform the pushing subtask optimization.

Fig. 7 shows the experiments related to the nominal reference task trajectory learning, for each subtask, applying the proposed HMM algorithm.

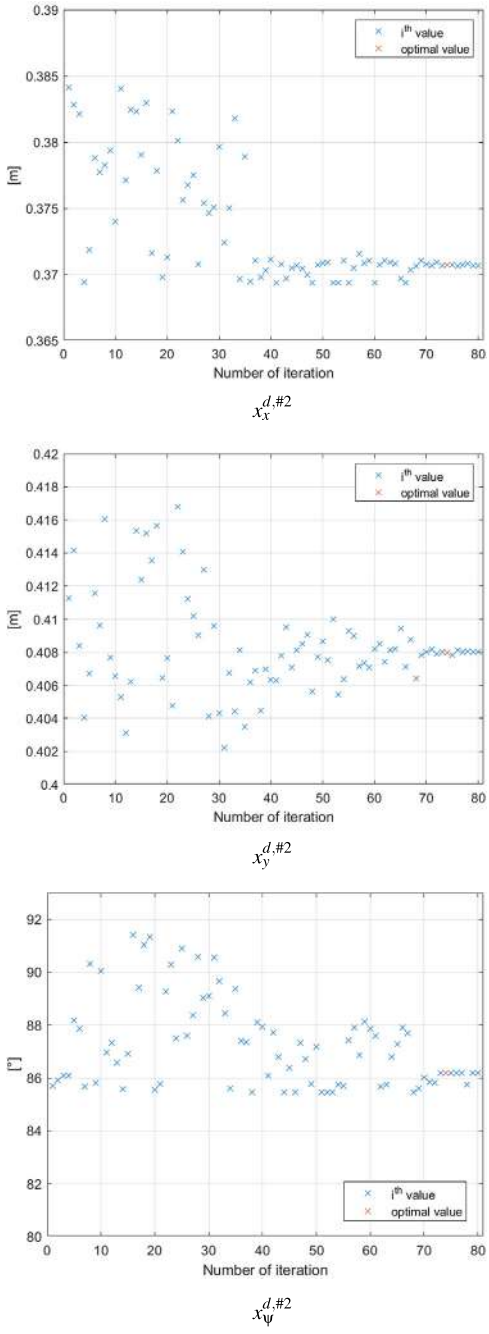


Fig. 10. $x_x^{d,\#2}$, $x_y^{d,\#2}$ and $x_\psi^{d,\#2}$ DoFs optimization related to the first sub-optimization of the insertion subtask. The best iteration is marked in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Remark 14. Results related to one subject are shown, considering the approaching subtask #1. Results related to the other 14 subjects and to the other two subtasks are similar.

6.3. Task optimization validation description and results

In order to perform the assembly task optimization, the proposed BO algorithm described in Section 5 has been implemented. After the task trajectories are learned independently for all the subjects (*i.e.*, 15 independent teached trajectories will be available after the demonstration phase), an optimization has been

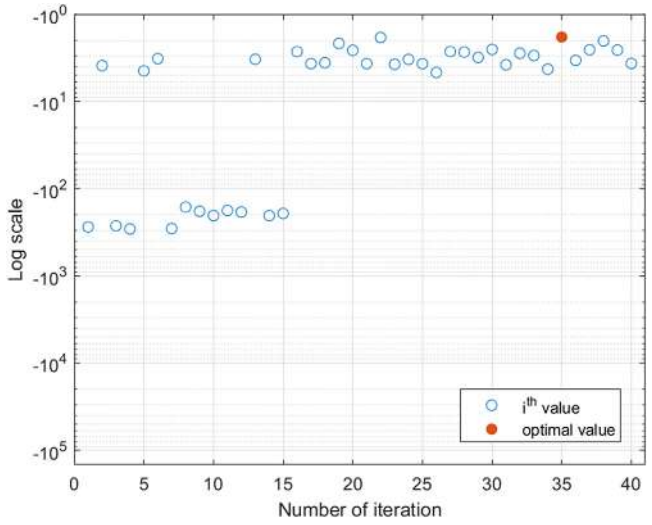


Fig. 11. Cost function $J_{BO}^{\#2}$ values over the experimental iterations for the second sub-optimization of the insertion subtask. The best iteration is marked by the red circle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

performed independently for each of the learned task trajectories. While the demonstrations of the task trajectories has been performed in nominal working conditions, the optimization has been performed modifying the assembly scenario (*i.e.*, modifying the assembly location pose differently for each optimization execution in the range of ±7.5 mm for translational DoFs, and ±5° for rotation; Fig. 8). In such a way, it is possible to evaluate the capabilities of the proposed approach to compensate for task uncertainties.

Each subtask has been independently optimized as described in Section 5, exploiting the specific cost function. In particular, the approaching subtask has not been considered for optimization, and it is executed exploiting the nominal learned task trajectories. Insertion subtask and pushing subtask have been instead optimized.

6.3.1. Insertion subtask optimization

Exploiting the cost function $J_{BO}^{\#2}$ proposed in (15), the insertion subtask can be optimized. In particular, the proposed cost function allows for the exploration and optimization of the assembly pose (*i.e.*, the sensorless Cartesian impedance control setpoint $\mathbf{x}^{d,\#2}$ to find the assembly insertion location), together with the control parameters (*i.e.*, stiffness and damping parameters of the sensorless Cartesian impedance control $\mathbf{K}^{\#2}$ and $\mathbf{D}^{\#2}$). Since the insertion subtask is the most critical assembly phase, its optimization has been divided in two sub-optimizations.

Remark 15. Results related to one optimization procedure are shown. Results related to the other 14 optimizations are similar.

The first sub-optimization allows to optimize the $x_x^{d,\#2}$, $x_y^{d,\#2}$ translational DoFs and the $x_\psi^{d,\#2}$ rotational DoF (rotation about the z axis). The considered ranges of the optimization variables are: $x_{f,x}^{d,\#2} \pm 7.5$ mm for the x DoF, $x_{f,y}^{d,\#2} \pm 7.5$ mm for the y DoF, and $x_{f,\psi}^{d,\#2} \pm 5°$ for the ψ DoF. The proposed ranges exploit the Cartesian pose $\mathbf{x}_f^{d,\#2}$ selected by the HMM in Section 6.2. In such a sub-optimization, it is possible to find the correct assembly location to perform the insertion.

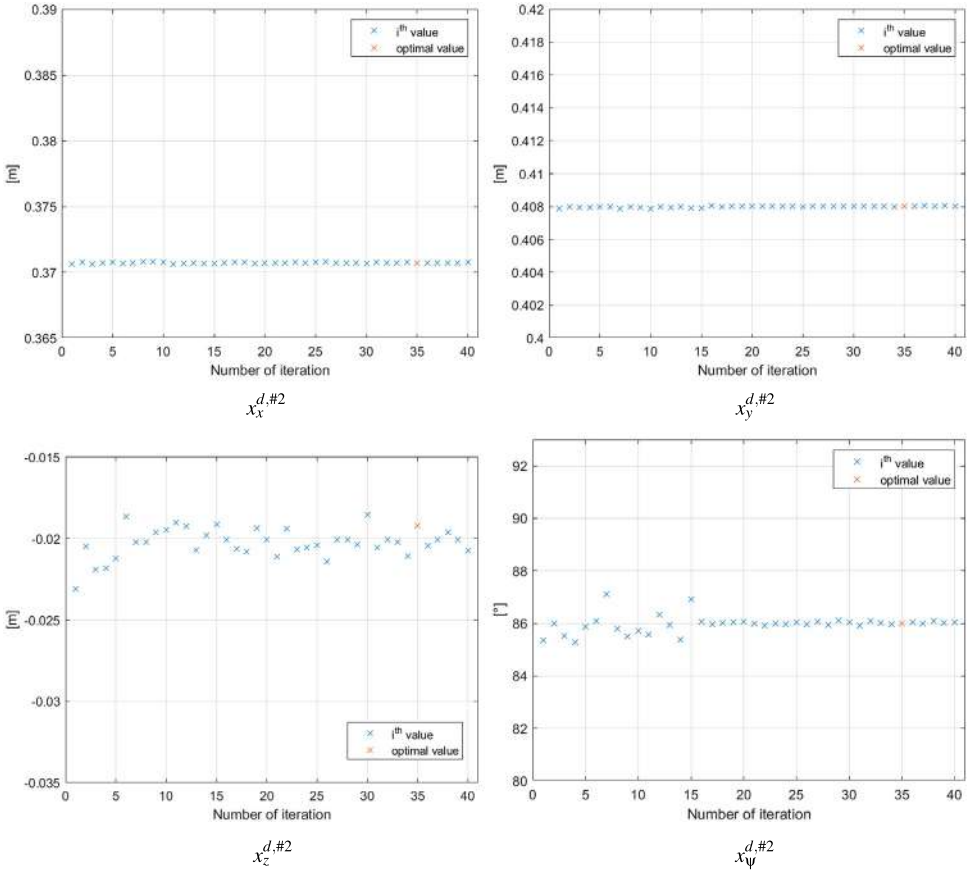


Fig. 12. $x_x^{d,\#2}$, $x_y^{d,\#2}$, $x_z^{d,\#2}$ and $x_\psi^{d,\#2}$ DoFs optimization related to the second sub-optimization of the insertion subtask. The best iteration is marked in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

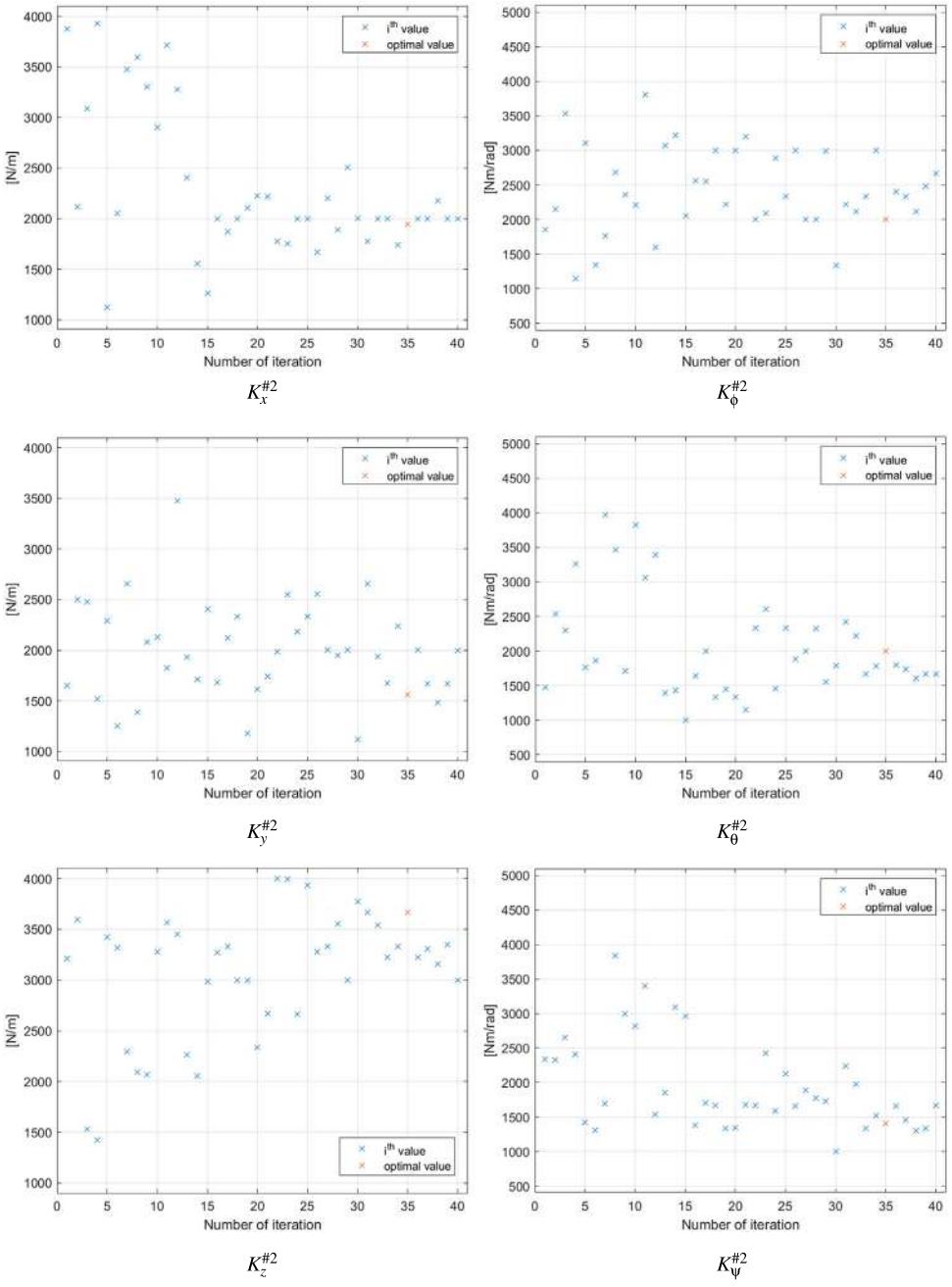
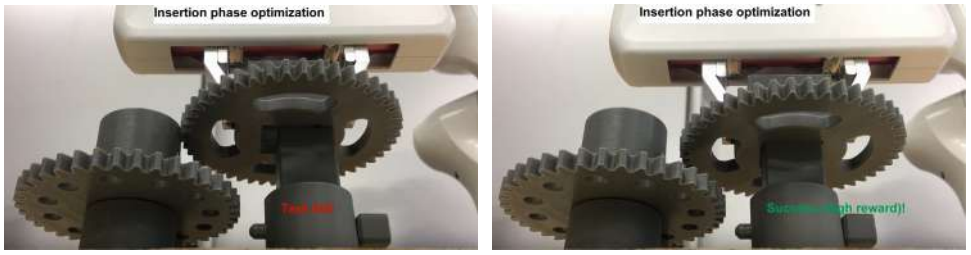


Fig. 13. $K_x^{\#2}$, $K_y^{\#2}$, $K_z^{\#2}$, $K_\phi^{\#2}$, $K_\theta^{\#2}$ and $K_\psi^{\#2}$ stiffness parameters optimization related to the second sub-optimization of the insertion subtask. The best iteration is marked in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Subtask #2 failure Subtask #2 success

Fig. 14. Insertion subtask #2 optimization: a subtask failure and a subtask successful execution are shown.

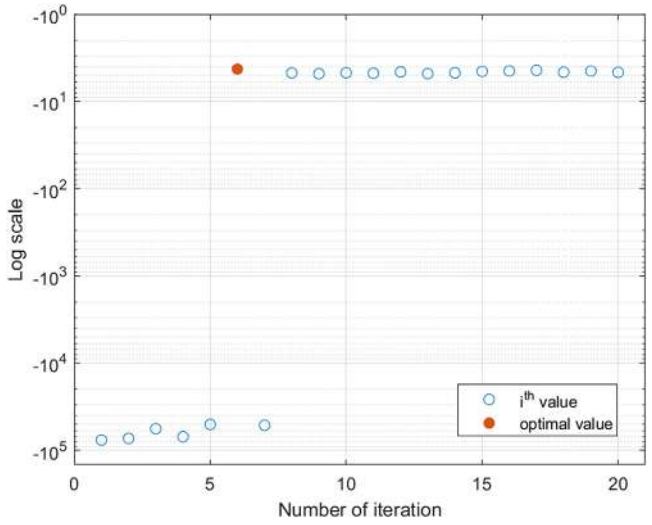


Fig. 15. Cost function $J_{BO}^{\#3}$ values over the experimental iterations for the optimization of the pushing subtask. The best iteration is marked by the red circle. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The cost function $J_{BO}^{\#2}$ observed values over the optimization iterations are shown in Fig. 9. The cost function has to be maximized. $J_{BO}^{\#2}$ values in the range $[-10^{-5}, -10^{-4}]$ are related to the penalty term in (15) (*i.e.*, safety emergency stop or subtask failure). The values of the $x_x^{d,\#2}$, $x_y^{d,\#2}$ and $x_\psi^{d,\#2}$ DoFs over the optimization iterations are shown in Fig. 10. It is possible to highlight the evolution of the optimization procedure, exploring in the initial iterations and optimizing while the optimization evolves, converging to the optimum set of optimization variables. The first sub-optimization returns the optimized parameters $x_x^{d,\#2,opt_1}$, $x_y^{d,\#2,opt_1}$ and $x_\psi^{d,\#2,opt_1}$. Such parameters will be used by the second sub-optimization in order to define the optimization variables ranges.

The second sub-optimization allows to optimize the $x_x^{d,\#2}$, $x_y^{d,\#2}$ and $x_z^{d,\#2}$ translational DoFs and the $x_\psi^{d,\#2}$ rotational DoF (rotation about the z axis). In addition, the stiffness parameters of $\mathbf{K}^{\#2}$ are also optimized. The damping parameters of $\mathbf{D}^{\#2}$ are then computed on the basis of the relation $\mathbf{D} = 2\boldsymbol{\epsilon}_D\sqrt{\mathbf{M}\mathbf{K}^{\#2}}$, where $\boldsymbol{\epsilon}_D$ is the diagonal damping ratio matrix, in which all the parameters have been imposed equal to 1 (*i.e.*, critical damping imposed to the controlled robot). The considered ranges of the optimization variables are: $x_x^{d,\#2,opt_1} \pm 1$ mm for the $x_x^{d,\#2}$ DoF, $x_y^{d,\#2,opt_1} \pm 1$ mm for the $x_y^{d,\#2}$ DoF, $[x_{f,z}^{d,\#2} - 15\text{ mm}, x_{f,z}^{d,\#2}]$ for the $x_z^{d,\#2}$ DoF, $x_\psi^{d,\#2,opt_1} \pm 1°$ for the $x_\psi^{d,\#2}$ DoF, [1000, 4000] N/m for the translational stiffness parameters, [500, 5000] Nm/rad for the rotational stiffness parameters. Considering the sensorless Cartesian impedance setpoint, the proposed ranges exploit the Cartesian pose $\mathbf{x}_f^{d,\#2}$ selected by the HMM in Section 6.2, and the previously optimized parameters $x_x^{d,\#2,opt_1}$, $x_y^{d,\#2,opt_1}$ and $x_\psi^{d,\#2,opt_1}$. In such a sub-optimization, it is possible to locally re-optimize $x_x^{d,\#2}$, $x_y^{d,\#2}$, and $x_\psi^{d,\#2}$ DoFs. In addition, the $x_z^{d,\#2}$ DoF is optimized to achieve good position-tracking performance. The optimization of the stiffness parameters $\mathbf{K}^{\#2}$ allows to implement the best compliant robot behavior to perform the target assembly (*i.e.*, being able to insert the part without applying excessive interaction forces).

The cost function $J_{BO}^{\#2}$ observed values over the optimization iterations are shown in Fig. 11. The cost function has to be maximized. In this second sub-optimization, no subtask failures are shown. In fact, the insertion location is identified in the previous sub-optimization. The second sub-optimization allows to optimize the control parameters, *i.e.*, maximizing the task performance. The values of the $x_x^{d,\#2}$, $x_y^{d,\#2}$, $x_z^{d,\#2}$ and $x_\psi^{d,\#2}$ DoFs over the optimization iterations are shown in Fig. 12. The values of the stiffness parameters $\mathbf{K}^{\#2}$ over the optimization iterations are shown in Fig. 13. It is possible to highlight the evolution of the optimization procedure, exploring in the initial iterations and optimizing while the optimization evolves, converging to the optimum set of optimization variables for both the DoFs optimization variables and for the stiffness optimization variables. The second sub-optimization returns the optimized parameters $x_x^{d,\#2,opt}$, $x_y^{d,\#2,opt}$, $x_z^{d,\#2,opt}$, $x_\psi^{d,\#2,opt}$, $K_x^{\#2,opt}$, $K_y^{\#2,opt}$, $K_z^{\#2,opt}$, $K_\phi^{\#2,opt}$, $K_\theta^{\#2,opt}$, and $K_\psi^{\#2,opt}$. Such parameters will be used by the pushing

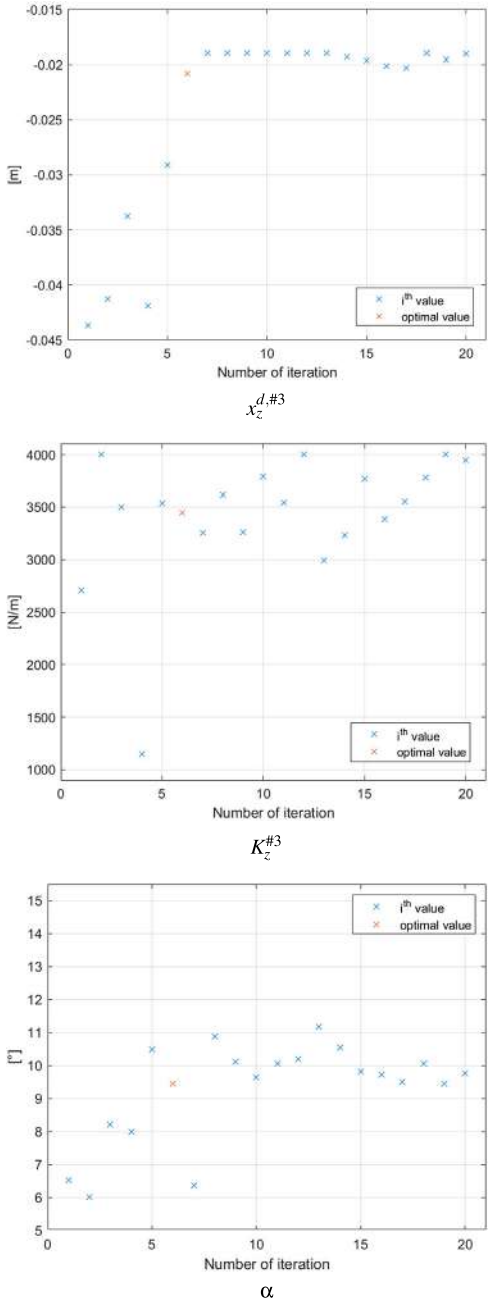
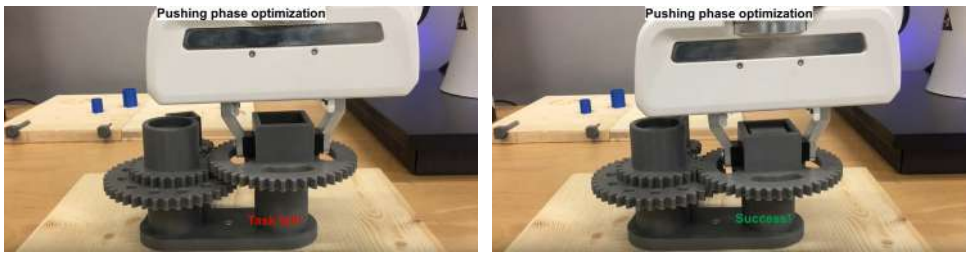
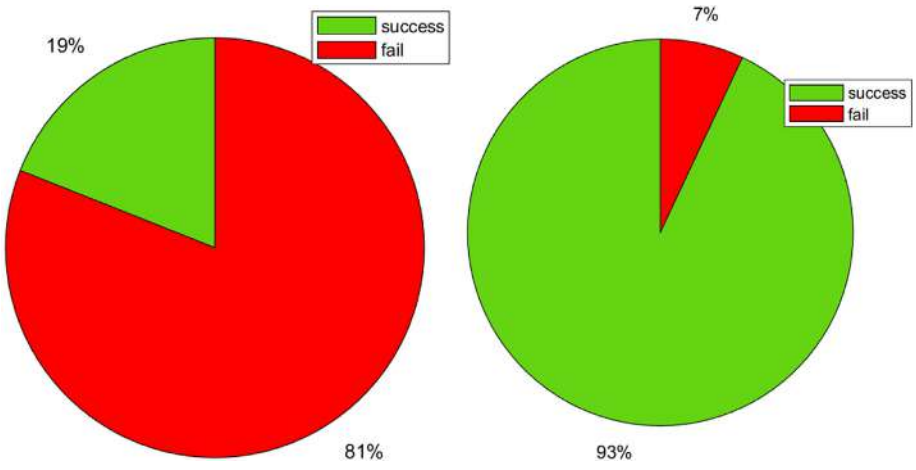


Fig. 16. $x_z^{d,\#3}$, $K_z^{\#3}$ and α values related to the optimization of the pushing subtask. The best iteration is marked in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



Subtask #2 failure Subtask #2 success

Fig. 17. Push subtask #3 optimization: a subtask failure and a subtask successful execution are shown.



HMM HMM+BO

Fig. 18. HMM assembly task success rate vs. HMM+BO task success rate.

subtask optimization in order to define the optimization variables ranges.

For the considered subtask #2, Fig. 14 shows an optimization iteration related to a subtask failure (the part is not inserted in its shaft) and an optimization iteration related to a successful subtask execution (the part is inserted in its shaft).

6.3.2. Pushing subtask optimization

Exploiting the cost function $J_{BO}^{\#3}$ proposed in (22), the pushing subtask can be optimized. In particular, the proposed cost function allows for the exploration and optimization of the required interaction force to be applied to finalize the assembly (*i.e.*, having the robot capable to find the reference surface to complete the assembly). The optimization of the pushing subtask is performed in one-shoot. In particular, the Cartesian DoF $x_z^{d,\#3}$ and the $K_z^{\#3}$ stiffness parameter (*i.e.*, related to the z DoF) are considered as optimization variables. In addition, a sinusoidal motion is superimposed to the DoF $x_\psi^{d,\#3}$ (*i.e.*, rotation about the z axis) in order to make the gears engagement effective:

$$\psi_{sin}(t) = \alpha + \sin(\omega_{sin} t),$$

where α is the sinusoidal motion amplitude and $\omega_{sin} = 1$ rad/s is the sinusoidal motion pulsation. α is included in the optimization variables. The parameters not involved in the pushing subtask optimization have been imposed to the values resulting from the output of the insertion subtask optimization. The considered ranges of the optimization variables are: $[x_z^{d,\#2,opt} - 20\text{ mm}, x_z^{d,\#2,opt}]$ mm for the $x_z^{d,\#3}$ DoF, [1000, 4000] N/m for $K_z^{\#3}$, $[0, 15]^\circ$ for α. Considering $x_z^{d,\#3}$, the proposed range exploit the Cartesian pose $\mathbf{x}_f^{d,\#3}$ selected by the HMM in Section 6.2.

The cost function $J_{BO}^{\#3}$ observed values over the optimization iterations are shown in Fig. 15. The cost function has to be maximized. $J_{BO}^{\#2}$ values in the range $[-10^{-5}, -10^{-4}]$ are related to the penalty term in (15) (*i.e.*, safety emergency stop or subtask failure). The values of the optimization variables $x_z^{d,\#3}$, $K_z^{\#3}$, and α over the optimization iterations are shown in Fig. 16. It is possible to highlight the evolution of the optimization procedure, exploring in the initial iterations and optimizing while the optimization evolves, converging to the optimum set of optimization variables. The pushing subtask optimization returns the optimized parameters $x_z^{d,\#3,opt}$, $K_z^{\#3,opt}$, and $\alpha^{\#3,opt}$.

For the considered subtask #3, Fig. 17 shows an optimization iteration related to a subtask failure (the engagement between the two gear is not performed) and an optimization iteration related to a successful subtask execution (the engagement between the two gear is performed).

Remark 16. It has to be underlined that it is possible to extend or reduce the range of the optimization variables on the basis of the target assembly task to be performed (such as, mounting tolerances). It is, in addition, possible to add/remove optimization variables.

Remark 17. Once the optimization of all the considered subtasks is concluded, the optimized values are stored and applied to the autonomous execution of the assembly task.

6.4. HMM+BO vs. HMM

The proposed HMM+BO procedure has been compared with the HMM procedure without optimization, in order to highlight the improved performance provided by the BO algorithm, capable to compensate for task uncertainties. In fact, in the second case, the task uncertainties cannot be compensated.

In order to evaluate the two methodologies, for each of the 15 subject-learned tasks (as described in Section 6.2), 10 repetitions of the assembly task have been performed for both the HMM+BO procedure (*i.e.*, using the optimized parameters) and for the HMM procedure (*i.e.*, using the learned non-optimized nominal trajectories). Therefore, 150 repetitions of the assembly task have been performed for both the HMM+BO procedure and for the HMM procedure. Fig. 18 shows the comparison between the two methodologies, highlighting the success rate of the assembly task. An assembly task is considered successful if all the three subtasks are correctly performed. If a subtasks is not correctly performed (*e.g.*, the insertion subtask fails due to the impossibility to insert the gear into its shaft), the assembly task is ended and the failure is reported. From Fig. 18 it can be noted that the proposed HMM+BO procedure allows to compensate for the task uncertainties, resulting in a successful execution rate of 93%, while the HMM algorithm alone is not capable to take such uncertainties into account, resulting in a successful execution rate of only 19%.

7. Conclusions

The presented paper proposed machine learning techniques to make a sensorless robot (*i.e.*, no force/torque sensor is used) able to learn and optimize an industrial assembly task. Exploiting sensorless Cartesian impedance control (and being extendible to position control-based impedance control for the robots not equipped with torque control), (i) a task-trajectory learning algorithm (implying a few human's demonstrations) based on a Hidden Markov Model approach and (ii) an autonomous task optimization procedure based on a Bayesian Optimization algorithm have been derived. In such a way, human's knowledge related to the task is transferred to the robot. In addition, sensorless Cartesian impedance control parameters are optimized in order to maximize the task performance. To validate the proposed methodology, an assembly task of a gear into its square shaft has been selected as a reference application. A Franka EMIKA Panda manipulator has been used as a test platform, implementing the proposed methodology. Validation experiments show the capabilities of the proposed approach, making the robot learning the task execution while compensating for the task uncertainties. The HMM+BO methodology and the HMM algorithm without optimization have been compared. Such comparison shows the capabilities of the optimization stage to compensate for task uncertainties. In particular, the HMM+BO methodology shows an assembly task success rate of 93%, while the HMM algorithm shows a success rate of only 19%.

Current and future work is investigating inverse reinforcement learning approaches and model-based reinforcement learning approaches for task learning. In particular, continuous task learning is under investigation, in order to avoid the task segmentation. In addition, collaborative controllers to relieve the human from the manipulation of heavy loads are under development.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The work has been developed within the project ASSASSINN, funded from H2020 CleanSky 2, Switzerland under grant agreement n. 886977.

References

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, M. Hoffmann, Industry 4.0, Bus. Inf. Syst. Eng. 6 (4) (2014) 239–242.
[2] R. Goel, P. Gupta, Robotics and industry 4.0, in: A Roadmap to Industry 4.0: Smart Production, Sharp Business and Sustainable Development, Springer, 2020, pp. 157–169.
[3] ISO 10218-1:2011: Robots and Robotic Devices–Safety Requirements for Industrial Robots–Part 1: Robots, Standard, International Organization for Standardization, Geneva, CH, 2011.
[4] ISO TS 15066-Robots and Robotic Devices-Collaborative Robots, Standard, International Organization for Standardization, Geneva, CH, 2016.
[5] L. Roveda, Adaptive interaction controller for compliant robot base applications, IEEE Access 7 (2018) 6553–6561.
[6] F. Vicentini, N. Pedrocchi, M. Beschi, M. Giussani, N. Iannacci, P. Magnoni, S. Pellegrinelli, L. Roveda, E. Villagrossi, M. Askarpour, I. Maurtua, A. Tellaeche, F. Becchi, G. Stellin, G. Fogliazza, PIROS: Cooperative, safe and reconfigurable robotic companion for CNC pallets load/unload stations, in: Bringing Innovative Robotic Technologies from Research Labs to Industrial End-Users: The Experience of the European Robotics Challenges, Springer International Publishing, Cham, 2020, pp. 57–96, http://dx.doi.org/10.1007/978-3-030-34507-5_4.
[7] R. French, M. Benakis, H. Marin-Reyes, Intelligent sensing for robotic re-manufacturing in aerospace—An industry 4.0 design based prototype, in: 2017 IEEE International Symposium on Robotics and Intelligent Sensors, IRIS, IEEE, 2017, pp. 272–277.
[8] L. Roveda, N. Pedrocchi, L.M. Tosatti, Exploiting impedance shaping approaches to overcome force overshoots in delicate interaction tasks, Int. J. Adv. Robot. Syst. 13 (5) (2016) 1729881416662771.
[9] S. Bragança, E. Costa, I. Castellucci, P.M. Arezes, A brief overview of the use of collaborative robots in industry 4.0: Human role and safety, in: Occupational and Environmental Safety and Health, Springer, 2019, pp. 641–650.
[10] A.R. Mahmood, D. Korenkevych, B.J. Komer, J. Bergstra, Setting up a reinforcement learning task with a real-world robot, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2018, pp. 4635–4640.
[11] G. Canal, G. Alenyà, C. Torras, Adapting robot task planning to user preferences: An assistive shoe dressing example, Auton. Robots 43 (6) (2019) 1343–1356.
[12] P. Tsarouchi, A.-S. Matthaiakis, S. Makris, G. Chryssolouris, On a human-robot collaboration in an assembly cell, Int. J. Comput. Integr. Manuf. 30 (6) (2017) 580–589.
[13] Y. Ma, K. Du, D. Zhou, J. Zhang, X. Liu, D. Xu, Automatic precision robot assembly system with microscopic vision and force sensor, Int. J. Adv. Robot. Syst. 16 (3) (2019) 1729881419851619.
[14] T. Fitzgerald, A.K. Goel, A. Thomaz, Human-Robot co-creativity: Task transfer on a spectrum of similarity, in: ICCC, 2017, pp. 104–111.
[15] N. Koenig, M.J. Matarić, Robot life-long task learning from human demonstrations: A Bayesian approach, Auton. Robots 41 (5) (2017) 1173–1188.
[16] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, S. Levine, Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 3758–3765.
[17] R. Caccavale, M. Saveriano, A. Finzi, D. Lee, Kinesthetic teaching and attentional supervision of structured tasks in human–robot interaction, Auton. Robots 43 (6) (2019) 1291–1307.
[18] S. Pellegrinelli, H. Admoni, S. Javdani, S. Srinivasa, Human-robot shared workspace collaboration via hindsight optimization, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2016, pp. 831–838.
[19] M.P. Polverini, R. Rossi, G. Morandi, L. Bascetta, A.M. Zanchettin, P. Rocco, Performance improvement of implicit integral robot force control through constraint-based optimization, in: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, IEEE, 2016, pp. 3368–3373.
[20] A.S. Polydoros, L. Nalpantidis, Survey of model-based reinforcement learning: Applications on robotics, J. Intell. Robot. Syst. 86 (2) (2017) 153–173.
[21] A. Mosavi, A. Varkonyi, Learning in robotics, Int. J. Comput. Appl. 157 (1) (2017) 8–11.
[22] J. Peters, Machine learning for motor skills in robotics, KI-Künstliche Intell. 2008 (4) (2008) 41–43.
[23] D. Nguyen-Tuong, J. Peters, Model learning for robot control: A survey, Cogn. Process. 12 (4) (2011) 319–340.
[24] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J.A. Ojea, E. Solowjow, S. Levine, Residual reinforcement learning for robot control, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 6023–6029.
[25] P. Van Molle, T. Verbelen, E. De Coninck, C. De Boom, P. Simoens, B. Dhoedt, Learning to grasp from a single demonstration, 2018, arXiv preprint arXiv:1806.03486.

[26] J.E. Laird, K. Gluck, J. Anderson, K.D. Forbus, O.C. Jenkins, C. Lebiere, D. Salvucci, M. Scheutz, A. Thomaz, G. Trafton, et al., Interactive task learning, IEEE Intell. Syst. 32 (4) (2017) 6–21.
[27] L. Roveda, G. Pallucca, N. Pedrocchi, F. Braghin, L.M. Tosatti, Iterative learning procedure with reinforcement for high-accuracy force tracking in robotized tasks, IEEE Trans. Ind. Inf. 14 (4) (2017) 1753–1763.
[28] S. Levine, N. Wagener, P. Abbeel, Learning contact-rich manipulation skills with guided policy search, CoRR abs/1501.05611 (2015).
[29] G. Thomas, M. Chien, A. Tamar, J.A. Ojea, P. Abbeel, Learning robotic assembly from CAD, CoRR abs/1803.07635 (2018).
[30] R.A. Knepper, T. Layton, J.W. Romanishin, D.L. Rus, IkeaBot: An autonomous multi-robot coordinated furniture assembly system, in: Proceedings of the 2013 IEEE International Conference on Robotics and Automation, 2013.
[31] J. Peters, S. Schaal, Reinforcement learning of motor skills with policy gradients, Neural Netw. (2008).
[32] S. Gu, E. Holly, T. Lillicrap, S. Levine, Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, CoRR (2016).
[33] A. Vakanski, I. Mantegh, A. Irish, F. Janabi-Sharifi, Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping, IEEE Trans. Syst. Man Cybern. B 42 (4) (2012) 1039–1052.
[34] C.G. Atkeson, S. Schaal, Robot learning from demonstration, in: ICML, 1997.
[35] A. Billard, D. Grollman, Robot learning by demonstration, Scholarpedia 8 (12) (2013).
[36] S. Calinon, F. Guenter, A. Billard, On learning, representing and generalizing a task in a humanoid robot, IEEE Trans. Syst. Man Cybern. 37 (2007) 286–298.
[37] J. Chen, A. Zelinski, Programming by demonstration: Coping with suboptimal teaching actions, Int. J. Robot. Res. (2003).
[38] K. Kronander, A. Billard, Learning compliant manipulation through kinesthetic and tactile human-robot interaction, IEEE Trans. Haptics 7 (3) (2014) 367–380.
[39] J. Xu, Z. Hou, Z. Liu, H. Qiao, Compare contact model-based control and contact model-free learning: A survey of robotic peg-in-hole assembly strategies, 2019, arXiv preprint arXiv:1904.05240.
[40] G. Thomas, M. Chien, A. Tamar, J.A. Ojea, P. Abbeel, Learning robotic assembly from CAD, in: 2018 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2018, pp. 1–9.
[41] A. Abbas, F.D. Maire, F. Dayoub, S. Shirazi, Combining Learning from Demonstration and Search Algorithm for Dynamic Goal-Directed Assembly Task Planning, 2018.
[42] Z. Hou, M. Philipp, K. Zhang, Y. Guan, K. Chen, J. Xu, The learning-based optimization algorithm for robotic dual peg-in-hole assembly, Assem. Autom. (2018).
[43] Y. Fan, J. Luo, M. Tomizuka, Teach industrial robots peg-hole-insertion by human demonstration, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 811–817.
[44] M.A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, J. Bohg, Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks, in: 2019 International Conference on Robotics and Automation, ICRA, IEEE, 2019, pp. 8943–8950.
[45] M.P. Polverini, A.M. Zanchettin, S. Castello, P. Rocco, Sensorless and constraint based peg-in-hole task execution with a dual-arm robot, in: 2016 IEEE International Conference on Robotics and Automation, ICRA, IEEE, 2016, pp. 415–420.
[46] L. Roveda, D. Piga, Robust state dependent riccati equation variable impedance control for robotic force-tracking tasks, Int. J. Intell. Robot. Appl. (2020) 1–13.
[47] K. Kaipa, C. Morato, B. Zhao, S.K. Gupta, Instruction generation for assembly operations performed by humans, in: ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, American Society of Mechanical Engineers Digital Collection, 2012, pp. 1121–1130.
[48] P. Pastor, M. Kalakrishnan, F. Meier, F. Stulp, J. Buchli, E. Theodorou, S. Schaal, From dynamic movement primitives to associative skill memories, Robot. Auton. Syst. 61 (4) (2013) 351–361.
[49] M.A. Lee, Y. Zhu, P. Zachares, M. Tan, K. Srinivasan, S. Savarese, L. Fei-Fei, A. Garg, J. Bohg, Making sense of vision and touch: Learning multimodal representations for contact-rich tasks, IEEE Trans. Robot. (2020).
[50] H. Park, J. Park, D.-H. Lee, J.-H. Park, M.-H. Baeg, J.-H. Bae, Compliance-based robotic peg-in-hole assembly strategy without force feedback, IEEE Trans. Ind. Electron. 64 (8) (2017) 6299–6309.
[51] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.
[52] L. Roveda, M. Forgione, D. Piga, Robot control parameters auto-tuning in trajectory tracking applications, Control Eng. Pract. 101 (2020) 104488.
[53] B. Siciliano, L. Villani, Robot Force Control, first ed., Kluwer Academic Publishers, Norwell, MA, USA, 2000.
[54] P.R. Chang, C.G. Lee, Residue arithmetic VLSI array architecture for manipulator pseudo-inverse jacobian computation, in: Proceedings. 1988 IEEE International Conference on Robotics and Automation, IEEE, 1988, pp. 297–302.
[55] N. Pedrocchi, E. Villagrossi, F. Vicentini, L. Molinari Tosatti, On robot dynamic model identification through sub-workspace evolved trajectories for optimal torque estimation, in: Intelligent Robots and Systems, IROS, 2013 IEEE/RSJ International Conference on, IEEE, 2013, pp. 2370–2376.
[56] A. Vakanski, I. Mantegh, A. Irish, F. Janabi-Sharifi, Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping, IEEE Trans. Syst. Man Cybern. B 42 (4) (2012) 1039–1052.
[57] L. Durak, O. Arikan, Short-time Fourier transform: Two fundamental properties and an optimal implementation, IEEE Trans. Signal Process. 51 (5) (2003) 1231–1242, http://dx.doi.org/10.1109/TSP.2003.810293.
[58] Y. Linde, A. Buzo, R. Gray, An algorithm for vector quantizer design, IEEE Trans. Commun. 28 (1) (1980) 84–95.
[59] A.R. De Pierro, A modified expectation maximization algorithm for penalized likelihood estimation in emission tomography, IEEE Trans. Med. Imaging 14 (1) (1995) 132–137, http://dx.doi.org/10.1109/42.370409.
[60] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.
[61] L. Biagiotti, C. Melchiorri, Trajectory Planning for Automatic Machines and Robots, first ed., Springer Publishing Company, Incorporated, 2008.
[62] R. Zhao, Trajectory Planning and Control for Robot Manipulators (Ph.D. thesis), 2015.
[63] E. Brochu, V.M. Cora, N. De Freitas, A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning, 2010, arXiv preprint:1012.2599.
[64] L. Roveda, D. Piga, Interaction force computation exploiting environment stiffness estimation for sensorless robot applications, in: Metrology for Industry 4.0 and IoT, 2020.
[65] L. Roveda, A. Bussolan, F. Braghin, D. Piga, 6D virtual sensor for wrench estimation in robotized interaction tasks exploiting extended Kalman filter, Machines 8 (4) (2020) 67.
[66] D.R. Jones, M. Schonlau, W.J. Welch, Efficient global optimization of expensive black-box functions, J. Global Optim. 13 (4) (1998) 455–492.
[67] C. Gaz, M. Cognetti, A. Oliva, P.R. Giordano, A. De Luca, Dynamic identification of the franka emika panda robot with retrieval of feasible parameters using penalty-based optimization, IEEE Robot. Autom. Lett. 4 (4) (2019) 4147–4154.
[68] A. Cully, K. Chatzilygeroudis, F. Allocati, J.-B. Mouret, Limbo: A fast and flexible library for Bayesian optimization, 2016, arXiv preprint arXiv:1611.07343.
[69] S.G. Johnson, The NLopt nonlinear-optimization package, 2020, http://github.com/stevengj/nlopt.

Ph.D. Loris Roveda received the M.Sc. (2011) and the Ph.D. (2015) in Mechanical Engineering at Politecnico di Milano. Currently, he is a senior researcher at SUPSI-IDSIA, working on AI and ML techniques applied to industrial robotics (such as robot control, human–robot collaboration, dynamics identification). He has been involved in many national and European projects, and he is now coordinating the EUROBENCH STEPbySTEP project and he is the P.I. of the H2020 CS2 ASSASSINN project.

Eng. Mauro Magni: Mauro Magni received the M.Sc. (2019) in Mechanical Engineering at Politecnico di Milano. Currently, he is a junior application engineering at CEA (Rancate, Switzerland), a brand of Vignal Lighting Group (Corbas, France), as "methods and maintenance" tasks. The mansion involves directly in the assembly line and it relates to industrial robots, PLC programming, design and manufacturing of the process control.

Eng. Martina Cantoni: Martina Cantoni received the M.Sc. (2019) in Mechanical Engineering at Politecnico di Milano. Currently, she is a project designer at Serinex (Italy), a company active in the field of machine tools, working on the design and development of new lines of products.

Ph.D. Dario Piga: Dario Piga received his Ph.D. in Systems Engineering from the Politecnico di Torino (Italy) in 2012. He was a Postdoctoral Researcher at the Delft University of Technology (The Netherlands) in 2012 and at the Eindhoven University of Technology (The Netherlands) in 2013. From 2014 to early 2017 he was Assistant Professor at the IMT School for Advanced Studies Lucca (Italy) and since March 2017 he has been Senior Researcher at the IDSIA Dalle Molle Institute for Artificial Intelligence in Lugano (Switzerland) and Lecturer at the SUPSI University of Applied Sciences and Arts of Southern Switzerland. His main research interests include system identification, robust control, Bayesian filtering and non-convex optimization, with applications to process control and smart manufacturing.

Prof. Giuseppe Bucca: Giuseppe Bucca is Associate Professor of Applied Mechanics at Politecnico di Milano. He graduated in Mechanical Engineering in 2001. In 2005 he received the Ph.D. in Mechanical Systems Engineering at Politecnico di Milano and in 2007 he joined the academic permanent staff in the same university. His research activity and teaching activity are focused on the dynamics and control of mechanical systems. He participated and participates to several research project funded by EU and he was and is responsible of research and technical projects funded by private companies.