Assembly Task Learning and Optimization through Human's Demonstration and Machine Learning

1st Loris Roveda
Istituto Dalle Molle di studi sull'Intelligenza Artificiale (IDSIA)
Scuola Universitaria Professionale della Svizzera Italiana (SUPSI)
Università della Svizzera Italiana (USI) IDSIA-SUPSI Lugano, Switzerland
email address: loris.roveda@idsia.ch. ORCID: 0000-0002-4427-536X

2nd Mauro Magni
Department of Mechanical Engineering
Politecnico di Milano
Lecco, Italy
email address: mauro2.magni@mail.polimi.it

3rd Martina Cantoni
Department of Mechanical Engineering
Politecnico di Milano
Lecco, Italy
email address: martina.cantoni@mail.polimi.it

4th Dario Piga
Istituto Dalle Molle di studi sull'Intelligenza Artificiale (IDSIA)
Scuola Universitaria Professionale della Svizzera Italiana (SUPSI)
Università della Svizzera Italiana (USI) IDSIA-SUPSI Lugano, Switzerland
email address: dario.piga@supsi.ch

5th Giuseppe Bucca
Department of Mechanical Engineering
Politecnico di Milano
Lecco, Italy
email address: giuseppe.bucca@polimi.it

***Abstract*—Robots are increasingly exploited in production plants, with the need to learn and to adapt themselves to new tasks. This 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, a task-trajectory learning algorithm exploiting a limited number of human's demonstrations (based on Hidden Markov Model), and an autonomous optimization procedure (based on Bayesian Optimization) are proposed to learn and optimize the assembly task. To validate the proposed methodology, an assembly task of a gear into its square-section shaft has been considered. A Franka EMIKA Panda manipulator has been used as a test platform. The experiments show the effectiveness of the proposed strategy, making the robot able to learn and optimize its behaviour to accomplish the assembly task, even in the presence of uncertainties.**

***Index Terms*—Industrial Robots, Task Learning and Optimization, Sensorless Impedance Control, Hidden Markov Model, Bayesian Optimization.**

I. Introduction

Industry 4.0 paradigm [1] is proposing an enhanced production environment in which robots are increasingly required to be intelligent systems. Assembly tasks are one of the most required industrial applications to be executed by a manipulator, autonomously or in collaboration with the human operator. 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. [2].

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

Machine learning offers to robotics a framework and a set of tools for the design of sophisticated and hard-to-engineer behaviours and applications; conversely, the challenges of robotic problems provide both inspiration, impact, and validation for developments in robot learning [3]–[6]. Machine learning techniques have been widely applied to robotics in different fields, such as motor skill learning for human-robot collaboration purposes [7], robot model learning [8], control design and tuning [9], grasping capabilities [10], etc.

Machine learning has found a huge application in robotized task learning [11]. 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 [12] the iterative learning and reinforcement learning procedures are applied to an automotive industrial assembly task, to automatize the compliance controllers parameters tuning. In [13] a policy search method is used to learn a range of dynamics manipulation behaviors without using known models or example demonstrations. [14] proposes a method that combines motion planning with reinforcement learning policy search for efficient learning of assembly tasks. 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. [15] developed a method for learning and

reproduction of complex trajectories for robot programming by demonstration applied to a painting process. In [16] 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 [17] 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 [18], where a robot learns how to make a chess move by generalizing across different demonstrations.

Assembly tasks are one of the most investigated applications considering task learning methodologies [19]. In fact, it is not trivial to define *a priori* the sequence of all the subtasks, the control parameters, etc. In [20] 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 [21] a strategy to guide the user during the assembly task phases is proposed, on the basis of the assembly CAD data. In [22] the learning of the required assembly forces is performed on the basis of human's demonstrations. In [23] an optimization algorithm is presented, in combination with impedance control, to optimize the robotic dual peg-in hole assembly task, reducing the assembly time while smoothing the contact forces during the assembly.

Once the reference task has been learned, its (autonomous) optimization 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). 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 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) industrial robot is even more difficult [24]. In fact, without the measurements of the interaction forces the assembly procedure has to rely only on the robot Cartesian pose information.

A. Paper contribution

This paper proposes a methodology to make a sensorless robot (*i.e.*, no force/torque sensor is used to close the control loop) 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, and (2) an optimization procedure of the task execution. Considering (1), a limited number of human's demonstrations of the assembly task execution are performed, recording the robot Cartesian end-effector pose. The recorded data are then processed by a Hidden Markov Model (HMM) to select the nominal reference assembly task trajectory. Considering (2), a Bayesian Optimization (BO)-based algorithm has been designed to autonomously optimize the impedance control parameters. The objective of the BO algorithm is to maximize the task performance, making the robot able to compensate for task uncertainties. 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, which can be derived [25] or identified from the part assembly-datasheet. To validate the proposed methodology, an assembly task of a gear into its square-section shaft has been considered. A Franka EMIKA Panda manipulator has been used as a test platform (Figure 1). Experimental results show the effectiveness of the proposed strategy, making the robot able to learn and optimize its behaviour to accomplish the assembly task, even in the presence of task uncertainties.

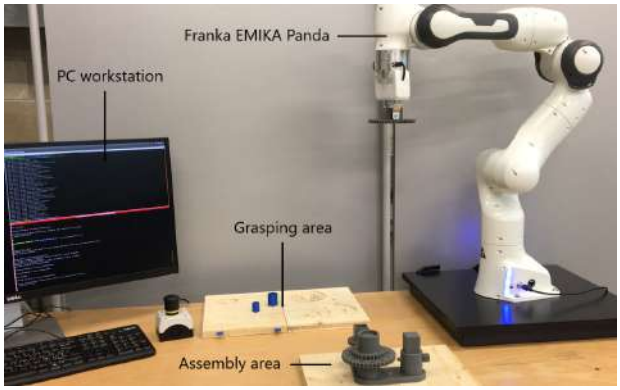


Fig. 1: Reference assembly task of a gear component performed by a Franka EMIKA panda robot.

II. Sensorless Cartesian impedance control

Sensorless Cartesian impedance control allows to implement a compliant behavior for the target robot, avoiding the use of force/torque sensors. The resulting controlled robot dynamics results in [26]:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{D}\dot{\mathbf{x}} + \mathbf{K}\Delta\mathbf{x} = -\overline{\mathbf{H}}(\mathbf{q})\mathbf{h}_{ext}, \tag{1}$$

where $\mathbf{M}$ is the impedance control mass matrix, $\mathbf{D}$ is the impedance control damping matrix, and $\mathbf{K}$ is the impedance control stiffness matrix (all matrices including translational and rotational components). $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}^d$, where $\mathbf{x}^d$ is the impedance setpoint and $\mathbf{x}$ is the current robot Cartesian pose. $\mathbf{h}_{ext}$ is the robot external force/torque vector. $\overline{\mathbf{H}}(\mathbf{q}) = \mathbf{M}\mathbf{J}(\mathbf{q})\mathbf{B}(\mathbf{q})^{-1}\mathbf{J}(\mathbf{q})^T$ (where $\mathbf{B}(\mathbf{q})$ is the robot inertia matrix and $\mathbf{J}(\mathbf{q})$ is the robot Jacobian matrix) couples the Cartesian degrees of freedom (DoFs). Such a coupling has to be compensated while executing the task.

III. Nominal reference task trajectory learning

Exploiting a limited number of human's demonstrations of the target task, the proposed HMM algorithm allows to select the most consistent demonstrated task trajectory to be used by the robot for the autonomous task execution. In particular, the HMM algorithm is applied to each Cartesian DoF separately, allowing to reduce teaching uncertainties. 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.

A. Hidden Markov Model-based methodology

Following the proposed methodology detailed in Figure 2, the following steps are implemented for the selection of the nominal reference task trajectory (*i.e.*, the impedance setpoint $\mathbf{x}^d(t)$) through the HMM algorithm:

- human's demonstrations data recording [27] (*Observed trajectories* block in Figure 2): the human guides the robot (in manual guidance control mode) to perform the task. The robot end-effector Cartesian pose $\mathbf{x}(t)$ is recorded for each time-instant t. The recording time is fixed, according to the task execution time. Each trajectory is then divided in 32 sub-intervals with equal duration;
- Short Time Fourier transformation [28] of the recorded data (*Short Time Fourier Transformation* block in Figure 2): each end-effector Cartesian pose $\mathbf{x}$ shifts into the frequency domain for each sub-interval;
- Linde–Buzo–Gray (LBG) clustering [29] (*LBG clustering* block in Figure 2): it is used to partition a set of input data into a number of clusters (states of the HMM), generating a codebook, exploiting the frequency domain data and the created sub-intervals;
- HMM algorithm [30] (*HMM initialization and training* block in Figure 2): the HMM approach is applied to select the most consistent task 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). The trajectory selection is performed on the basis of the likelihood index [31]: the observation giving the maximum likelihood index is considered to be the most consistent one, and it will be selected as the nominal reference trajectory;
- smoothing the selected trajectory (*Trajectory smoothing* block in Figure 2): the selected trajectory is post-processed by applying a *cubic smoothing spline* [32]. The aim of this post-processing is to limit the selected trajectory accelerations (*i.e.*, avoiding motion discontinuities) while preserving relevant trajectory information.

IV. Task optimization

The proposed BO algorithm [33] allows to maximize the task performance (*i.e.*, avoid task failures, limit interaction forces, guarantee the execution time) and to compensate for task uncertainties (*e.g.*, parts positioning, teaching uncertainties - *i.e.*, residual noise in the nominal reference trajectory, - coupled sensorless Cartesian impedance control dynamics). Such algorithm optimizes both the nominal learned reference trajectory (*i.e.*, the impedance setpoint $\mathbf{x}^{d,\#k}$) and the impedance control parameters (*i.e.*, the stiffness and damping parameters $\mathbf{K}^{\#k}$ and $\mathbf{D}^{\#k}$).

V. Experimental results

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.

A. Reference assembly task description

To validate the proposed methodology, an assembly of a gear into its square-section shaft has been considered (Figure 1). The task is considered successful if the gear is inserted in its shaft, making it engaging with the already installed gear.

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. 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;
- *pushing subtask* (subtask #3): the final pushing is performed to complete the assembly.

The robotic platform involved in the experiments is a Franka EMIKA panda manipulator. Exploiting its torque control mode (control frequency 1000 Hz), the sensorless Cartesian impedance controller described in Section II has been implemented.

B. Nominal reference task trajectory learning validation description and results

In order to evaluate the proposed nominal reference task trajectory learning methodology (described in Section III), 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 5 times in order to record the robot end-effector Cartesian pose trajectory $\mathbf{x}^{\#k}(t)$ (with a sampling frequency of 1000 Hz). The subject manually guided the robot (controlled in manual guidance mode) to perform each subtask. After the recording is performed, each subtask has been processed by the proposed methodology in order to select the most consistent subtask trajectory. As shown in Figure 3, the proposed HMM algorithm is capable to select the most consistent trajectory for each Cartesian DoF independently. In particular, the selected trajectory is the one which shows the maximum likelihood index computed by the HMM methodology, and it is highlighted in the plots.

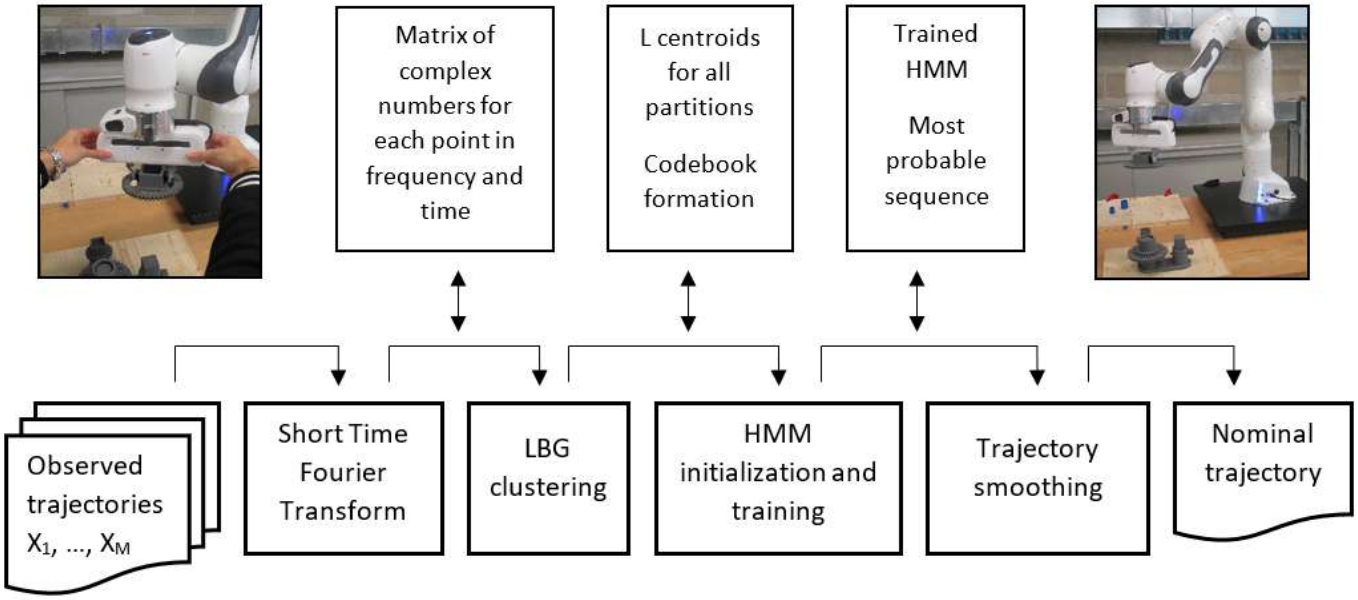


Fig. 2: Reference task trajectory learning methodology: on the basis of a limited number of human's demonstrations of the target task, the most consistent demonstrated task trajectory is selected for autonomous execution and optimization.

Considering the approaching subtask, the selected nominal reference trajectory $\mathbf{x}^{d,\#1}(t)$ is executed to position 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. 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.

Remark 1. 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.

C. Cost function definition

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 at pre-positioning the manipulated part in the correspondence of the assembly location.

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:

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

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

$$e_{xy}^{\#2} = max\left(\sqrt{\left(x_x^{d,\#2}(t) - x_x(t)\right)^2 + \left(x_y^{d,\#2}(t) - x_y(t)\right)^2}\right). \quad (3)$$

$x_x^{d,\#2}(t)$ 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 teaching uncertainties related to the approaching subtask.

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

$$e_z^{\#2} = max\left(\mid x_z^{d,\#2}(t) - x_z(t) \mid\right). \quad (4)$$

$x_z^{d,\#2}(t)$ 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 (2) is computed as the maximum value of the rotational error along the z axis:

$$e_\psi^{\#2} = max\left(\mid x_\psi^{d,\#2}(t) - x_\psi(t) \mid\right). \quad (5)$$

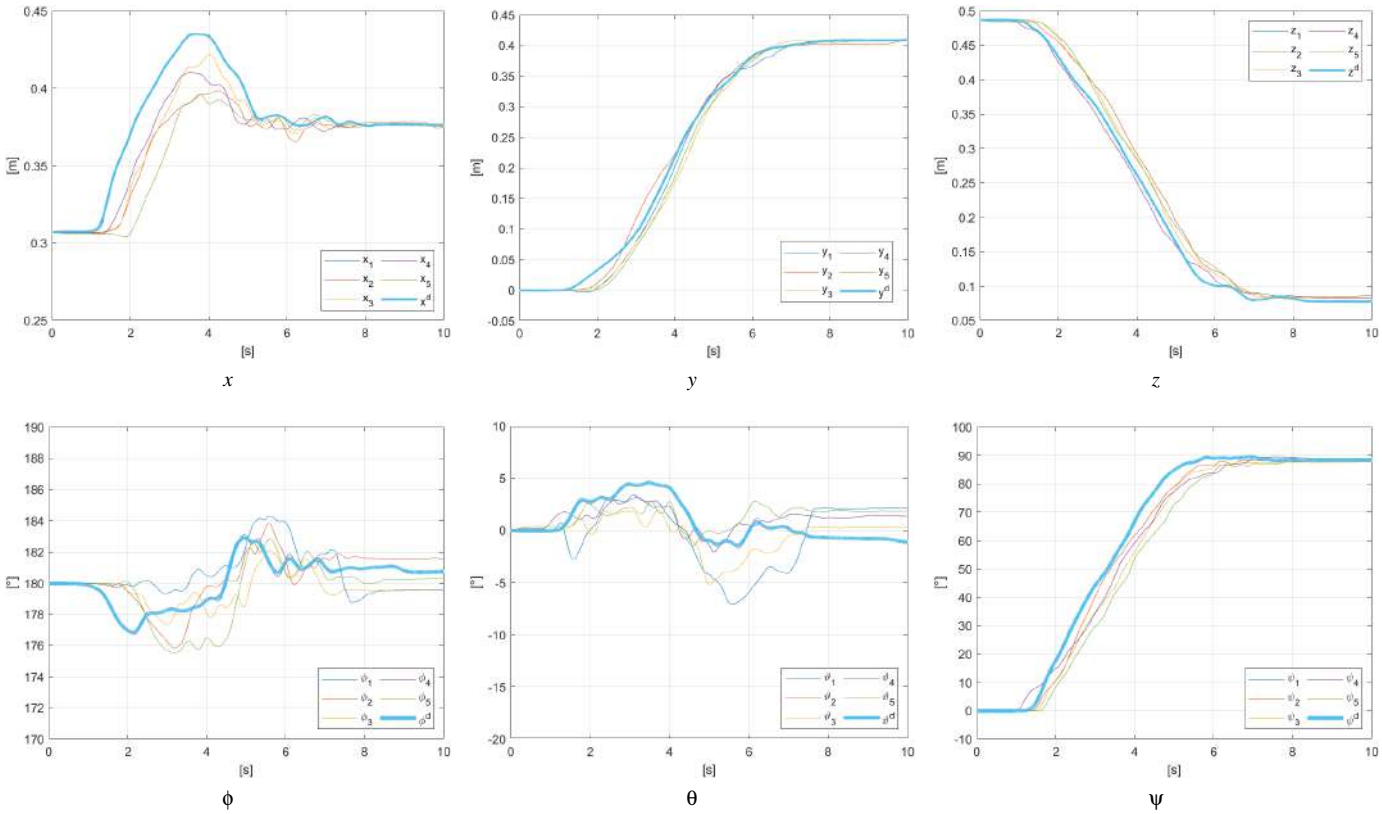


Fig. 3: The 5 subjects's demonstrations are shown for the Cartesian translational DoFs x, y, z and for the Cartesian rotational DoFs ϕ, θ, ψ. The computed nominal reference trajectory learned exploiting the proposed HMM methodology is highlighted for all the DoFs.

$x_{\psi}^{d,\#2}(t)$ 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 (2) is computed as the maximum value of the estimated interaction force:

$$e_f^{\#2} = max\left(\sqrt{f_x(t)^2 + f_y(t)^2 + f_z(t)^2}\right). \tag{6}$$

The i^{th} estimated interaction force is computed as:

$$f_i(t) = K_i^{\#2}\left(x_i^{d,\#2}(t) - x_i(t)\right). \tag{7}$$

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 (2) 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:

$$L^{\#2} = e^{-t/T^{\#2}} \text{ if } e_z^{\#2} > \bar{e}_z^{\#2}, \tag{8}$$

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.

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:

$$J_{BO}^{\#3} = -\left(G_z^{\#3} e_z^{\#3} + G_{\psi}^{\#3} e_{\psi}^{\#3} + G_f^{\#3} e_f^{\#3} + G_L^{\#3} L^{\#3}\right). \tag{9}$$

The term $e_z^{\#3}$ in (9) is computed similarly to $e_z^{\#2}$ in (4). Such term allows to avoid any stacking behavior during the subtask execution.

The term $e_{\psi}^{\#3}$ in (9) is computed similarly to $e_{\psi}^{\#2}$ in (5). Such term allows to compensate for misalignments related

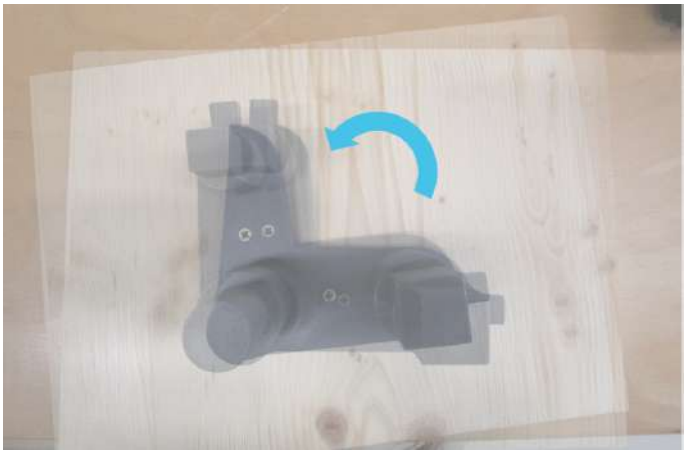


Fig. 4: Task assembly location uncertainties for the evaluation of the BO algorithm capabilities.

to parts/assembly location orientation uncertainties and/or for parts engagement.

The term $e_f^{\#3}$ in (9) is computed as the maximum value of the force error along the vertical direction:

$$e_f^{\#3} = max\left(\mid f_z^d - f_z(t)^2 \mid\right). \tag{10}$$

where the estimated interaction force $f_z(t)$ is computed as in (7) 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 (9) is computed similarly to $L^{\#2}$ in (8), 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.

D. Task optimization validation description and results

After the task trajectories are learned for all the subjects, an optimization has been performed independently for each subject-teached task, modifying the assembly location pose differently for each optimization execution (Figure 4). The following range of variation has been considered: ± 3 cm for translations x and y, and $\pm 3^\circ$ for rotation ψ. These uncertainties are, in fact, higher than what can be expected from real industrial assembly tasks. 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, exploiting its specific cost function.

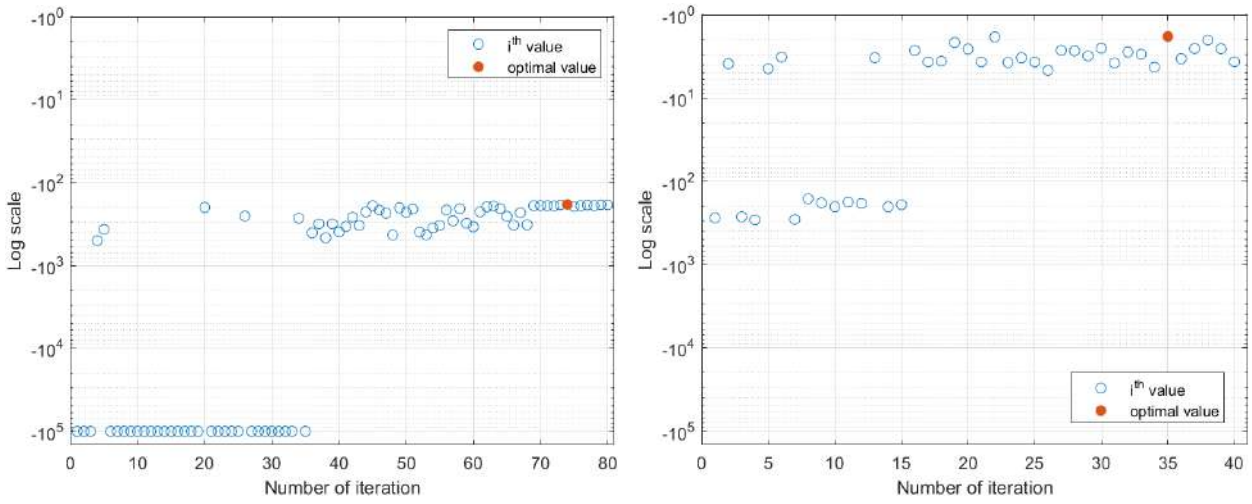
Remark 2. Results related to one optimization procedure are shown. Results related to the other 14 optimizations are similar.

1) Insertion subtask optimization: Exploiting the cost function $J_{BO}^{\#2}$ proposed in (2), the insertion subtask can be optimized. Since the insertion subtask is the most critical assembly phase, its optimization has been divided in two sub-optimizations.

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 $x_{f,\psi}^{d,\#2} \pm 5^\circ$ for the ψ DoF. The proposed ranges exploit the Cartesian pose $\mathbf{x}_f^{d,\#2}$ selected by the HMM in Section V-B. In such a sub-optimization, it is possible to find the correct assembly location to perform the insertion. The cost function $J_{BO}^{\#2}$ observed values over the optimization iterations are shown in Figure 5. 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 (2) (*i.e.*, safety emergency stop or subtask failure). 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 $\mathbf{D} = 2\boldsymbol{\varepsilon}_D\sqrt{\mathbf{M}\mathbf{K}^{\#2}}$, where $\boldsymbol{\varepsilon}_D$ 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, $x_\psi^{d,\#2,opt_1} \pm 1^\circ$ 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 V-B, 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 Figure 5. 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 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 subtask optimization in order to define the optimization variables ranges.

2) Pushing subtask optimization: Exploiting the cost function $J_{BO}^{\#3}$ proposed in (9), 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).



sub-optimization 1 sub-optimization 2

Fig. 5: Cost function $J_{BO}^{\#2}$ values over the experimental iterations for the first sub-optimization and the second sub-optimization of the insertion subtask. The best iteration is marked by the red circle.

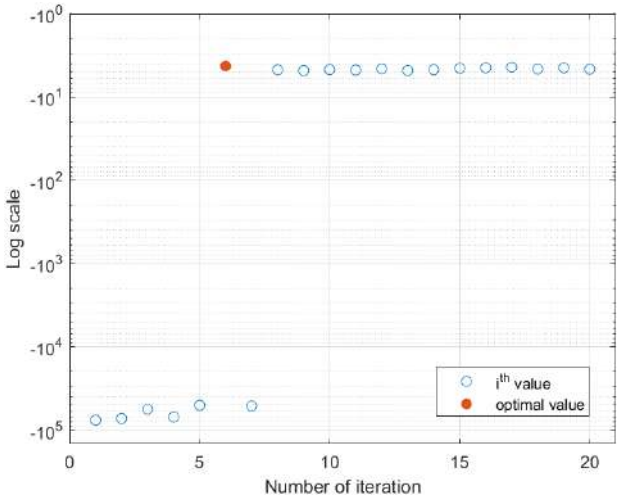


Fig. 6: Cost function $J_{BO}^{\#3}$ values over the experimental iterations for the optimization of the pushing subtask. The best iteration is marked by the red circle.

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 such optimization are imposed equal to 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 V-B. The cost function $J_{BO}^{\#3}$ observed values over the optimization iterations are shown in Figure 6. 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 (2) (*i.e.*, safety emergency stop or subtask failure). The pushing subtask optimization returns the optimized parameters $x_z^{d,\#3,opt}$, $K_z^{\#3,opt}$, and $\alpha^{\#3,opt}$.

Remark 3. 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.

E. Autonomous assembly task execution evaluation

After the proposed HMM+BO procedure has been executed, the autonomous assembly task execution can be evaluated. The proposed approach has been compared with the HMM approach (*i.e.*, without the optimization stage). For each of the 15 subject-teached tasks, 10 repetitions of the assembly task have been performed. Therefore, 150 repetitions of the assembly task have been executed for both the HMM+BO approach and for the HMM approach. The HMM+BO approach has resulted in a successful execution rate of 93%, while the HMM algorithm has resulted in a successful execution rate of 19%. The optimization stage is therefore required in order to compensate for task uncertainties.

VI. Conclusions

The presented paper proposed machine learning techniques to make a sensorless robot able to learn and optimize an industrial assembly task. Exploiting sensorless Cartesian impedance control, a task-trajectory learning algorithm (exploiting Hidden Markov Model) and an autonomous task optimization procedure (exploiting Bayesian Optimization) are derived. To

validate the proposed methodology, an assembly task of a gear into its square-section shaft has been selected. A Franka EMIKA Panda manipulator has been used as a test platform. Experimental results show a task success rate of 93% for the HMM+BO approach, w.r.t. a task success rate of 19% for the HMM approach, highlighting the capabilities of the proposed approach to compensate for task uncertainties.

The human demonstrations-based procedure to learn the nominal task trajectories is shown to be very fast, requiring only few demonstrations (5 demonstrations in the paper) to train the robot. This means that $5-10$ minutes are enough to accomplish this task. The optimization procedure has been performed as three subsequent optimizations, requiring in total 140 iterations. It is shown that approximately half an hour is needed to perform the complete optimization procedure, that has been completely automatized.

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

REFERENCES

[1] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014.

[2] 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, and G. Fogliazza, *PIROS: Cooperative, Safe and Reconfigurable Robotic Companion for CNC Pallets Load/Unload Stations*. Cham: Springer International Publishing, 2020, pp. 57–96. [Online]. Available: https://doi.org/10.1007/978-3-030-34507-5_4

[3] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[4] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.

[5] A. S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 153–173, 2017.

[6] A. Mosavi and A. Varkonyi, "Learning in robotics," *International Journal of Computer Applications*, vol. 157, no. 1, pp. 8–11, 2017.

[7] J. Peters, "Machine learning for motor skills in robotics," *KI-Künstliche Intelligenz*, vol. 2008, no. 4, pp. 41–43, 2008.

[8] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[9] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6023–6029.

[10] P. Van Molle, T. Verbelen, E. De Coninck, C. De Boom, P. Simoens, and B. Dhoedt, "Learning to grasp from a single demonstration," *arXiv preprint arXiv:1806.03486*, 2018.

[11] 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 Intelligent Systems*, vol. 32, no. 4, pp. 6–21, 2017.

[12] L. Roveda, G. Pallucca, N. Pedrocchi, F. Braghin, and L. M. Tosatti, "Iterative learning procedure with reinforcement for high-accuracy force tracking in robotized tasks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1753–1763, 2017.

[13] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," *CoRR*, vol. abs/1501.05611, 2015.

[14] G. Thomas, M. Chien, A. Tamar, J. A. Ojea, and P. Abbeel, "Learning robotic assembly from CAD," *CoRR*, vol. abs/1803.07635, 2018.

[15] A. Vakanski, I. Mantegh, A. Irish, and F. Janabi-Sharifi, "Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1039–1052, Aug 2012.

[16] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *ICML*, 1997.

[17] A. Billard and D. Grollman, "Robot learning by demonstration," *Scholarpedia*, vol. 8, no. 12, 2013.

[18] S. Calinon, F. Guenter, and A. Billard, "On learning, representing and generalizing a task in a humanoid robot," *IEEE Trans. Syst. Man Cybernet*, vol. 37, pp. 286–298, 2007.

[19] J. Xu, Z. Hou, Z. Liu, and H. Qiao, "Compare contact model-based control and contact model-free learning: A survey of robotic peg-in-hole assembly strategies," *arXiv preprint arXiv:1904.05240*, 2019.

[20] G. Thomas, M. Chien, A. Tamar, J. A. Ojea, and P. Abbeel, "Learning robotic assembly from cad," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–9.

[21] A. Abbas, F. D. Maire, F. Dayoub, and S. Shirazi, "Combining learning from demonstration and search algorithm for dynamic goal-directed assembly task planning," 2018.

[22] Y. Fan, J. Luo, and M. Tomizuka, "A learning framework for high precision industrial assembly," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 811–817.

[23] Z. Hou, M. Philipp, K. Zhang, Y. Guan, K. Chen, and J. Xu, "The learning-based optimization algorithm for robotic dual peg-in-hole assembly," *Assembly Automation*, 2018.

[24] M. P. Polverini, A. M. Zanchettin, S. Castello, and 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.

[25] K. Kaipa, C. Morato, B. Zhao, and 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.

[26] B. Siciliano and L. Villani, *Robot Force Control*, 1st ed. Norwell, MA, USA: Kluwer Academic Publishers, 2000.

[27] A. Vakanski, I. Mantegh, A. Irish, and F. Janabi-Sharifi, "Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 4, pp. 1039–1052, 2012.

[28] L. Durak and O. Arikan, "Short-time fourier transform: two fundamental properties and an optimal implementation," *IEEE Transactions on Signal Processing*, vol. 51, no. 5, pp. 1231–1242, May 2003.

[29] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[30] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[31] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[32] L. Biagiotti and C. Melchiorri, *Trajectory Planning for Automatic Machines and Robots*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[33] E. Brochu, V. M. Cora, and N. De Freitas, "A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning," *arXiv preprint:1012.2599*, 2010.