

A Robot Learning from Demonstration Framework for Skillful Small Parts Assembly

Haopeng Hu (✉ haopeng_hu@sina.com)

Harbin Institute of Technology Shenzhen <https://orcid.org/0000-0002-4069-6431>

Xiansheng Yang

Harbin Institute of Technology Shenzhen

Yunjiang Lou

Harbin Institute of Technology Shenzhen

Research Article

Keywords: Flexible Manufacturing, Learning from Demonstration, Robotic Assembly

Posted Date: March 11th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-281027/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A Robot Learning from Demonstration Framework for Skillful Small Parts Assembly

Haopeng Hu · Xiansheng Yang · Yunjiang Lou

Received: date / Accepted: date

Abstract Increasing demand for higher production flexibility and smaller production batch size pushes the development of manufacturing expertise towards robotic solutions with fast setup and reprogram capability. Aiming to facilitate assembly lines with robots, the learning from demonstration (LfD) paradigm has attracted attention. A robot LfD framework designed for skillful small parts assembly applications is developed, which takes position, orientation and wrench demonstration data into consideration while utilizes impedance control to deal with the motion error. In view of constraints in industrial assembly applications, we propose a robot LfD framework where policy learning is carried out with separated assembly demonstration data to avoid potential under-fitting problem. With the proposed assembly policies, reference orientation and wrench trajectories are generated as well as coupled with the position data to boost their generalization and robust performance. Effectiveness of the proposed LfD framework is validated by a printed circuit board assembly experiment with a 7-DOF torque-controlled robot.

Keywords Flexible Manufacturing · Learning from Demonstration · Robotic Assembly

1 Introduction

For the sake of adapting to the ongoing shift from mass production to mass customization of products, man-

ufacturing enterprises have to keep evolving their assembly lines to handle more product variation, shorter product life cycles, and smaller batch sizes [5, 18, 21], i.e. developing flexible assembly lines. Current industrial robots are notoriously difficult to program, leading to high change-over time and expert labor consumption. To make robotic assembly applicable to low-volume high-mix production, one approach is to make the programming of assembly tasks so intuitive that can be accomplished by workers without traditional robot programming skills [26, 29]. Learning from demonstration (LfD) [23], or programming by demonstration [9], is a paradigm that aims to transfer human's skills to robots by human demonstration instead of the unintuitive and tedious robot programming by expert robot users [10, 30]. The strength of LfD pays in industrial assembly applications as some assembly processes are such complicate that can be neither easily scripted nor easily defined as an optimization problem, but can be easily demonstrated [28]. With LfD, one or more assembly policies can be learned from the demonstration data which can also generate robots' actions under new conditions i.e. enable robots' adaptive behavior [34].

Considering the assembly tasks requiring skillful motions [28], to accomplish those tasks with robots, two constraints must be taken into account, namely the *object constraint* and the *environment constraint*. The object constraint includes the assembly motion (position and orientation trajectories) required and the corresponding force and torque pattern throughout the assembly process. It is totally defined by the design of product, i.e. it allows of little variation in assembly motion. In the industrial fields, the environment constraint is introduced by the deployment of industrial agents such as placement of robots, sensors and conveyers. It varies from line to line, or even from day to

This work was supported in part by NSFC-Shenzhen Robotics Basic Research Center Program Grant U1713202.

Y. Lou
School of Mechanical Engineering and Automation, Harbin
Institute of Technology Shenzhen, Shenzhen, China.
E-mail: louyj@hit.edu.cn

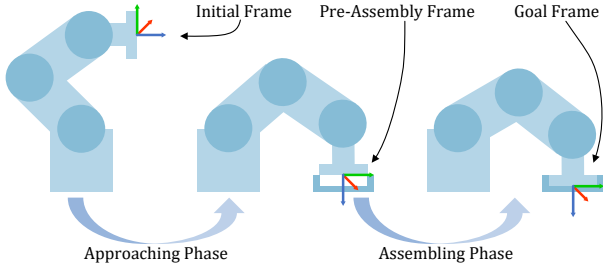


Fig. 1 General view of approaching and assembling phases.

day in one reconfigurable assembly line [18] and allows of large variation in the assembly motion of robots. In view of the two constraints, we divide a robotic assembly process into two phases, namely the *approaching phase* during which the robot takes one part of some product to some specific pose (position and orientation) in contact with the other part, and the *assembling phase* during which the robot assembles the two parts subject to the object constraint. Fig. 1 demonstrates the proposed two successive assembly phases. In the approaching phase, the learned assembly policy is required to drive the robot to the user-defined *pre-assembly pose*, which may be estimated by sensors during execution, from any *initial pose* without unexpected collision against any agents in the environment. Since the environment constraint may vary frequently in practice, generalization capability of the learned assembly policy significantly matters in this phase. As indicated by Fig. 1, the *pre-assembly pose* is the pose where one part just comes in contact with the other part. It is defined by the human operators in demonstration process but estimated by the sensor in robot assembly process. In the assembling phase, only the object constraint affects the motion of robots. Position and orientation variation in demonstration may be relatively small which alleviates the generalization performance requirement of the assembly policy. However, there is a force and torque constraint for the robot to successfully assemble the parts.

In the literature, some of the LfD studies upon assembly applications formulate the assembly tasks as pick-and-place processes [6, 13, 15, 19]. Given the demonstration data, a pre-structured policy is learned through supervised [13] or unsupervised [6, 15] machine learning algorithm. Those policies can be classified into *trajectory primitives*, such as the dynamic movement primitive (DMP), and *location primitives*, such as the Gaussian mixture model (GMM) [15]. Given the novel initial and goal pose of the robot, both of them can generate suitable robot assembly motion trajectories. In view of the pick-and-place processes only one policy is learned for the whole assembly process therein, never-

theless, it may fail to work in the small parts assembly tasks since features of the assembly motion vary significantly between the two phases. Another formulation of small parts assembly tasks is peg-in-hole problem [14, 16, 26] and Reinforcement Learning (RL) based methods are widely applied to learn a decision making policy that maps states to actions through trial-and-error [7, 8, 12, 27]. RL-based approaches typically require the robot to explore the state space. Methods that directly learn the robot control policies from demonstration data are more practicable [22]. Another class of solution to precise assembly is the compliance-based assembly strategy [16, 28]. By compliant control strategies such as impedance control, the robots can perform adaptive behavior in precise assembly tasks, e.g. automatic position and orientation error compensation.

Concentrating on the assembly policy learning and execution, a robot LfD framework is tailored for skillful small parts assembly with impedance control strategy in this work. By exploiting the stochastic structure of task-parameterized Gaussian mixture model (TP-GMM), two policies in Cartesian space sharing similar structure are learned from segmented demonstrations for the approaching and assembling phases, respectively. Impedance control strategy in Cartesian space is utilized to drive the robot following the assembly trajectory generated by the learned policies. It helps to avoid the tedious searching time of learning from scratching methods and follow the required wrench profile during the precise assembly process. Contributions of this work go as follows. A robot LfD framework for assembly applications that governs an assembly task as two assembly phases is developed with two policies that coupled position, orientation and force-torque (wrench) are learned separately to avoid the under-fitting problem. The assembly policies are designed to couple the orientation and wrench information with the generated position trajectories which enjoys more robustness over the unique phase variable trick and enables online trajectory adaptation.

Remained content of this article is organized follows. Section 2 expounds the proposed robot LfD assembly framework from system overview (2.1) to policy learning (2.2) and then assembly task execution with impedance control (2.3). Experiment verification of it lies in section 3. Section 4 concludes this article.

2 Methodology

2.1 Framework Overview

Work flow of the proposed robot LfD assembly framework is exhibited in Fig. 2. A typical LfD framework al-

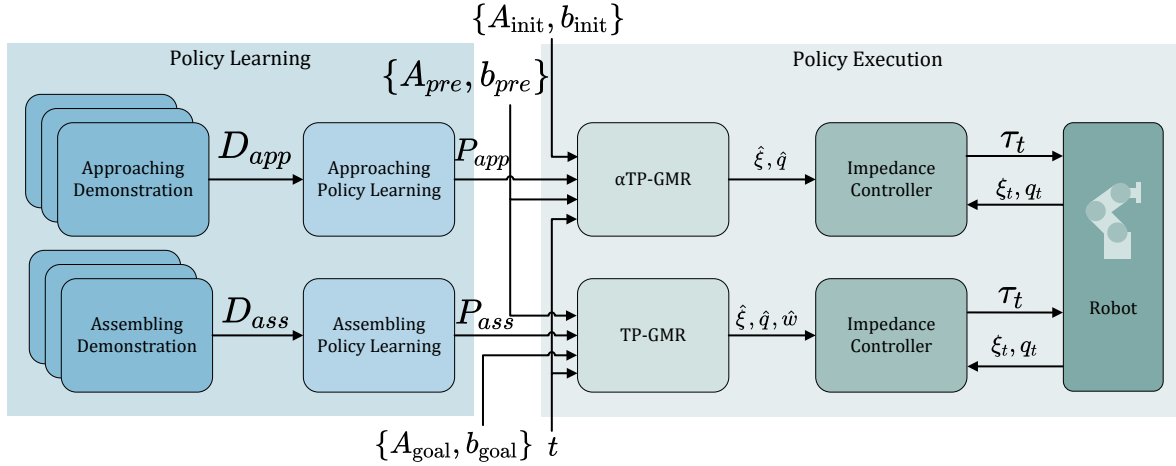


Fig. 2 Work flow of the robot LfD assembly framework.

ways contains three work steps, namely demonstration, policy learning and policy execution (under strange environments), that is, the first decision to be made when designing a LfD framework is which demonstration technique will be used given specific applications. Here the *kinesthetic teaching* technique is selected for the sake of its usability for non-expert robot users. During demonstration, the operator successively performs the approaching motion and assembling motion. On accounting of its capability of modeling latent key location states and generalization, assembly policies in this work are pre-structured by TP-GMM whose parameters are estimated by EM algorithm. In fact, as indicated in Fig. 1 numerical range of demonstrated position data in approaching phase may be hundreds of millimeters but that in assembling phase falls to several millimeters, which leads to a dilemma in the policy learning process. Once it is learned by EM algorithm with this demonstration data, the assembly policy could encapsulate a large amount of redundant latent location states or fail to encapsulate the necessary latent location states in the assembling phase. It is hard to balance the efficiency and precision requirement for us to assign the hyper parameters of the policy, let alone non-expert workers. Hence, in this work two parametric policies pre-structured by TP-GMM are learned respectively to model the two phases of the demonstrated assembly skill and generate the assembly motion trajectory by α TP-GMR. As for robots' execution of the policies given the initial, pre-assembly and goal pose, a Cartesian impedance controller serves to enact those policies.

2.2 Assembly Policy

By an assembly policy, what we mean here is an assembly skill model that takes the initial pose, goal pose and

query variable as inputs while generates a suitable robot motion trajectory as output. Given multiple demonstration trajectories of one task, a stochastic model of the skill is learned by taking into account the variations and correlations observed along the movement. In assembly tasks, both position and orientation play important roles while the force and torque information (wrench) also matters. Policies related to position, orientation and wrench are separately learned but coupled in this work.

2.2.1 Position Policy

To model the position trajectory $\xi = [t, x, y, z]^T$, the task-parameterized GMM is used to pre-structure assembly policies which takes the time variable as query. For the purpose of generating the trajectory distribution $P(\xi)$, a typical GMM takes the form of

$$P(\xi) = \sum_{k=1}^K \pi_k \mathcal{N}(\xi | \mu_{\xi,k}, \Sigma_{\xi,k})$$

where π_{ξ} is the *prior* of each multivariate Gaussian distribution with $\sum_{k=1}^K \pi_{\xi,k} = 1$, μ_{ξ} and Σ_{ξ} are centers and covariances of the k th Gaussian distribution. By extending the typical GMM to multiple task frames to enable generalization, parameters of a TP-GMM model is given by

$$\left\{ \pi_{\xi,k}, \left\{ \mu_{\xi,k}^{(j)}, \Sigma_{\xi,k}^{(j)} \right\}_{j=1}^J \right\}_{k=1}^K \quad (1)$$

where $\mu_k^{(j)} \in \mathbb{R}^4$ and $\Sigma_k^{(j)} \in \mathbb{R}^{4 \times 4}$ are centers and covariances of Gaussian distribution k in frame j respectively. The so-called *task parameters* are given by the J

task frames $\{A_j, b_j\}_{j=1}^J$ that are time-invariant in this work and practical industrial assembly tasks,

$$A_j = \begin{pmatrix} 1 & \mathbf{0} \\ \mathbf{0} & R_j \end{pmatrix}, \quad b_j = \begin{pmatrix} 0 \\ p_j \end{pmatrix}$$

in which $R_j \in SO(3)$ and $p_j \in \mathbb{R}^3$ are rotation matrix and translation vector of frame j with respect to the world frame. Given the demonstration data $D : \{\xi_n\}_{n=1}^N$, rather than directly used to estimate the parameters, it is transformed to J frames firstly through

$$\zeta_n^{(j)} = A_j^{-1}(\xi_n - b_j)$$

Parameters in (1) are estimated in each frame individually by Expectation-Maximization algorithm in which the joint distribution of data in all frames are used to compute the expectation (E-Step) but parameters are estimated only with the data in corresponded frame (M-Step).

By taking time/phase variable t as *query variable*, given the task frames, a retrieved or generalized assembly motion trajectory will be computed through TP-GMR algorithm [3] as described in Fig. 2. Anyway, generalization (interpolation and extrapolation) capability of the approaching policy counts for much more than that of the precise assembly policy as discussed in section 1. In order to further boost the generalization capability of the approaching policy P_{app} , i.e. to preserve the *local structure* in demonstration around the approaching frame, the frame-weighted TP-GMR algorithm (α TP-GMR) [25] is utilized here instead of the typical one. In the j -th frame, the Gaussian center μ and covariace Σ can be partitioned as

$$\mu_{\zeta,k}^{(j)} = \begin{pmatrix} \mu_{t,k}^{(j)} \\ \mu_{\zeta_{\mathcal{O},k}}^{(j)} \end{pmatrix}, \quad \Sigma_{\zeta,k}^{(j)} = \begin{pmatrix} \Sigma_{t,k}^{(j)} & \Sigma_{t\zeta_{\mathcal{O},k}}^{(j)} \\ \Sigma_{\zeta_{\mathcal{O},k}t}^{(j)} & \Sigma_{\zeta_{\mathcal{O},k}}^{(j)} \end{pmatrix}$$

and the trajectory distribution in the j th frame generated by GMR is given by

$$P(\hat{\zeta}_{\mathcal{O},n}^{(j)} | t_n) = \mathcal{N} \left(\sum_{k=1}^K \phi_k(t_n) \hat{\zeta}_{\mathcal{O},k,n}^{(j)}, \sum_{k=1}^K \phi_k^2(t_n) \hat{\Sigma}_{\zeta_{\mathcal{O},k,n}}^{(j)} \right) \quad (2)$$

where

$$\begin{aligned} \hat{\zeta}_{\mathcal{O},k,n}^{(j)} &= \mu_{\zeta_{\mathcal{O},k}}^{(j)} + \Sigma_{\zeta_{\mathcal{O},k}t}^{(j)} \left(\Sigma_{t,k}^{(j)} \right)^{-1} (t_n - \mu_{t,k}^{(j)}) \\ \hat{\Sigma}_{\zeta_{\mathcal{O},k,n}}^{(j)} &= \Sigma_{\zeta_{\mathcal{O},k}}^{(j)} - \Sigma_{\zeta_{\mathcal{O},k}t}^{(j)} \left(\Sigma_{t,k}^{(j)} \right)^{-1} \Sigma_{t\zeta_{\mathcal{O},k}}^{(j)} \\ \phi_k(t_n) &= \frac{\pi_k \mathcal{N}(t_n | \mu_{t,k}, \Sigma_{t,k})}{\sum_{l=1}^K \pi_k \mathcal{N}(t_n | \mu_{t,l}, \Sigma_{t,l})} \end{aligned}$$

Then the generated trajectory distribution in the world frame is estimated by

$$\mathcal{N}(\hat{\xi}_{\mathcal{O},n}, \hat{\Sigma}_{\zeta_{\mathcal{O},n}}) \propto \prod_{j=1}^J \mathcal{N} \left(R_j \hat{\zeta}_{\mathcal{O},n}^{(j)} + p_j, \frac{1}{\alpha_n^{(j)}} R_j \hat{\Sigma}_{\zeta_{\mathcal{O},n}}^{(j)} R_j^T \right)$$

in which $\hat{\xi}_{\mathcal{O}}$ and $\hat{\zeta}_{\mathcal{O}}^{(j)}$ are *queried variable* by GMR in world frame and frame j respectively ($[x, y, z]^T$ in this work), and $\hat{\Sigma}_{\zeta_{\mathcal{O}}}$ stands for the corresponded covariance. $\alpha_n^{(j)}$ indicates the *weight* of frame j at step n ,

$$\alpha_n^{(j)} = \frac{\left\| \left(\tilde{\Sigma}_n^{(j)} \right)^{-1} \right\|}{\sum_{l=1}^J \left\| \left(\tilde{\Sigma}_n^{(l)} \right)^{-1} \right\|} \quad (3)$$

$\tilde{\Sigma}_n^{(j)}$ in (3) is the covariance of demonstration data in frame j at time step n , computation of which costs additional effort but only required once after demonstration. As for the assembling policy P_{cis} , typical TP-GMR algorithm ($\alpha_n^{(j)} = 1$) will work well since there is little variation in the profile of position trajectories in assembling phase.

2.2.2 Orientation Policy

As one of the key aspects, it is inevitable to take the orientation trajectories required in many assembly tasks. However, the challenge lies in the fact that measuring the difference between two orientations with Euclidean distance is not proper as we do in position difference measurement. In the literature, unit quaternion is a popular representation of orientation in LfD studies [11, 13, 15], which is a kind of non-minimal representation defined on an unit sphere manifold \mathcal{S}^3 whose tangent space $\mathcal{T}_q \mathcal{S}^3$ locally linearizes the manifold and behaves like Euclidean space \mathbb{R}^3 [4]. Compared with minimal representation of orientation such as Euler angle, it benefits from no singularity. Taking the orientation of pre-assembly or goal pose in demonstration as *auxiliary quaternion* q_a [11], difference between unit quaternions can be measured by the Euclidean length of the displacement of unit quaternions $q_1 * \bar{q}_a \triangleq [v, \mathbf{u}]$, $v \in \mathbb{R}$, $\mathbf{u} \in \mathbb{R}^3$ mapped to the tangent space $\mathcal{T}_{q_a} \mathcal{S}^3$ centered at q_a through the *logarithmic map* $\mathcal{S}^3 \rightarrow \mathcal{T}_{q_a} \mathcal{S}^3$

$$\log(q_1 * \bar{q}_a) = \begin{cases} \arccos(v) \frac{\mathbf{u}}{\|\mathbf{u}\|}, & \|\mathbf{u}\| \neq 0 \\ [0, 0, 0]^T, & \text{otherwise} \end{cases} \quad (4)$$

where \bar{q} stands for the *conjugate* of q . Recall that $q_1 \triangleq [v_1, \mathbf{u}_1]$, $q_2 \triangleq [v_2, \mathbf{u}_2]$,

$$\mathbf{q}_1 * \mathbf{q}_2 = v_1 v_2 - \mathbf{u}_1 \mathbf{u}_2 + v_2 \mathbf{u}_1 + v_1 \mathbf{u}_2 + \mathbf{u}_1 \times \mathbf{u}_2$$

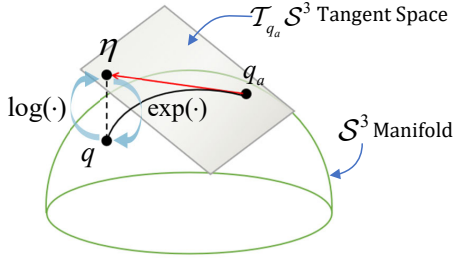


Fig. 3 Graphic representation of the relationship between q and η .

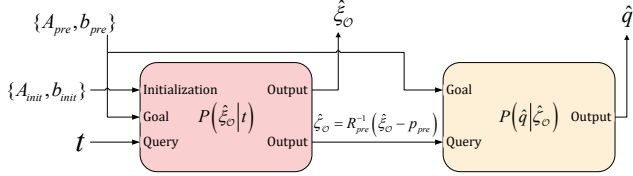


Fig. 4 Structure of the approaching policy.

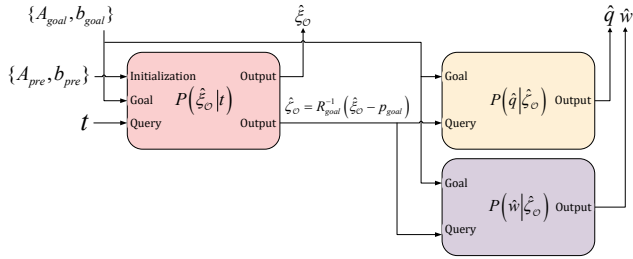


Fig. 5 Structure of the precise assembly policy.

Since the unit quaternion manifold \mathcal{S}^3 provides a double covering over rotations, we set $q = -q$ when $v < 0$ to ensure zero distance between antipodal rotations [33]. Afterwards, policy learning (EM algorithm) and trajectory generation (GMR) processes are carried out on $\mathcal{T}_{q_a} \mathcal{S}^3$ with Euclidean distance. Finally, orientation trajectories generated by the GMR ought to be projected back to the unit sphere manifold \mathcal{S}^3 via the *exponential map* $\mathcal{T}_{q_a} \mathcal{S}^3 \rightarrow \mathcal{S}^3$

$$\exp(\eta) = \begin{cases} \cos(\|\eta\|) + \sin(\|\eta\|) \frac{\eta}{\|\eta\|}, & \eta \neq 0 \\ [1, 0, 0, 0]^T, & \text{otherwise} \end{cases} \quad (5)$$

Fig. 3 exhibits the relationship between q and η . Note that both (4) and (5) are defined with respect to the same auxiliary unit quaternion q_a . It is obvious that the *unit constraint* of unit quaternions will never be violated with map (5). In many assembly tasks, the coupling of position and orientation must be taken into consideration. In \mathcal{S}^3 manifold, it is also feasible to make use of TP-GMM for generalization to new goal orientation with TP-GMR taking time/phase variable as query [33]. By doing so, translation and rotation movements

are coupled by the unique query variable, similar to how the phase variable works in DMP [13]. However, it is evident that during approaching phase orientation of the end-effector does not matter in the early phase but does matter around the vicinity of approaching pose. Therefore, in this work the positions with respect to the pre-assembly and goal frame act as query for retrieving or generalizing the trajectories of orientation, i.e. rotation and translation movements are coupled by the covariance of multi-variate Gaussian distributions. Compared with coupling by the unique query variable, strength of the proposed coupling strategy lie in the fact that the robot may not exactly follow the generated assembly trajectories owing to the performance of controller, noise of sensors and perturbation, which makes online adaptation of orientation plays a valuable role. By doing so, we estimate the end-effector's orientation given the current position rather than current time/phase. Fig. 4 and 5 indicate the proposed coupling technique and describe the structure of the approaching policy and assembling policy respectively. Taking the output of (2) in pre-assembly frame or goal frame as query variable, state of the GMM is defined as $\varphi \triangleq [\zeta_O, \eta]^T$, i.e. parameters of the GMM are $\{\pi_{\varphi,k}, \mu_{\varphi,k}, \Sigma_{\varphi,k}\}_{k=1}^K$. Similar to the (2), orientation trajectory distribution generated by GMR turns out to be

$$P(\hat{\eta}_n | \hat{\zeta}_{O,n}) = \mathcal{N} \left(\sum_{k=1}^K \phi_k(\hat{\zeta}_{O,n}) \hat{\eta}_{k,n}, \sum_{k=1}^K \phi_k(\hat{\zeta}_{O,n})^2 \hat{\Sigma}_{\eta,k,n} \right) \quad (6)$$

$$\hat{q}_n = \exp(\hat{\eta}_n) * q_a$$

where $\hat{\eta}_n$, $\hat{\Sigma}_{\eta,k,n}$, $\phi_k(\cdot)$ are counterparts of those in (2) and the number of Gaussian distributions K may not be the same as that in (1).

2.2.3 Wrench Policy in Assembling Phase

A typical assembly operation requires the knowledge of not only position and orientation trajectories but also the accompanying wrench profiles for successful assembly [13]. There is no probabilistic model in terms of the wrench w in the approaching policy since the end-effector of the robot is moving through the air during this phase as indicated by Fig. 1. In assembling phase, similar to [17], the wrench pattern is modeled by GMM. As shown in Fig. 5, the wrench policy also takes position with respect to the goal frame as query variable as shown in Fig. 5. Wrench trajectory distribution generated by GMR shares the similar policy structure and

coupling strategy as (6) just by replacing $\hat{\eta}$ by \hat{w}

$$P(\hat{w}_n | \hat{\zeta}_{\mathcal{O},n}) = \mathcal{N} \left(\sum_{k=1}^K \phi_k(\hat{\zeta}_{\mathcal{O},n}) \hat{w}_{k,n}, \sum_{k=1}^K \phi_k(\hat{\zeta}_{\mathcal{O},n})^2 \hat{\Sigma}_{w,k,n} \right) \quad (7)$$

Anyway, there is no need to exactly follow the generated wrench trajectory during assembly processes [13]. In this work, the queried wrench trajectory distribution (7) plays two roles. First and foremost, it provides a reference wrench pattern for successful execution of the task since the parts require enough wrench to be assembled. Secondly, it also serves as an indicator of possible damaging of the parts. Variability of the demonstrations is encapsulated in the covariance matrices of the trajectory distribution which can be exploited to detect if the robot reaches an unexpected pose [24]. By eigenvalue decomposition of covariance matrix $\hat{\Sigma}_{w,n} = V \Lambda_{w,n} V^T$, each eigenvalue $\lambda_{i,t}$ corresponds to the allowable variability of the force/torque term at time/phase t_n . In the experiment, we stop motion of the robot once any one dimension of the current external wrench $w_{i,t}$ exceeds the limit

$$[\hat{w}_{i,t} - \beta|\lambda_{i,t}|, \hat{w}_{i,t} + \beta|\lambda_{i,t}|], \beta > 1$$

2.3 Cartesian Impedance Controller

Assembly policies in this work can only provide assembly trajectories, which calls for a suitable controller to drive the robot to track. As in LfD studies [17, 34], *impedance control* is widely applied control strategy due to its simplicity and flexibility. In assembly tasks that require physical interaction between a robot and its surroundings, using impedance control helps overcome position/orientation uncertainties and subsequently avoid large impact forces [1, 31]. Taking both position and orientation into consideration, an impedance controller with joint torque τ as control input turns out to be

$$\tau = J^T \left(K \begin{bmatrix} \hat{\xi}_{\mathcal{O}} - \xi_{\mathcal{O}} \\ \log(\hat{q} * \bar{q}) \end{bmatrix} - D \begin{bmatrix} v \\ \omega \end{bmatrix} \right) + J^T \hat{w} + h(\theta, \dot{\theta}) \quad (8)$$

where

$$K = \begin{pmatrix} K^{pos} & \mathbf{0} \\ \mathbf{0} & K^{ori} \end{pmatrix}, D = \begin{pmatrix} D^{pos} & \mathbf{0} \\ \mathbf{0} & D^{ori} \end{pmatrix}$$

are semi-positive definite matrices of impedance parameters. Compliance of robot's assembly motion can be realized through regulating the impedance parameters, i.e. the apparent damping $D^{pos}, D^{ori} \in \mathbb{R}^{3 \times 3}$ and stiffness $K^{pos}, K^{ori} \in \mathbb{R}^{3 \times 3}$. J in (8) is the Jacobian and

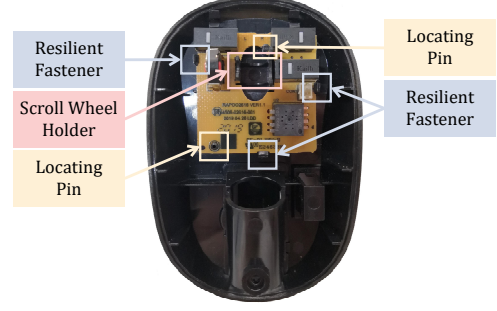


Fig. 6 The PCB and bottom case of a mouse.

$h(\cdot)$ represents dynamic model of the robot that allows for compensation of gravity, Coriolis force and friction. v and ω stand for linear and angular velocities of the robot's end-effector. $J^T \hat{w}$ serves as a feed forward term. In approaching phase, compared with compliance what makes greater sense for the controller is driving the robot to the approaching pose as soon as possible without any unexpected collision, that is, efficiency dominates. Hence, in this phase, high stiffness is assigned to the controller and there is no feed forward term in the controller (8). Moreover, the assembling phase, which is an in-contact task, puts an requirement on the compliance of the robot's end-effector due to unavoidable position or orientation error and tight tolerances between the parts. During this phase, stiff execution may not be safe for robots and objects in interaction [20]. Therefore, low stiffness is assigned to the controller in operation to perform compliant assembly which is robust against motion inaccuracy and disturbance [31]. In both phases, damping parameters are specified according to the robot's dynamic limits.

3 Experiment Verification

3.1 Experiment Setup

In this section, a printed circuit board (PCB) [2] assembly task is selected to demonstrate the work flow of the proposed robot LfD assembly framework and verify its effectiveness. As shown in Fig. 6, to successfully assemble the PCB to its bottom case, after plugging it through the scroll wheel holder the robot ought to carry out an insertion motion and then fit the PCB to the locating pins while pressing it to make it fastened by the resilient fasteners. A torque-controlled Franka Emika 7-DOF robot serves as the policy executor while a structured light camera works to acquire the goal frame in the experiment, as exhibited in Fig. 7. The structured light camera is capable of obtaining 6-DOF pose information of the bottom case. The initial pose of the PCB

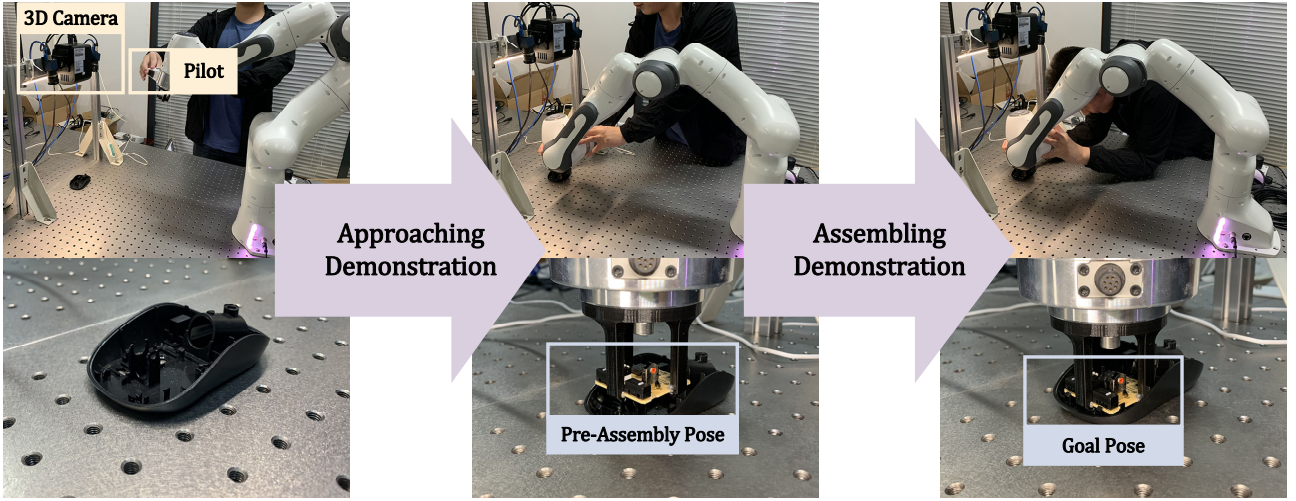


Fig. 7 Kinesthetic guiding demonstration process.

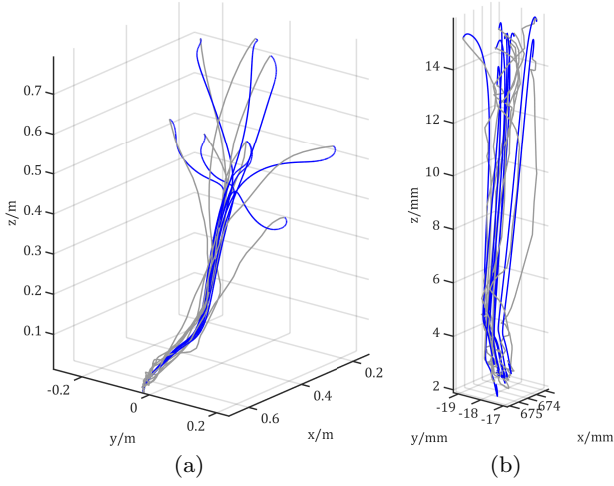


Fig. 8 Demonstrated (grey) and retrieved (blue) position data in approaching phase (a) and assembling phase (b).

is obtained by reading the joint position and the kinematics of the robot. Fig. 7 demonstrates the kinesthetic teaching process. To begin with, the operator kinesthetically guides the robot with gravity compensation by the *pilot* on the robot's 7th joint from any initial pose to the approaching pose (approaching phase). Afterwards, through carefully adjusting the pose of the end-effector to guarantee safety, the operator performs the PCBA task (assembling phase). Issues in terms of grasping the components by robots are beyond our research scope. In both approaching and assembling phases, the number of task frames $J = 3$ in (1) as indicated by Fig. 2. The task frames are defined by the initial, pre-assembly and goal poses. The goal frame in this experiment is estimated via observing the pose of the bottom case with the 3D camera. The pre-assembly frame is computed by the demonstrated transformation from the

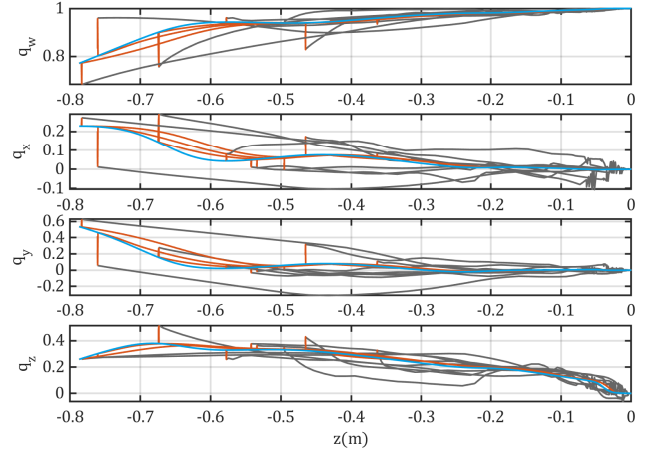


Fig. 9 Demonstrated (grey), retrieved (blue) and recorded (orange) assembly orientation trajectories in approaching phase.

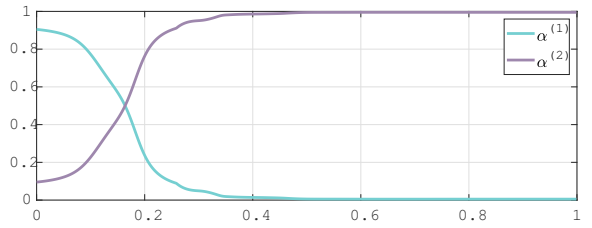


Fig. 10 Weight of each frame during position data retrieving.

goal frame to it. However, there are always slight differences between each demonstrated one, which brings in no trouble in policy learning process but it is required to figure out one approaching frame given the observed goal frame. In the experiment, transformation from goal frame to pre-assembly frame is figured out by taking average of all the approaching frames in demonstration data. Here we take position together with Euler angle

as representation for averaging to estimate the transform $T_{pre,goal} \in SE(3)$. Then, given the observed goal frame \hat{T}_{goal} , the corresponded approaching frame is computed by $\hat{T}_{pre} = T_{pre,goal}\hat{T}_{goal}$. Since human operators can never perform one task multiple times within the same period. *Dynamic time wrapping* is a populous temporal alignment method in LfD studies [15, 30, 32] including this work. In assembling phase, the human operator corrupts the wrench profile estimated by joints torques [13]. To attenuate this effect, in this experiment the robot plays back the demonstrated precise assembly motion to obtain clean wrench demonstration data. In this work, segmentation of demonstration data is performed manually, nonetheless, automatic segmentation is also possible by analyzing velocity and acceleration feature of it [28, 30].

3.2 Experiment Results & Discussion

Demonstrations are performed from randomly selected initial pose to a constant pre-assembly and goal pose, as shown in Fig. 8 and Fig. 9. Demonstrated position data are temporally aligned by DTW in both assembly phases but orientation data are not aligned because the orientation is queried by the position in the pre-assembly or goal frame. In this experiment, rather than absolute time a monotonously increasing phase variable $t \in [0, 1]$ is selected as query variable for position data generation. By TP-GMR, smooth motion trajectories are retrieved without unexpected jagged data around the approaching pose which is introduced by unavoidable trembling of human hands during demonstration. Motion trajectories generated by the policies are fed to the impedance controller at a frequency of 10Hz. It can be seen by Fig. 8 that there is large magnitude of difference between the robot's spatial motion in approaching phase and that in assembling phase, which partially accounts for why it is necessary to segment the demonstration and learn two specific assembly policies. Some key motion in assembling phase may be deemed to be noise once only one policy is used to model both phases. Variation of the motion trajectories decreases gradually since the weight of the pre-assembly frame increases gradually with the robot approaching the pre-assembly pose. Fig. 10 shows the variance of α value in the α TP-GMR process, local structure with respect to the pre-assembly frame dominates the shape of the trajectories when $t \geq 0.4$. Fig. 9 exhibits the demonstrated, retrieved and recorded orientation trajectories in the approaching phase. All of these trajectories converge to the orientation $[1, 0, 0, 0]^T$ because the demonstrated orientation data are displayed with respect to

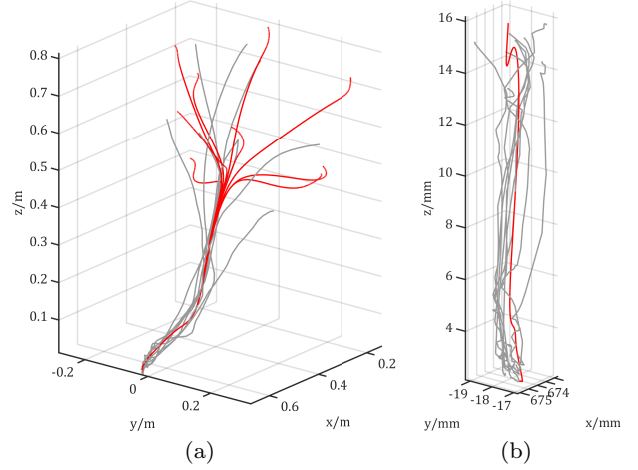


Fig. 11 Demonstrated (grey) and generalized (red) position data in approaching phase (a) and assembling phase (b).

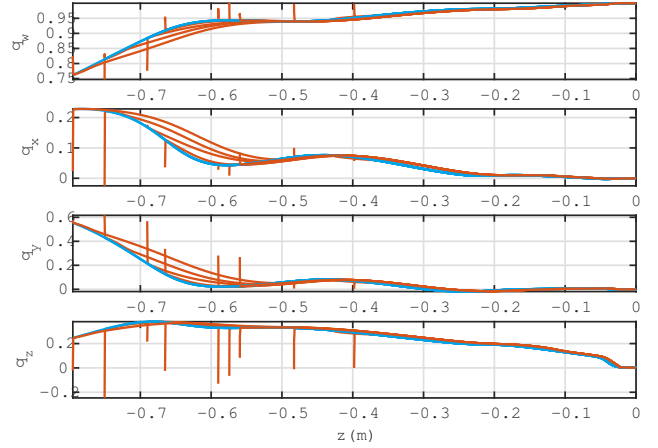


Fig. 12 Generalized (red) and recorded (orange) assembly orientation trajectories in approaching phase.

the pre-assembly frame and then the corresponded η in the tangent space are computed by (4) to learn the GMM based policy. In this experiment, since only z position with respect to the pre-assembly frame is used as query variable of GMR, the retrieved orientation trajectories (blue curves) overlap each other. As shown in Fig. 4, the orientation trajectories are retrieved in advance in this experiment. However, with the proposed coupling strategy, reference orientation can also be queried at any time step given current Cartesian position. This online adaptation technique can deal with unexpected perturbation during the assembly process. As exhibited in Fig. 11, new initial poses are given to verify the generalization capability of the learned assembly policies. Those generalized assembly motion trajectories share the same pre-assembly and goal frame. Similar to the retrieved trajectories in Fig. 8, the generalized position trajectories quickly converge. Since the motion trajectories are only subject to the invariant object constraints

in the assembling phase, typical TP-GMR algorithm works well. Fig. 12 presents the generalized orientation trajectories (red curves) via GMR and the recorded orientation trajectories during assembly process (orange curves). In Fig. 9 and 12, it can be seen that during the approaching phase the recorded orientation trajectories quickly converge to the retrieved/generalized ones almost before the motion of position. This is mainly because the absolute value of the initial orientation error $\log(\hat{q}(0) * \bar{q}(0))$ in (8) is numerically much larger than that of position which is 0.

4 Conclusion

To enable robots quickly acquire new assembly skills in assembly lines, the learning from demonstration paradigm provides a flexible solution to directly transfer workers' assembly skills to robots. A robot LfD framework is developed in this article for skillful industrial assembly applications. In this framework, the assembly task is divided into approaching and assembling phases, which is the key to avoid the potential under-fitting problem introduced by the large difference in variability of demonstration data between the two assembly phases. By exploiting the stochastic nature and generalization capability of TP-GMM, the demonstrated position data are modeled and retrieved through TP-GMR. Corresponded orientation and wrench are also queried by the position via GMM, which turns out to be a effective coupling strategy that enables online adaptation of orientation and wrench. An impedance control strategy is developed to drive the robot to follow the assembly motion trajectories generated by those policies. A PCB assembly task is taken as illustration for the proposed robot LfD assembly framework. It indicates that the learned policies can generate suitable assembly position and orientation trajectories in both phases while the impedance controller is capable of driving the robot to accomplish the assembly task with compliant behavior.

Ethical Approval

Not applicable

Consent to Participate

We consent

Consent to Publish

We consent

Authors Contributions

These authors contributed equally to this work.

Funding

This work was supported in part by NSFC-Shenzhen Robotics Basic Research Center Program Grant U1713202.

Conflict of interest

The authors declare that they have no conflict of interest.

Availability of data and materials

Not applicable

References

1. Abu-Dakka, F.J., Roza, L., Caldwell, D.G.: Force-based variable impedance learning for robotic manipulation. *Robotics and Autonomous Systems* **109**, 156–167 (2018)
2. Andrzejewski, K., Cooper, M., Griffiths, C., Giannetti, C.: Optimisation process for robotic assembly of electronic components. *The International Journal of Advanced Manufacturing Technology* **99**(9), 2523–2535 (2018)
3. Calinon, S.: A tutorial on task-parameterized movement learning and retrieval. *Intelligent Service Robotics* **9**(1), 1–29 (2016)
4. Calinon, S.: Gaussians on Riemannian Manifolds: Applications for Robot Learning and Adaptive Control. *IEEE Robotics Automation Magazine* **27**(2), 33–45 (2020)
5. Cohen, Y., Nasereldin, H., Chaudhuri, A., Pilati, F.: Assembly systems in industry 4.0 era: a road map to understand assembly 4.0. *The International Journal of Advanced Manufacturing Technology* **105**(9), 4037–4054 (2019)
6. Duque, D.A., Prieto, F.A., Hoyos, J.G.: Trajectory generation for robotic assembly operations using learning by demonstration. *Robotics and Computer-Integrated Manufacturing* **57**, 292–302 (2019)
7. Ehlers, D., Suomalainen, M., Lundell, J., Kyrki, V.: Imitating Human Search Strategies for Assembly. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 7821–7827 (2019)
8. Fan, Y., Luo, J., Tomizuka, M.: A Learning Framework for High Precision Industrial Assembly. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 811–817 (2019)

9. Ferreira, M., Costa, P., Rocha, L., Moreira, A.P.: Stereo-based real-time 6-dof work tool tracking for robot programming by demonstration. *The International Journal of Advanced Manufacturing Technology* **85**(1), 57–69 (2016)
10. Gašpar, T., Deniša, M., Radanovič, P., Ridge, B., Savarimuthu, T.R., Kramberger, A., Priggemeyer, M., Roßmann, J., Wörgötter, F., Ivanovska, T., Parizi, S., Gosar, Ž., Kovač, I., Ude, A.: Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells. *Robotics and Computer-Integrated Manufacturing* **66**, 101979 (2020)
11. Huang, Y., Abu-Dakka, F.J., Silvério, J., Caldwell, D.G.: Generalized Orientation Learning in Robot Task Space. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 2531–2537 (2019)
12. Inoue, T., De Magistris, G., Munawar, A., Yokoya, T., Tachibana, R.: Deep reinforcement learning for high precision assembly tasks. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 819–825 (2017)
13. Kramberger, A., Gams, A., Nemec, B., Chrysostomou, D., Madsen, O., Ude, A.: Generalization of orientation trajectories and force-torque profiles for robotic assembly. *Robotics and Autonomous Systems* **98**, 333–346 (2017)
14. Kwak, S.J., Hasegawa, T., Mozas, O.M., Chung, S.Y.: Elimination of unnecessary contact states in contact state graphs for robotic assembly tasks. *The International Journal of Advanced Manufacturing Technology* **70**(9–12), 1683–1697 (2014)
15. Kyrarini, M., Haseeb, M.A., Ristić-Durrant, D., Gräser, A.: Robot learning of industrial assembly task via human demonstrations. *Autonomous Robots* **43**(1), 239–257 (2019)
16. Park, H., Park, J., Lee, D.H., Park, J.H., Baeg, M.H., Bae, J.H.: Compliance-Based Robotic Peg-in-Hole Assembly Strategy Without Force Feedback. *IEEE Transactions on Industrial Electronics* **64**(8), 6299–6309 (2017)
17. Park, S., Lee, W., Chung, W.K., Kim, K.: Programming by Demonstration Using the Teleimpedance Control Scheme: Verification by an sEMG-Controlled Ball-Trapping Robot. *IEEE Transactions on Industrial Informatics* **15**(2), 998–1006 (2019)
18. Pedersen, M.R., Nalpantidis, L., Andersen, R.S., Schou, C., Bøgh, S., Krüger, V., Madsen, O.: Robot skills for manufacturing: From concept to industrial deployment. *Robotics and Computer-Integrated Manufacturing* **37**, 282–291 (2016)
19. Pervez, A., Lee, D.: Learning task-parameterized dynamic movement primitives using mixture of GMMs. *Intelligent Service Robotics* **11**(1), 61–78 (2018)
20. Petrič, T., Gams, A., Colasanto, L., Ijspeert, A.J., Ude, A.: Accelerated Sensorimotor Learning of Compliant Movement Primitives. *IEEE Transactions on Robotics* **34**(6), 1636–1642 (2018)
21. Qu, Y., Ming, X., Liu, Z., Zhang, X., Hou, Z.: Smart manufacturing systems: state of the art and future trends. *The International Journal of Advanced Manufacturing Technology* **103**(9), 3751–3768 (2019)
22. chaandar Ravichandar, H., Dani, A.: Learning position and orientation dynamics from demonstrations via contraction analysis. *Autonomous Robots* **43**(4), 897–912 (2019)
23. Ravichandar, H., Polydoros, A.S., Chernova, S., Billard, A.: Recent Advances in Robot Learning from Demonstration. *Annual Review of Control, Robotics, and Autonomous Systems* **3**(1), 297–330 (2020)
24. Roza, L., Calinon, S., Caldwell, D.G., Jiménez, P., Torras, C.: Learning Physical Collaborative Robot Behaviors From Human Demonstrations. *IEEE Transactions on Robotics* **32**(3), 513–527 (2016)
25. Sena, A., Michael, B., Howard, M.: Improving Task-Parameterised Movement Learning Generalisation with Frame-Weighted Trajectory Generation. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4281–4287 (2019)
26. Sloth, C., Kramberger, A., Iturrate, I.: Towards easy setup of robotic assembly tasks. *Advanced Robotics* **34**(7–8), 499–513 (2020)
27. Thomas, G., Chien, M., Tamar, A., Ojea, J.A., Abbeel, P.: Learning robotic assembly from cad. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 3524–3531. IEEE (2018)
28. Wang, Y., Harada, K., Wan, W.: Motion planning of skillful motions in assembly process through human demonstration. *Advanced Robotics* (2020)
29. Wang, Y., Xiong, R., Yu, H., Zhang, J., Liu, Y.: Perception of Demonstration for Automatic Programming of Robotic Assembly: Framework, Algorithm, and Validation. *IEEE/ASME Transactions on Mechatronics* **23**(3), 1059–1070 (2018)
30. Wierschem, D.C., Jimenez, J.A., Mediavilla, F.A.M.: A motion capture system for the study of human manufacturing repetitive motions. *The International Journal of Advanced Manufacturing Technology* **110**(3), 813–827 (2020)
31. Wirnshofer, F., Schmitt, P.S., Meister, P., v. Wichert, G., Burgard, W.: Robust, Compliant Assembly with Elastic Parts and Model Uncertainty. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 6044–6051 (2019)
32. Yang, C., Zeng, C., Cong, Y., Wang, N., Wang, M.: A Learning Framework of Adaptive Manipulative Skills From Human to Robot. *IEEE Transactions on Industrial Informatics* **15**(2), 1153–1161 (2019)
33. Zeestraten, M.J.A., Havoutis, I., Silvério, J., Calinon, S., Caldwell, D.G.: An Approach for Imitation Learning on Riemannian Manifolds. *IEEE Robotics and Automation Letters* **2**(3), 1240–1247 (2017)
34. Zeng, C., Yang, C., Cheng, H., Li, Y., Dai, S.L.: Simultaneously Encoding Movement and sEMG-based Stiffness for Robotic Skill Learning. *IEEE Transactions on Industrial Informatics* pp. 1–1 (2020)

Figures

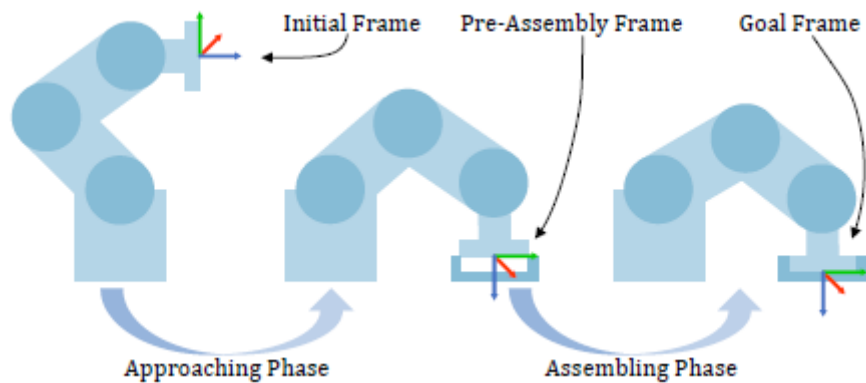


Figure 1

General view of approaching and assembling phases.

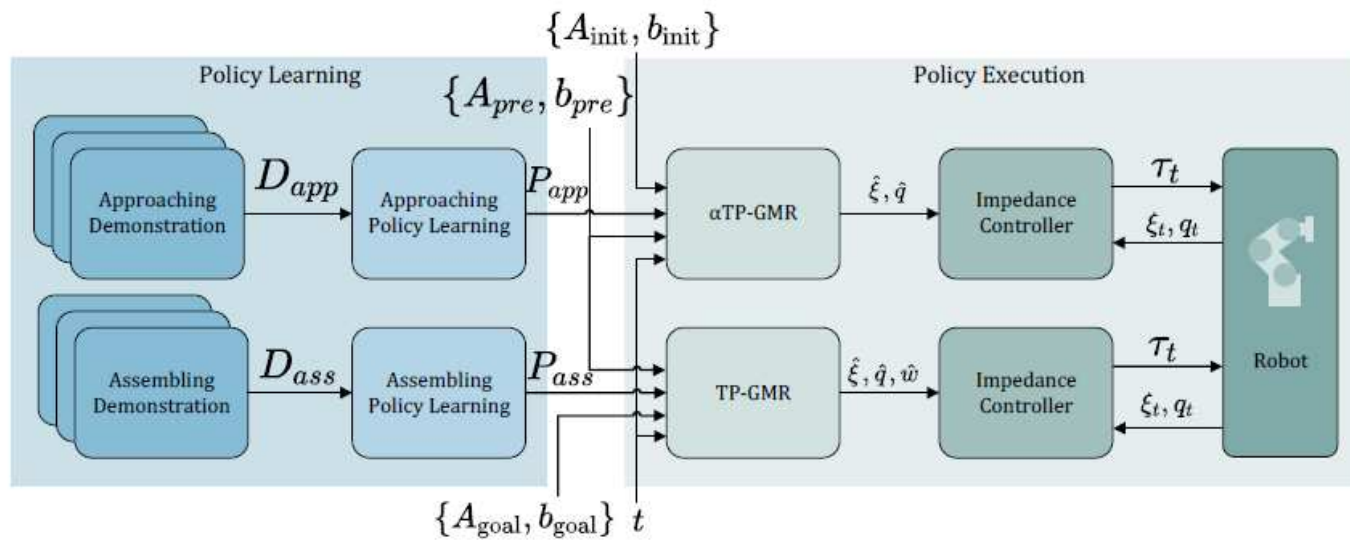


Figure 2

Work flow of the robot LfD assembly framework.

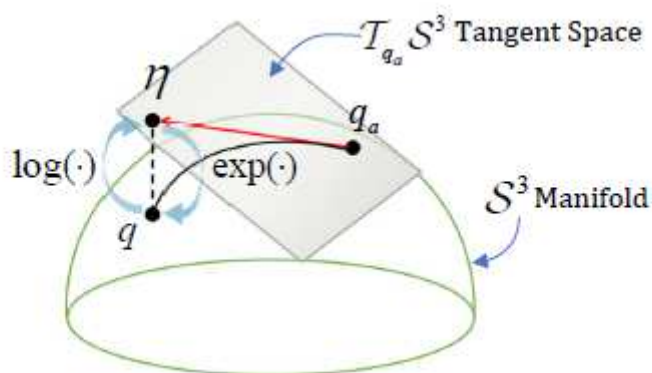


Figure 3

Graphic representation of the relationship between q and η.

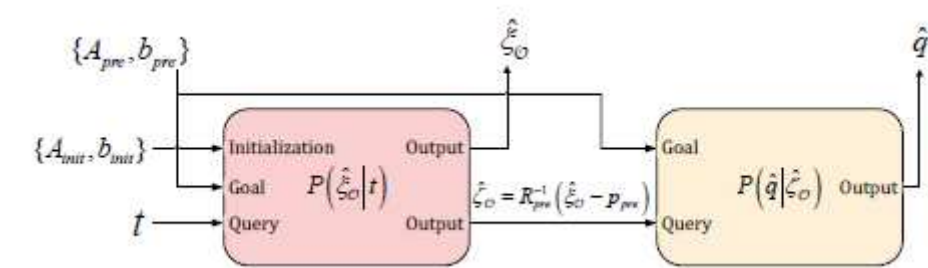


Figure 4

Structure of the approaching policy.

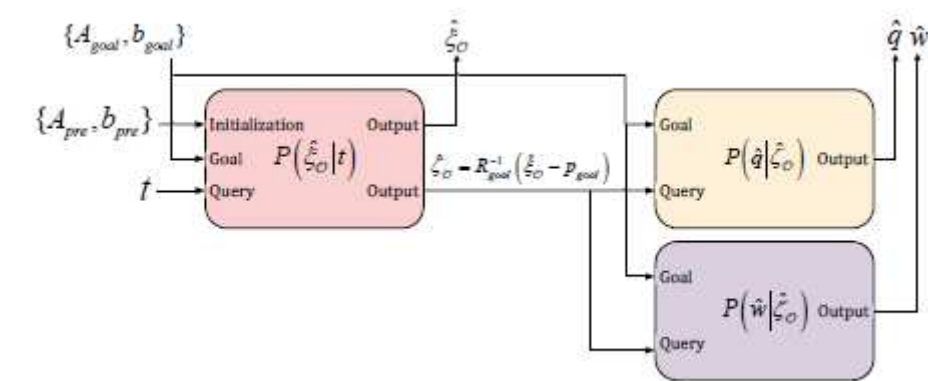


Figure 5

Structure of the precise assembly policy.

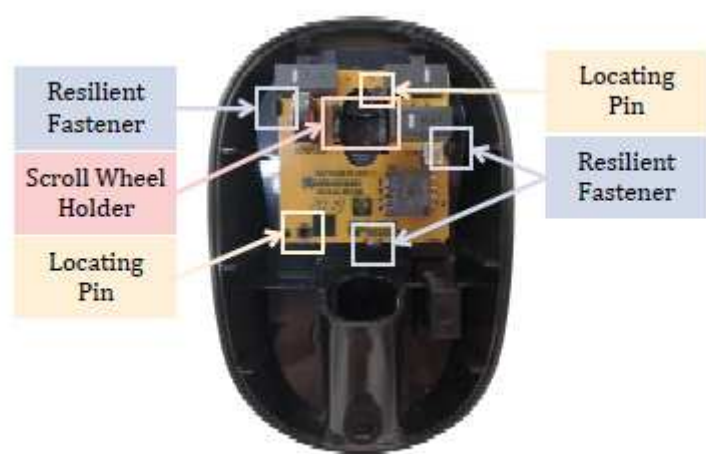


Figure 6

The PCB and bottom case of a mouse.

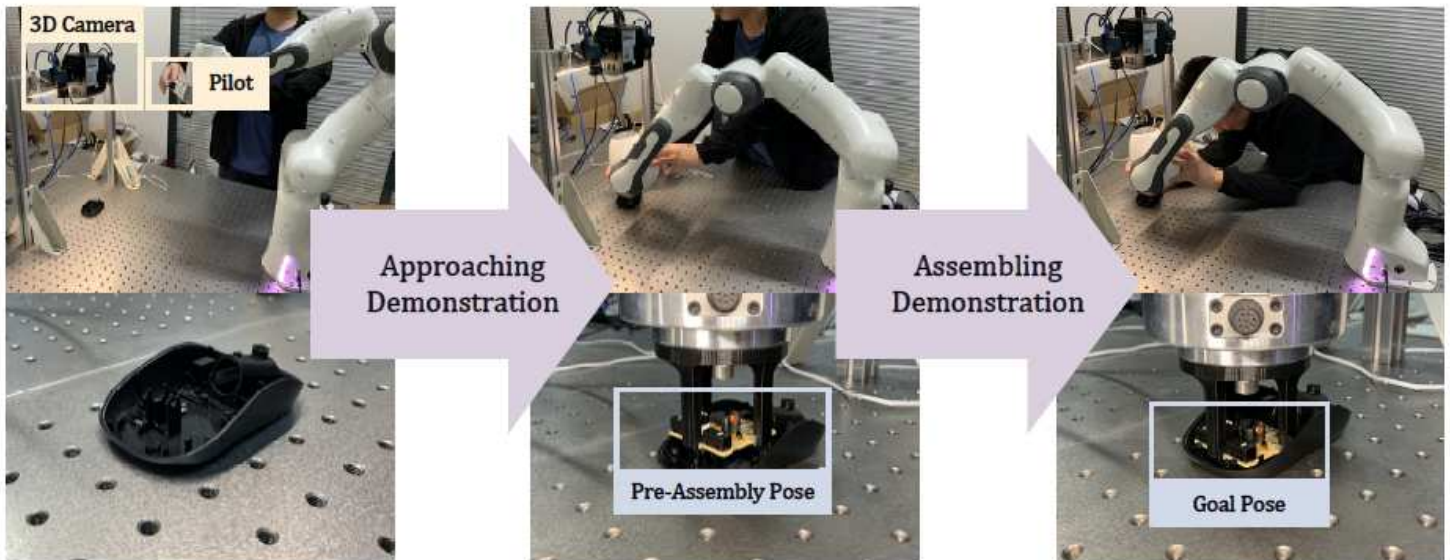


Figure 7

Kinesthetic guiding demonstration process.

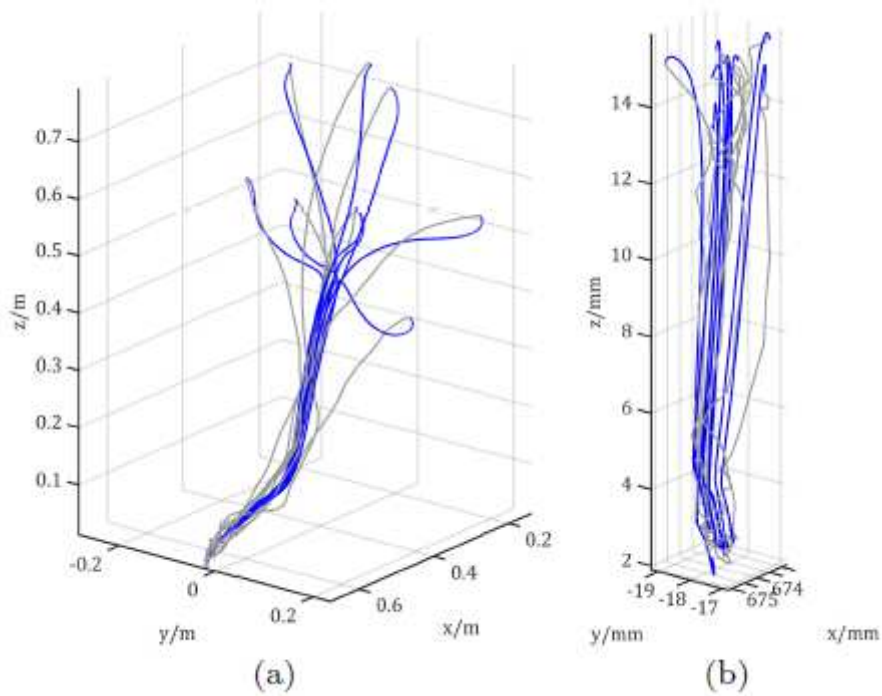


Figure 8

Demonstrated (grey) and retrieved (blue) position data in approaching phase (a) and assembling phase (b).

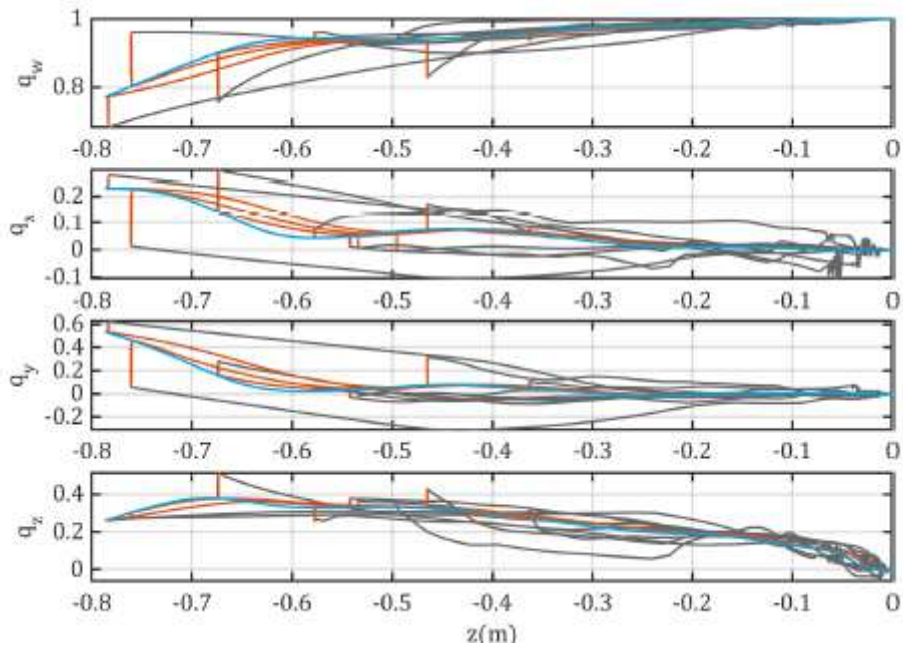


Figure 9

Demonstrated (grey), retrieved (blue) and recorded (orange) assembly orientation trajectories in approaching phase.

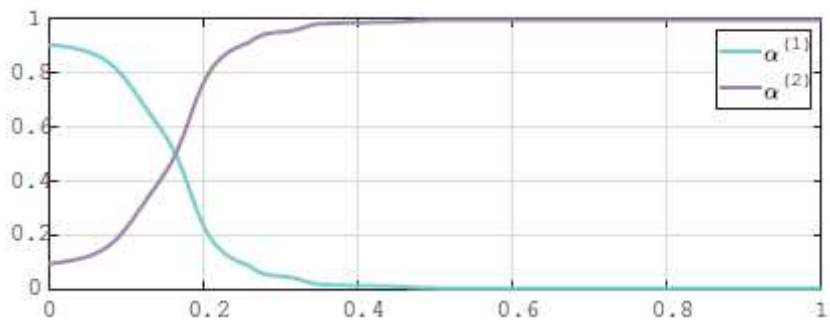


Figure 10

Weight of each frame during position data retrieving.

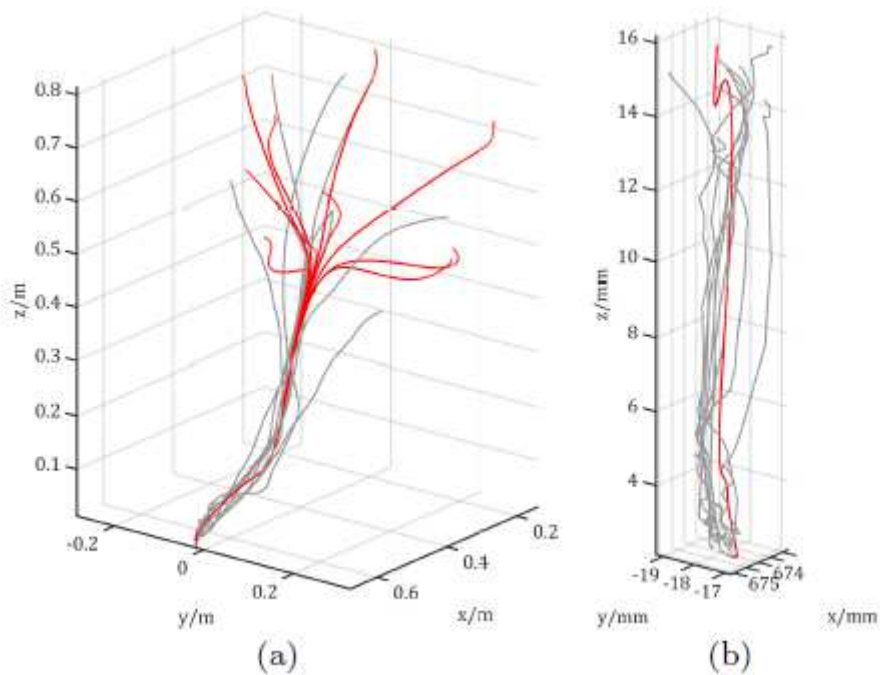


Figure 11

Demonstrated (grey) and generalized (red) position data in approaching phase (a) and assembling phase (b).

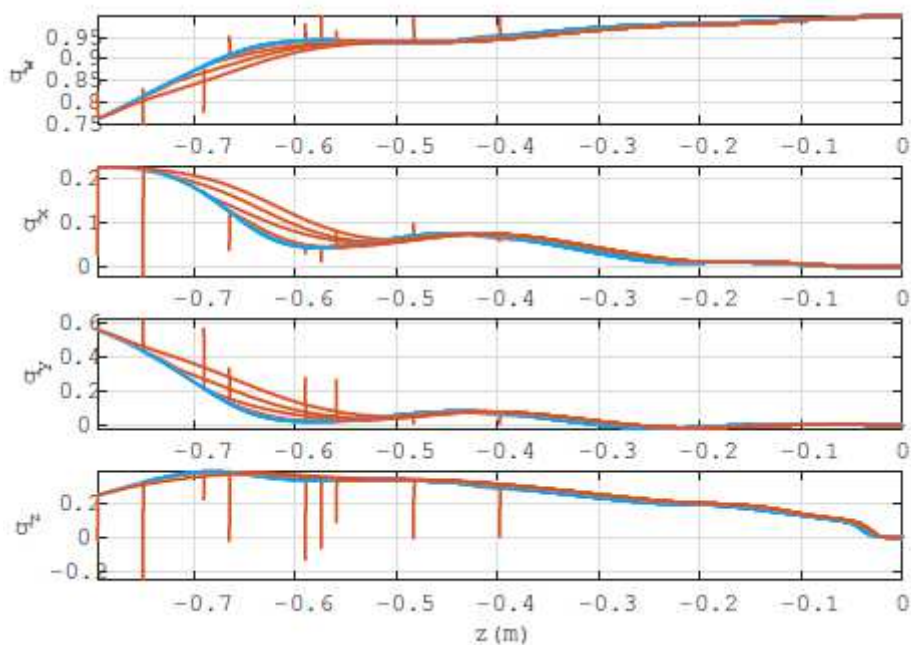


Figure 12

Generalized (red) and recorded (orange) assembly orientation trajectories in approaching phase.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [supplementedvideo.mp4](#)