Generalization of orientation trajectories and force-torque profiles for learning human assembly skill

Boyang Ti[a], Yongsheng Gao[a], Ming Shi[a], Jie Zhao[a]

[a] *State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China;*

Abstract

Constructing system models in industrial tasks is difficult, especially when multiple sensing information is involved. The peg-in-hole task is typically complex in industrial manufacturing. It is hard to model different assembly states precisely. Learning from demonstration can simplify the modelling process based on human demonstration. In this paper, we transfer a new kind of assembly skill to the robot. To intuitively represent the orientation of the assembly, we propose a novel expression named assembly-angle based on direction cosine and rotation matrix, which offers a good differential performance and avoids the singularity problem. Meanwhile, this approach can be encoded by the dynamic constraint dynamic movement primitives (DC-DMPs) method, which is proposed to consider the constraints among the assembly-angle during movement generalization. The RBF network is employed to generalize force-torque information to improve the accuracy of force-torque profiles prediction. We utilize the vision system and apply admittance control to locate the hole precisely. This proposed assembly skill learning framework can complete the task from random deviation of the pose, which makes it suitable for realistic assembly scenarios. To validate our method's generalization ability, pegs of two kinds of materials were designed, and the deviations of the initial pose were set from small to large. The success rate was 93.75% for high stiffness assembly and 86.25% for low stiffness assembly. The result validated the effectiveness of our method in the peg-in-hole task.

Keywords: Learning from demonstration, Robot skill learning, Peg-in-hole assembly, Movement generalization

1. Introduction

Assembly tasks are one of the prime tasks in industrial manufacturing. Robots are used extensively in industrial assembly scenarios with high efficiency and intelligence requirements. Robotic automated assembly is complex because assembly skills are a typical human behaviour that includes perception and reasoning in an unstructured environment. The traditional strategy primarily creates a specific policy for a particular state by analysing the characteristics between different states, which is not humanoid and is time-consuming to construct complicated mathematics models with multiple constraint equations and transition conditions for the varying situation. Another popular paradigm is learning through trial and error, which attempts to find a suitable assembly action model. Although the robot can find a solution to meet the task's requirements, its behaviour may be significantly different from that of humans. Currently, imitation learning develops a novel way for robots to learn from demonstration, a natural and intuitive way to demonstrate the desired behaviour for learners to emulate. Robots can generalize learned skills in an unseen assembly scenario. Nevertheless, current research is still unable to find the regularity of assembly skills, which leads to a failure to provide robots with better imitation templates. Therefore, analysis of the key steps of the assembly becomes necessary.

The peg-in-hole task can be separated into two subprocesses: the search and insertion phases. In the case of the search phase, most research has applied conventional techniques such as Archimedes spiral search, or grid search [1, 2], and random search [3], and recently advanced techniques such as convex optimization [4]. The prominent feature of the above strategy is a blind search without vision system embedding. However, such techniques traverse very long distances or take a great deal of time in the search stage, similar to humans searching for unknown targets with their eyes closed. These strategies are only applicable if the search space is broad enough, but they do not work if the hole thickness is small, leaving a tiny space for searching. Meanwhile, these strategies are different from human behaviour.

Insertion is a complicated manipulation involving judging contact states and escaping from the jamming state. A conventional assembly scheme of a single peg-in-hole [5, 6] focuses on finding the correlation between geometric and mechanical models of the insertion process, including jamming state analysis and transfer function construction between position and force. [7] proposed a discrete framework of insertion where each subprocess is regarded as a discrete action. Meanwhile, the thresholds of forces and torques at the connection of each action are set manually. To realize the transition between each action, [1] also discretized the insertion, and an inexpensive method was proposed for peg-in-hole assembly without force feedback or passive compliance mechanisms. [8] decomposed this process into more discrete actions and constructed a force and position model separately. [9] mainly studied the two-peg-in-hole task, which is essentially similar to the single peg task, but the contact state transitions became complicated. The essence of the above assembly scheme is to make the whole insertion process discrete by setting the characteristic force or position threshold and then realize compliance manipulation by force control.

Another insertion strategy uses reinforcement learning with the neural network, which manipulates in a learning and exploration manner. This method needs considerable trial and error to determine the optimal parameters of the strategy and a large amount of supervised data to train the network. [10] proposed a human-robot interaction collaborative assembly framework with the reinforcement learning method to optimize the insection sequence, which increases the efficiency of the assembly task with low assembly precision. [11] exploited multiple modal fusion networks to extract the features among vision, force and position information, and then they were transferred to a control policy network. However, this could not be applied to high precision assembly tasks. Additionally, to eliminate a cumbersome discrete action framework, [12] constructed an actor-critic network introducing stiffness in the task space, which achieved adaptive stiffness compliance control. [13] utilized deep reinforcement learning to complete a shell workpiece with multiple pegs by adapting only the end-effector orientation. During this process, it compacted the insertion work without considering the relation between position and orientation. [14] explored the optimal parameters through reinforcement learning and utilized an actor-critic network with supervised data to narrow the trial space. Without the mathematical analysis of complex state transitions, the cost was excessive trials and data. Fuzzy Q-learning was employed to improve the adaptive ability of an impedance controller, but its tolerance of the assembly hole was 1 mm, which cannot meet most of the industrial requirements [15]. [16] used the neural-network-based movement primitive (NNMP) method to generate the motion and the reinforcement learning to optimize the assembly policy. Although the assembly precision can be improved, learning by imitation of demonstrations and iterative learning of real robots to obtain the best assembly strategy is complex and cumbersome, which increases time and equipment maintenance costs.

A novel way to perform peg-in-hole tasks, based on LfD [17], utilizes small microdata learning [18] to overcome the big data method and complex analysis and programming. DMPs [19] is a typical method that has been developed from different studies in small data learning. [20, 21] introduced force information to the DMPs framework, and the parameters in the model trained by demonstration can be adjusted online according to the force feedback to complete the grasping task. Inspired by the concept of admittance control, [22, 23, 24] used the force regressed by LWR as an external factor to adapt the generalized trajectory by DMPs, using open-loop iterative learning control to optimize the strategy. However, these methods limit the self-adaptability of robots during assembly. The statistical model is also a small data learning common tool, and [25] exploited GMM and GMR to model feedback force and corrected angular velocity sampled from the human coordinative assembly. [26] used the TP-GMM method to generate the assembly motion and employed the Petri nets to coordinate independent nodes representing different assembly stages. Due to ignoring the reproduction of the orientation, it is constrained to the rough assembly such as the wooden toy car wheel assembly. The intelligent robotic precision assembly system proposed in [27] could rapidly teach robots assembly skills, but the generalization capability is limited to the demonstration due to the limitations of the GMR. Thus the displacement of the initial pose cannot be too far out of the demonstration. In addition to force, the stiffness parameter was introduced by regression methods GMR [28], and SVR [29], which could imitate and generalize the compliance of demonstration. [30] obtained the dynamic relationship between the force components and the pose information using GMM and divided the assembly process into five steps, which increased the complexity of the task.

The orientation representation of the peg for generalization is also an important factor. There are many methods used to represent a pegs orientation such as Euler angles [7], quaternions [22, 23, 24] and rotation matrices [1, 29]. The Euler angles form a chart of all SO(3), the special orthogonal group of rotations in 3D space. However, this is smooth except for singularity. [31] specified the orientation trajectory $R(t) \in \mathbb{R}^{3\times3}$ and introduced two equations to integrate the coefficients of $R(t)$ independently applied for DMPs. However, it is complicated to represent the assembly orientation using a rotation matrix with nine coefficients. Therefore, the quaternion is used, which concisely encodes the orientation using four coefficients and can also avoid the problem of singularity. The quaternion is also differentiable and integrable and has good interpolation property; however, intuitively, it is difficult to understand the orientation represented by quaternions in industrial applications. Furthermore, if expressed as quaternions, the assembly pose is not unique since $-q$ and q can represent the same pose. The rotation matrix needs nine components to represent an orientation, which consumes considerable memory and time to solve the pose. To simplify the representation of orientation, [9, 12, 30] projected the angle onto a plane and controlled only one dimension orientation; however, these methods ignored the relationship between different dimensions of the orientation.

The imitation of the force-torque profile is another important factor. One of the simplest ways is to take the demonstration force information as a reference to imitate [23, 24, 32]. [22] initially considered the nearest force profile as the reference, which needs ample demonstration to cover all the situations of the task. Even if physical derivations to construct mathematical models between force and position can be avoided, it is costly and cumbersome for humans to demonstrate robots learning assembly skills. To improve the prediction ability of interactive force information, they utilized the LWR method and followed the least squares optimization problem for each dimension of the force-torque profile, which endows the robot with the prediction of force. However, this cannot guarantee prediction accuracy.

Motivated by the earlier discussions and considerations, the main contributions of this paper are as follows: In terms of locating holes with narrow widths, we adopt a vision system to fuzzily obtain the pose of the target hole and employ admittance control to compensate for the vision error to reach a three-point contact state. A novel method is proposed to encode assembly orientation called assembly-angle, which reduces the complexity of the model and intuitively represents the orientation. To consider the coupling of the assembly-angle into the generalization model, a dynamic constrained dynamic movement primitives (DC-DMPs) method is proposed to generalize the orientation trajectory to adapt to the varying task requirements. For improving the accuracy of imitation, the RBF network is integrated into the online generalization of the assembly force-torque profile.

The rest of this paper is organized as follows: Section 2 describes the assembly strategy and introduces the assembly-angle; Section 3 introduces the representation and generalization of force-torque profiles; Section 4 describes the construction of the experimental platform and procedure; Section 5 discusses the limitation of our method and finally, conclusions are drawn in Section 6.

2. A Novel Peg-in-Hole Strategy

2.1. Hole Searching

There are multiple ways to locate the hole pose before insertion. For the narrow width of the hole locating problem, we utilize the vision system to detect the hole position. However, its accuracy cannot meet our requirement. Thus, we propose a strategy to improve the accuracy. The first step is to bring the peg into three-point contact with the hole, as shown in the diagram above. We formulate this as a geometric problem to accurately describe the hole searching process.

As Fig. 1(a) shows, the edges of the peg and the hole are ideally tangent at the same point. The tilt angle of the axis c is set randomly according to the task requirements. The distance between the peg head and hole axis is k, and the height to the hole plane is z. d is the angle between the projection of the peg axis onto the hole plane and the y-axis of the hole. The peg needs to move along the axis until it touches the hole, where L is the displacement distance. To eliminate the effects of errors caused by the vision system, radial admittance control is used to endow the peg with compliant behaviour along this direction. Since the tolerance between peg and hole is much smaller than the peg diameter, in the following derivation, we assume that the diameter D of the peg and the hole are the same. We then obtain the position of the initial searching state, as Eq.(1) and Eq.(2) show. Finally, we can obtain the three-point contact situation following the movement mentioned above.

$$k = \frac{D}{2} - \frac{D\cos(c)}{2} \qquad z = \frac{D}{2}\sin(c) \tag{1}$$

$$\begin{gathered} x = k\sin d \qquad y = k\cos d \\ \tan d = \frac{\cos a}{\cos b} \end{gathered} \tag{2}$$

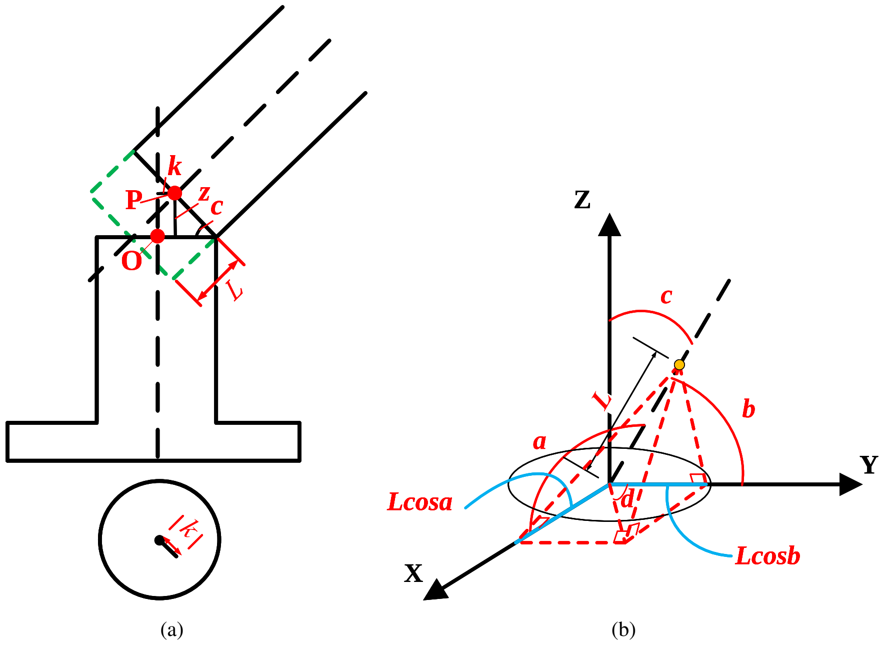


Fig. 1. Geometric illustration of the search initial state. (a) Illustration of the position part, where O is the centre of the hole and P is the centre of the peg. The horizontal distance between these two centres is k, and the vertical height is z. c is the angle between two axes. (b) Illustration of the assembly-angle part, where a, b, c is the assembly-angle and L is the distance to touch the edge of the hole. $Lcosa$ and $Lcosb$ are the projection lengths on the x and y axes

2.2. A Typical Human Assembly Technique

In this part, we illustrate a kind of human assembly technique that is often applied to peg-in-hole tasks starting from a three-point contact state.

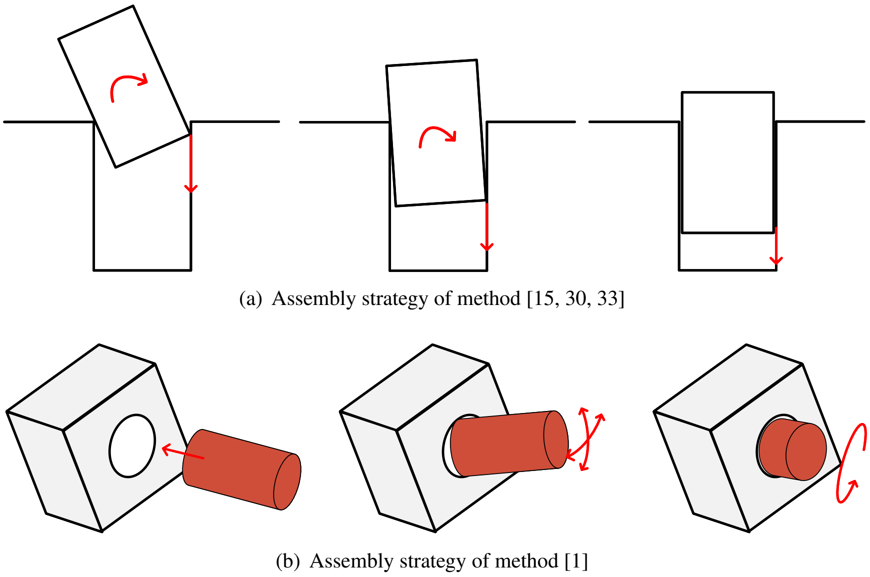
As Fig. 2(a) and Fig. 2(b) show, [15, 30, 33] and [1] these two typical peg-in-hole strategies divide the whole process into several units to simplify this skill. However, state judgment is needed to choose suitable action units and switch among them, which cannot realize a truly smooth insertion.

By observing several cases of human assembly behaviour, we find that the essential feature of human assembly is to lean the peg against the edge of the hole, with the other side leaving a maximum space, equivalent to the tolerance between pegs and holes for insertion. Therefore, we capture this feature to propose a novel assembly strategy, where we hold the peg head against the hole edge and rotate it while adjusting the insertion depth until the angle between the peg and hole approximately turns to 0. We showcase this strategy in Fig. 3. Fig. 3(a) shows the initial state of the peg, using the hole searching method introduced below to achieve a three-point contact state. Fig. 3(b) shows the key step of the assembly skill in this paper, with the 2D plane illustration shown in Fig. 3(d). Fig. 3(c) shows the vertical insertion stage.

2.3. Orientation Represented by assembly-angle

Quaternion is an accurate and general way to represent the orientation and is widely used in many applications. It has excellent mathematic performance and is easier to interpolate between quaternions. In our opinion, it requires more mathematics knowledge and is not intuitive, especially for the learning from demonstration. It is not designed for non-expert people to manipulate the peg-in-hole task. In this section, we introduce a new expression for non-expert people to visualize and understand the orientation of the assembly movement.

In the case of a single-axis assembly task, the angle between the axis of the peg and hole is a critical feature to describe the orientation in the assembly task. However, the orientation of the peg cannot be uniquely determined considering only one angle, and it is necessary to add two additional angles separately, namely, the axis of the peg between the other two axes of the coordinate. In this way, the original representation of orientation, such as quaternion,



(a) Assembly strategy of method [15, 30, 33]

(b) Assembly strategy of method [1]

Fig. 2. Typical assembly strategies of previous research

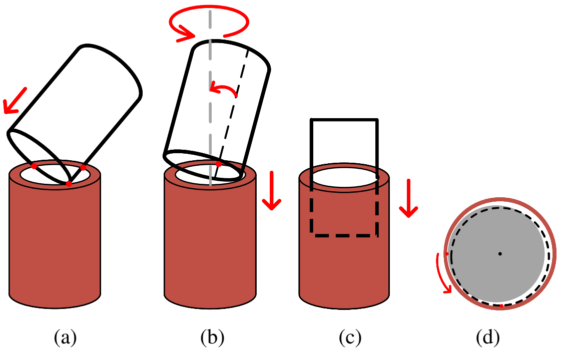


Fig. 3. State of contact of cylinder peg and hole under assembly force. (a) Three-point contact state. (b) Lean on edge while rotating and inserting. (c) Vertical insertion. (d) 2D plane illustration of rotation along the hole edge.

rotation matrix or Euler angles, can be converted into another form, which we define as the assembly-angle. This idea is inspired by the direction cosine [34]. In analytic geometry, the direction cosines of a vector are the cosines of the angles between the vector and the three positive coordinate axes. Equivalently, they are the contributions of each component of the basis to a unit vector in that direction. First, we take the Euler angles as an example to derive its relation with the assembly-angle as Eq.(3) shows. Therefore, we record orientation using the form of Euler angles.

$$ {}_{B}^{A}R_{X'Y'Z'} = R_X(\alpha)R_Y(\beta)R_Z(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\alpha & -s\alpha \\ 0 & s\alpha & c\alpha \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3} $$

Corresponding to the peg coordinate system, we use the XYZ rotational order to represent the Euler angles, where $c\{\alpha,\beta,\gamma\} = \cos\{\alpha,\beta,\gamma\}$, $s\{\alpha,\beta,\gamma\} = \sin\{\alpha,\beta,\gamma\}$. Then we merge them into Eq.(3) and obtain

$$ {}_{B}^{A}R_{X'Y'Z'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta \\ s\alpha s\beta c\gamma + c\alpha s\gamma & c\alpha c\gamma - s\alpha s\beta s\gamma & -s\alpha c\beta \\ s\alpha s\gamma - c\alpha c\gamma s\beta & c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta \end{bmatrix} \tag{4} $$

Here, we convert the rotation matrix to the assembly-angle. We take two points ^{B}P=(0, 0, 1) and ^{B}O=(0, 0, 0) on the peg, and then we transfer these two points to the hole coordinate.

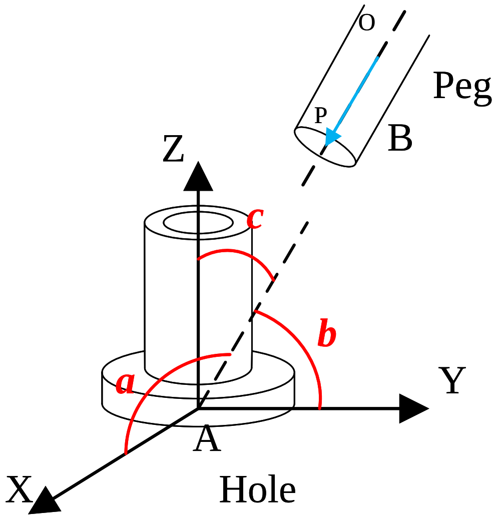


Fig. 4. Illustration of the Assembly-Angle. a, b, c are defined as the angles between the peg axis and these three positive axes, respectively

$$ {}^{A}P = {}_{B}^{A}T\,{}^{B}P \qquad {}^{A}O = {}_{B}^{A}T\,{}^{B}O \tag{5} $$

$$ {}_{B}^{A}T = \begin{bmatrix} c\beta c\gamma & -c\beta s\gamma & s\beta & P_x \\ s\alpha s\beta c\gamma + c\alpha s\gamma & c\alpha c\gamma - s\alpha s\beta s\gamma & -s\alpha c\beta & P_y \\ s\alpha s\gamma - c\alpha c\gamma s\beta & c\alpha s\beta s\gamma + s\alpha c\gamma & c\alpha c\beta & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{6} $$

where P_x, P_y and P_z are the offsets of the peg coordinate system relative to the hole. The vector OP represented in the hole coordinate system is

$$ {}^{A}P - {}^{A}O = \begin{bmatrix} s\beta \\ -s\alpha c\beta \\ c\alpha c\beta \end{bmatrix} \tag{7} $$

Finally, we obtain the assembly-angle as shown in Eq.(8).

$$a = \arccos(s\beta) \qquad b= \arccos(-s\alpha c\beta) \qquad c = \arccos(c\alpha c\beta) \tag{8}$$

Quaternions can also be transformed into the assembly-angle depending on the form we record in the demonstrations. In particular, the assembly-angle can be taken into the DMPs framework for learning and generalizing. Since the assembly-angle cannot be passed to the robot directly, we need to derive backwards to obtain the corresponding Euler angles for the controller when executing the movement.

$$\alpha = \arctan 2(-\cos b, \cos c) \qquad \beta= \arctan 2(-\sin \alpha \cos a, \cos b) \qquad \gamma = 0 \tag{9}$$

As the rotation around the z axis of peg does not influence the success of the round peg assembly task, this rotation is simplified in this paper by setting the value of γ to 0. The assembly-angle obtained by transformation simplifies representing the orientation, and we can apply less effort to the solution of orientation.

3. Record and Generalize the Assembly Information

3.1. Record the Assembly Information

Human skills can be transferred to robots in several ways. In this paper, we refer to [35] to design a kind of demonstration tool consisting of handling with NDI markers, peg and ATI force/torque sensor as Fig. 5 shows. The six dimensional Cartesian space movements are recorded along with the force and torque acting on the peg. The force and torque are captured in the tool coordinate, and movement is in the hole coordinate. The recorded data can be directly passed to the robot, which is more convenient than a human operator physically guiding the robot. In the case of precise assembly, it is difficult for humans to drag the robot to execute precise manipulation, and humans are unable to demonstrate movement smoothly while dragging the heavy robot. Through our method, the robot can obtain the human optimal assembly skill, and it is also efficient and simply for human operators to handle this demonstration tool to show the assembly skill rather than guiding the robot kinesthetically.

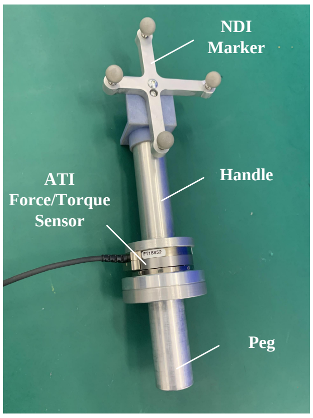


Fig. 5. Demonstration tool consists of peg, handle, force/torque sensor and marker

To learn the assembly skill from the demonstration, the following learning data should be recorded.

$S_d=\left\{\mathbf{x}_{i,j}, \boldsymbol{\sigma}_{i,j}, \dot{\mathbf{x}}_{i,j}, \dot{\boldsymbol{\sigma}}_{i,j}, \ddot{\mathbf{x}}_{i,j}, \ddot{\boldsymbol{\sigma}}_{i,j}\right\}_{j=1,i=1}^{T_i,NumDemo}$ where $\mathbf{x}_{i,j}, \boldsymbol{\sigma}_{i,j}$ are the measured positions and orientations (represented as the assembly-angle). $\dot{\mathbf{x}}_{i,j}, \dot{\boldsymbol{\sigma}}_{i,j}, \ddot{\mathbf{x}}_{i,j}, \ddot{\boldsymbol{\sigma}}_{i,j}$ are the corresponding linear and angle velocities and accelerations, estimated by the numerical differential of the above position and orientation. In addition, the force and torque generated from peg are $\Phi_d = \left\{F_{i,j}, M_{i,j}\right\}_{i=1,j=1}^{T_i,NumDemo}$

All of these data are recorded at the same time $t_{i,j}$, $j = 1...T_i$, where T_i represents the duration of the *ith* demonstration, $i = 1...NumDemo$, and *NumDemo* is the number of demonstrations. The demonstrations are started from different initial angle deviations. Note that the distribution and the number of demonstration trajectories influence the success of generalization. Therefore, we arrange the initial state generally covering the typical human assembly pose to obtain the best training data.

3.2. Generalization of Assembly Skill

3.2.1. Generalization of Position and Orientation

As this is important for the understanding of the paper, we introduce the DMPs employed in the learning and generalization part as Eq.(10)-Eq.(14). The method of DMPs consists of the following free parameters: desired goal state of the trajectory $\mathbf{g^p}$ and the weights $\mathbf{w_i^p} \in \mathbb{R}^3$, i=1 $\cdots N$ of the forcing term $\mathbf{f^p}(s)$, which is defined as a linear combination of radial basic functions $\psi_i(s)$ and trajectory duration τ. The forcing term represents the characteristic of each imitated movement, and it is not affected by the changes in the task, such as the goal and initial state, which endows it with the generalization ability. Here, $\mathbf{v}$ denotes the velocity of the state $\mathbf{x}$. $\mathbf{x}$ will converge toward $\mathbf{g^p}$ monotonically with appropriate values of K and D. s is the phase term that drives the generalization process. Because of its advantage to learn motion characteristics from only one demonstration, we extract one closest to the current assembly state for learning the assembly movement.

$$\tau\dot{\mathbf{v}} = K(\mathbf{g^p} - \mathbf{x}) - D\mathbf{v} + (\mathbf{g^p} - \mathbf{x}_0)\mathbf{f^p}(s) \tag{10}$$

$$\tau\dot{\mathbf{x}} = \mathbf{v} \tag{11}$$

$$\tau\dot{s} = -\alpha s \tag{12}$$

$$\mathbf{f^p}(s) = \frac{\sum_i \psi_i(s) w_i^p}{\sum_i \psi_i(s)} s \tag{13}$$

$$\psi_i(s) = \exp\left(-h_i(x - c_i)^2\right) \tag{14}$$

The orientation is represented by the assembly-angle, where there are interdependencies between different dimensions. The constraint relationships between them are shown in Fig. 6. There are two constraints in the assembly-angle representation

$$\begin{aligned} &\textbf{Constraint1}: \quad && c + b \geq \frac{\pi}{2}, c - b \geq -\frac{\pi}{2} \\ &\textbf{Constraint2}: \quad && a = \pm\sqrt{1 - (\cos^2 b + \cos^2 c)} \end{aligned}$$

Constraint 1 ensures that the solution of this pose is feasible, and Constraint 2 shows that any one of the angles can be solved based on the other two angles. Since the above equation has two solutions, we determine the unique solution according to the current position. Therefore, the generalization of the 3D assembly-angle can be simplified to a 2D problem in DMPs; meanwhile, the range of one angle is limited by the other angle.

Then, to solve the constrained generalization situation, Constrained Dynamic Movement Primitives (CDMPs) [36] enables the generalization of trajectories within fixed upper and lower boundaries. However, this does not work in scenarios where the boundary varies with a particular variable. Therefore, in this paper, we propose the Dynamic Constraint Dynamic Movement Primitives (DC-DMPs), which can generalize movement within variable limitations. The core concept of our method is to translate the demonstration state into exogenous states and parameterize the exogenous state as a function instead so that DMPs can evolve motion within variable limits. The following Eq.(15)-Eq.(19) is the derivation of DC-DMPs, which is optimized based on CDMPs. We take the assembly-angle as an example where the demonstration trajectory of assembly-angle $\sigma \in \{a, b, c\}$ is parameterized by the DC-DMPs state ξ.

$$\sigma(\xi) = \sigma_\delta(t)\tanh(\xi) + \sigma_0(t) \tag{15}$$

Where $\sigma_0(t) = \frac{1}{2}(\sigma_{\max}(t) + \sigma_{\min}(t))$ defines the mean angle of the current state at t, $\sigma_\delta(t) = diag\left(\frac{1}{2}(\sigma_{\max}(t) - \sigma_{\min}(t))\right)$ defines half of the deviation between the upper and lower boundaries. ξ can be simply obtained by using the inverse transformation from Eq.(15). Then, we can construct the DC-DMPs model of ξ in the exogenous space.

$$\xi = \operatorname{arctanh}\left(\sigma_\delta^{-1}(t)\left(\sigma(\xi) - \sigma_0(t)\right)\right) \tag{16}$$

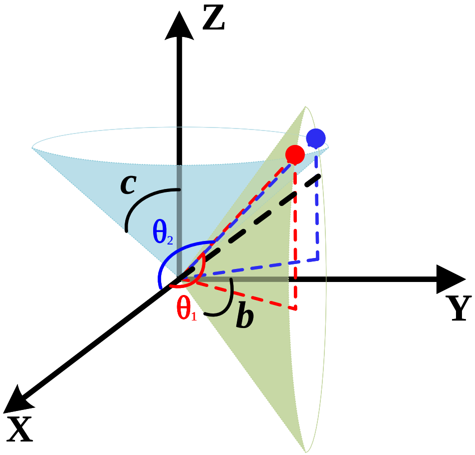


Fig. 6. Two constraints of the assembly-angle. The θ_1 and θ_2 circles represent two available solutions, where one is on the positive side and the other is on the negative side

The evolution rule of DC-DMPs follows the transformation system in DMPs. The parameters of this equation are similar to those mentioned above, we substitute ξ into Eq.(10), Eq.(11), Eq.(13), and we can obtain Eq.(17)-Eq.(19)

$$\tau\dot{\varsigma} = K(g^{\xi} - \xi) - D\varsigma + (g^{\xi} - \xi_0)f^{\xi}(s) \tag{17}$$

$$\tau\dot{\xi} = \varsigma \tag{18}$$

$$f^{\xi}(s) = \frac{\sum_i \psi_i(s) w_i^{\xi}}{\sum_i \psi_i(s)} s \tag{19}$$

Since the range of c in the assembly task is generally $[0, \pi/2)$, to simplify the process of generalization, we define c as the dimension of unconstrained generalization using DMPs. The evolution of c in DMPs is

$$\tau\dot{\eta}^{c} = K(g^{c} - c) - D\eta + (g^{c} - c_0)f^{c}(s) \tag{20}$$

$$\tau\dot{c} = \eta^{c} \tag{21}$$

On the other hand, DC-DMPs is applied to the generalization of the constrained angle b. Therefore, Constraint 1 of the assembly-angle can be converted to $\pi/2 - c \le b \le \pi/2 + c$, where the boundary of b is a function of c, while the c varies with the phase term s. Therefore, we substitute b and c into Eq.(15)-Eq.(19), and we can obtain Eq.(22)-Eq.(26). We can observe that it retains the forcing term expressed in the exogenous space to maintain the ability to generalize. By comparing Eq.(22) with Eq.(16), we find that the difference from CDMPs is that we consider the constraints between different dimensions in the calculation of exogenous state ξ. Therefore, the generalization can be limited in the dynamic constraint.

$$\xi^{b} = \operatorname{arctanh}\left(b_{\delta}^{-1}(c(s))\,(b - b_0(c(s)))\right) \tag{22}$$

$$b_{\delta}(c(s))\text{=diag}(c(s)),\ b_0(c(s)) = \frac{\pi}{2} \tag{23}$$

$$\tau\dot{\varsigma}^{b} = K(g^{\xi^{b}} - \xi^{b}) - D\varsigma^{b} + (g^{\xi^{b}} - \xi_0^{b})f^{\xi^{b}}(s) \tag{24}$$

$$\tau\dot{\xi}^{b} = \varsigma^{b} \tag{25}$$

$$f^{\xi^{b}}(s) = \frac{\sum_i \psi_i(s) w_i^{\xi^{b}}}{\sum_i \psi_i(s)} s \tag{26}$$

After reproduction of position and two angles of assembly orientation, we can uniquely identify the third angle a.

3.2.2. Representation of the Force-Torque Profiles

Force-torque representation in Cartesian space generally consists of 6 dimensions $\left\{F_x,\ F_y,\ F_z,\ T_x,\ T_y,\ T_z\right\}$; however, it is not the most concise form of expression in the assembly task. In this paper, we use a simple form, which is illustrated in Fig. 7. First, we ignore T_z because it has a minimal impact on the round peg-in-hole task. Then, we transfer F_x and F_y to the polar representation, where F_p and F_{p_a} denote the force perpendicular to the peg and the angle between plane constructed by the axis of the peg and the vertical z axis of the hole, respectively. The torque generated by F_p is denoted as T_p, of which the direction is determined by F_p.

$$F_p{=}F_x{+}F_y \qquad F_z = F_z \tag{27}$$

$$F_{p\_a}{=}\arccos(\frac{F_p \cdot Base}{\left|F_p\right| \cdot |Base|}) \tag{28}$$

$$T_p = \sqrt{{T_x}^2 + {T_y}^2} \qquad T_z \approx 0 \tag{29}$$

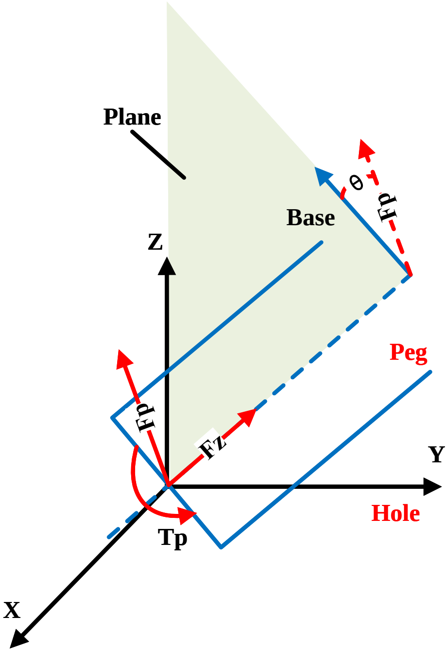


Fig. 7. Force-torque concise representation using $\{F_z,F_p,F_{p\_a},T_p\}$. F_z is the force along the axis of the peg, and F_p is the force perpendicular to the axis. F_{p_a} is the angle between F_p and the base plane, which is constructed by the axis of the peg and the vertical z axis of the hole. T_p is the torque applied on the peg.

3.2.3. Generalization of the Force-Torque Profiles

Based on the above form of representation, the approach of generalizing force-torque differs from position and orientation in that the derivatives of force and torque have no physical meaning. Therefore, DMPs is not applicable to such problems. [22, 24] proposed that the force-torque profiles are encoded as linear combinations of radial basis functions and introduced a kernel function on task parameters in the least-squares problem. In this paper, we employ an RBF network which is much easier to be designed and trained to construct a nonlinear relation between the force-torque and the phase term s in DMPs, as Fig. 8 and Fig. 9 show, and obtain the reference trajectory by weighting the neural network trained by the demonstration. The weights consist of evaluating its prediction performance R and similarity to a new task query D among these networks.

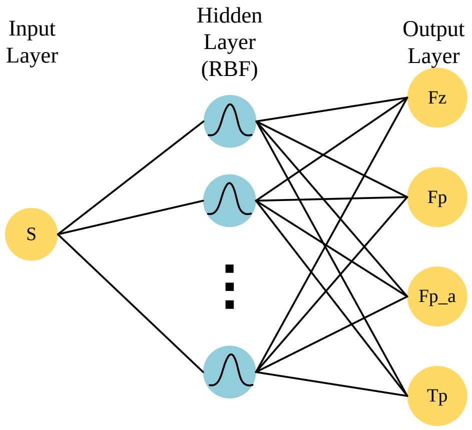


Fig. 8. A single network used to train the nonlinear relation between the force-torque information $\{F_z,F_p,F_{p\_a},T_p\}$ and phase term s. The input to the network is the phase term s, the output is the force-torque information, and the network has only one hidden layer.

$$r_j^N = \frac{\sum_{i=1}^{p} mse_i^N - mse_j^N}{\sum_{i=1}^{p} mse_i^N} \qquad R_j^N = \frac{r_j^N}{\sum_{n=1}^{N} r_j^n} \qquad j = 1, 2, \cdots, N \tag{30}$$

$$d(x_s, x_j) = \sqrt{\sum_{u=1}^{U} \left(\frac{x_{su} - x_{ju}}{s_u} \right)^2} \in \mathbb{R} \tag{31}$$

Since the distribution of each dimensional component of the task parameter is different, we introduce the Standardized Euclidean Distance to measure the distance between the new query task state and the demonstration to normalize each component as Eq.(31), where x_s represents the new query of the task, $x_j, j = 1 \ldots NumDemo$ is the demonstration task parameter, u is the feature index of the task parameter. s_u is the standard deviation of the individual components of the task parameters. We then obtain the distance metric as follows:

$$d_j^s = 1 - \frac{d(x_s, x_j)}{\sum_{j=1}^{NumDemo} d(x_s, x_j)} \qquad D_j^s = \frac{d_j^s}{\sum_{j=1}^{NumDemo} d_j^s} \tag{32}$$

We combine these two metrics to weigh the demonstration force-torque network to obtain the generalized network for the new task setting. Where μ is used to decide the importance of these two metrics. Finally, *metric* is defined as the weight of each network trained from the demonstration.

$$metric{=}\mu \times \mathbf{D} + (1 - \mu) \times \mathbf{R} \tag{33}$$

3.2.4. Online Trajectory Modification using Sensory Feedback

The main concept of this section refers to [20] the online modification of the desired trajectory generated by DMPs using sensor information from previous task executions. Real-time force-torque is sampled by the sensor mounted on the wrist to establish a direct relationship with movement acceleration. The desired acceleration can be modified to compensate for sensing errors. This external term is given by

$$\mathbf{F}_{ext} = \mathbf{K}_1 \mathbf{J}_{sensor}^{\mathbf{T}} \mathbf{K}_2 (\mathbf{F} - \mathbf{F}_{des}) \tag{34}$$

and it is optimized as

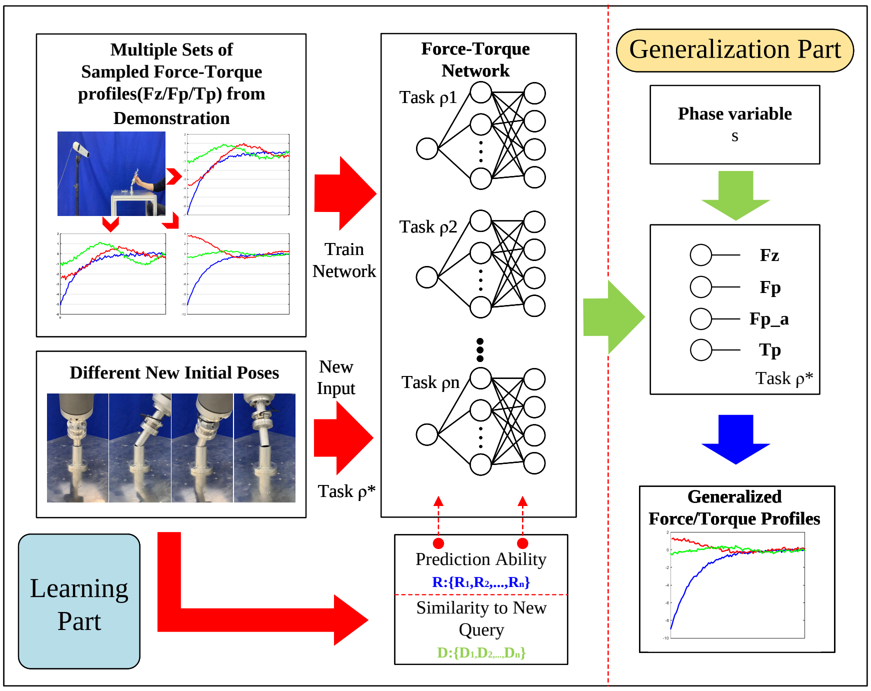


Fig. 9. Proposed framework to generalize the force-torque profiles

$$\tau\dot{\mathbf{v}} = K(\mathbf{g}^{\mathbf{p}} - \mathbf{x}) - D\mathbf{v} + (\mathbf{g}^{\mathbf{p}} - \mathbf{x}_0)\mathbf{f}^{\mathbf{p}}(s)+\mathbf{F}_{ext} \tag{35}$$

J_{sensor} is the Jacobian of the task controlled by the movement primitives with respect to sensors, F is the actual force sampled from sensors, and F_{des} is the corresponding desired force generated from the RBF network. Then, the previously generalized F_p and F_{p_a} need to be transformed back to a Cartesian expression. To modify the assembly-angle online, we find that F_{p_a} is close to zero; therefore, we take $T_p - T_{p_{des}}$ as the error term in the generalization of c.

$$\mathbf{T}_{ext} = \mathbf{K}_1\mathbf{J}^{\mathbf{T}}_{sensor}\mathbf{K}_2(\mathbf{T} - \mathbf{T}_{des}) \tag{36}$$

$$\tau\dot{\eta}^c = K(g^c - c) - D\eta^c + (g^c - c)f^c(s) + T_{ext} \tag{37}$$

To effectively minimize the force-torque error, we slow down the trajectory execution using DMPs slow-down feedback, referring to [24]. Specifically, for DMPs phase stopping, the original equation for phase Eq.(12) is replaced with Eq.(38)

$$\tau\dot{s} = -\frac{\alpha s}{1 + \alpha_e \varepsilon} \tag{38}$$

where ε is the tracking error estimated as the discrepancies between the desired force and actual force.

4. Experiment

To verify the effectiveness of the proposed method, an actual experiment is conducted with the UR10 arm. The ATI mini45 force/torque sensor is mounted on the end-effector to record the contact information between the peg and hole. The proposed algorithm and admittance controller are implemented on a PC with an Intel Core i7- 10750U CPU @ 2.60 GHz CPU, 16 GB RAM, and Ubuntu 18.04 robot operating system. The PC collects the force/torque

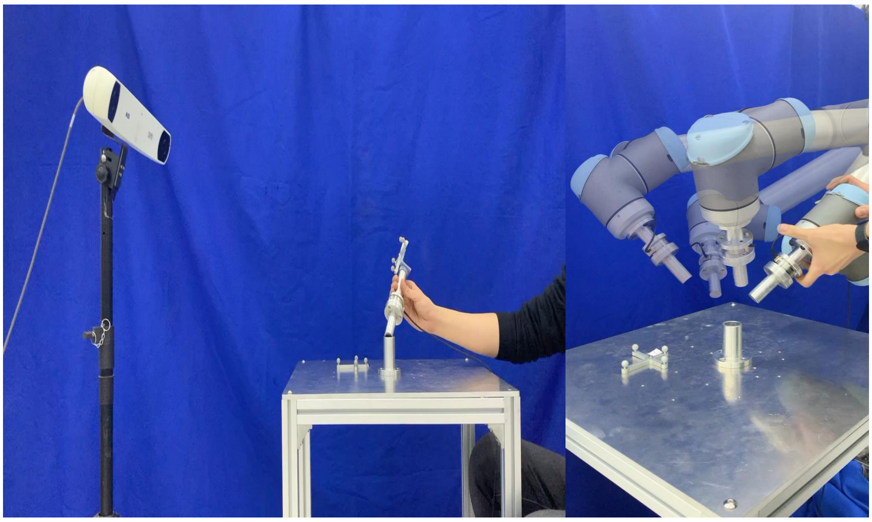


Fig. 10. Demonstration using the tool with NDI device and generalization setting from any pose

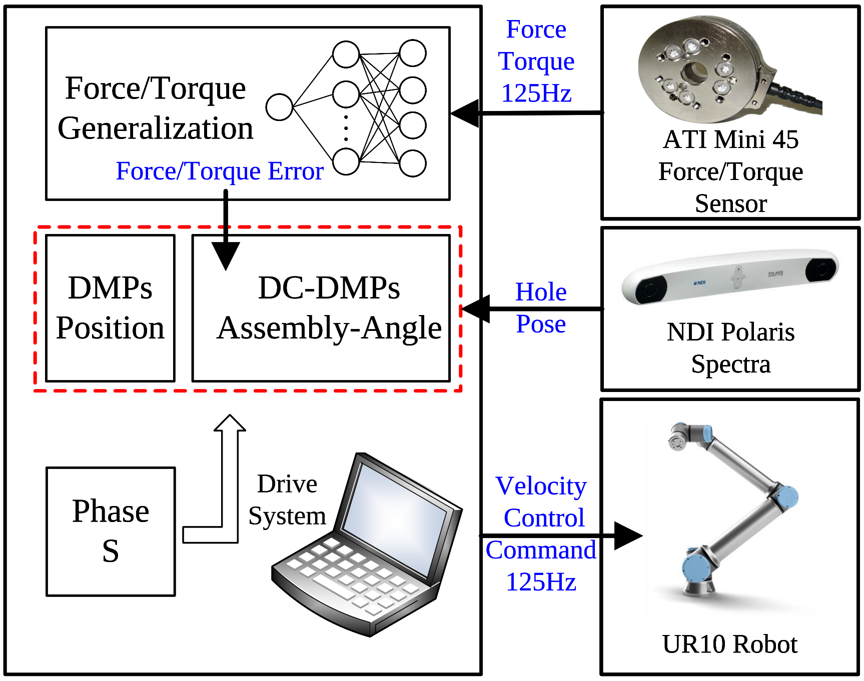
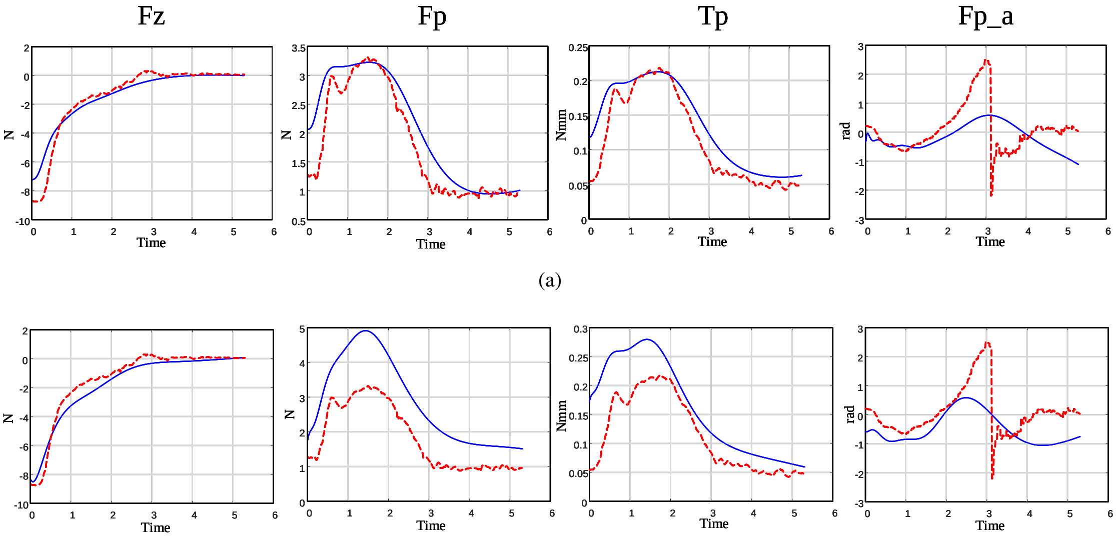


Fig. 11. Proposed generalization framework for assembly task

information from the force/torque sensor through the wireless device and obtains the current state of the robot through the robot driver package. Then, the RBF network generalizes the desired real-time contact information, while DMPs and DC-DMPs calculate the desired movement using these data. The control command is set at 125Hz. Our method can be applied to assembly tasks with different materials. We chose two materials for the peg: one made of aluminium, representing a high stiffness assembly, and the other one where we first 3D printed the base of the peg and then applied a silicone sleeve to the surface of the base, representing a low stiffness assembly. The tolerance of the peg and hole is 0.3 mm in both cases, where the peg and hole diameters are 25 mm and 25.3 mm, respectively. First, we utilize the proposed hole searching method to reach the three-point contact state. Then, we adopt the aforementioned skill generalization framework shown in Fig. 11 to perform the insertion stage. At the beginning of the experiment, the human drags the robot to a random pose. Then, the robot adopts the above search strategy to achieve the three-point contact state. With the phase variable s driven. Then, we choose the closest demonstration to learn the DMPs and DC-DMPs model and generalize the movement for the robot to perform the insertion action. Meanwhile, the network will provide the reference force/torque trajectory for the robot to modify its movement. We compare ours with two representative approaches [22, 24] whose assembly strategies are based on learning from demonstration. It should be noted that we generate the movement in the hole coordinate system and the generalized movements will not be affected by the pose of the hole.

4.1. Testing the Force/Torque Prediction Ability



(a)

(b)

Fig. 12. Prediction of F_z, F_p, T_p and F_{p_a}using (a) the proposed method and (b) the method in [22, 24]. The blue solid line is the predicted force/torque/angle trajectory and the red dash line is the actual force/torque/angle trajectory

First, we test the RBF network prediction ability by comparing it with the method in [22, 24]. We demonstrate the peg-in-hole movement 50 times, as Fig. 10 shows, where the initial states cover almost one quadrant of the hole coordinate system. We extract 40 groups of force/torque data for training, and the other 10 groups are randomly taken as the testing data. Compared to [22], they acquired 100 training trajectories for learning the force/torque prediction model, and their assembly tolerance was about 1mm. The number of units in the network increases until the output error reaches the specified error performance. The prediction results are shown in Fig. 12. According to the results, it can be seen that the prediction of F_z has the same good performance; however, for the prediction of F_p and T_p, our method has much higher accuracy, which will improve the success rate of the insertion movement. A point worth explaining is that there is a significant prediction error at the end of F_{p_a}. The reason is that the desired F_p is almost zero when the peg is entirely in the hole; however, it is easy to generate noise data because of the vibration of the peg in the hole; therefore, the effect of F_{p_a} during this stage is almost nonexistent. We showcase the prediction

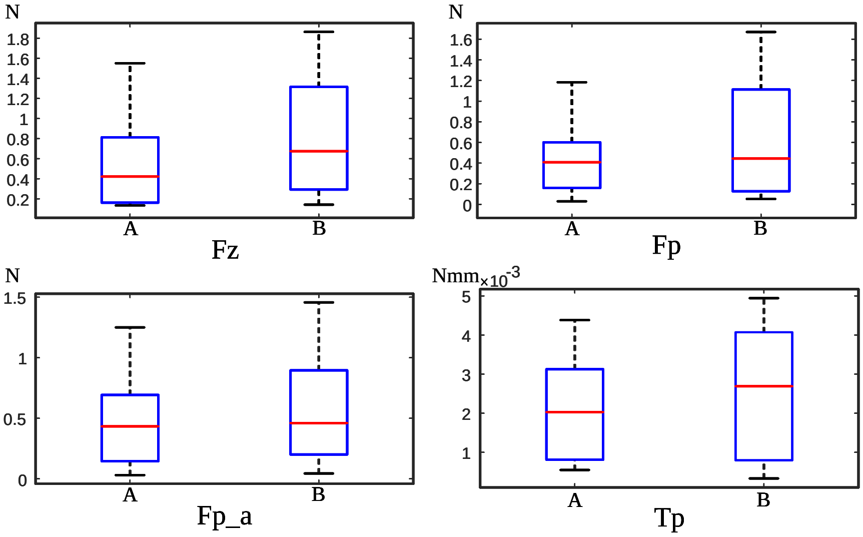


Fig. 13. Measurement errors of the proposed method (A) and the method in [22, 24] (B) expressed by Boxplot

accuracy in Fig. 13 with a box plot. Based on the above two figures, the median, maximum and minimum values of the prediction error are less than those of the comparison method. We can conclude that the prediction method used in this paper has higher accuracy than the method in [22, 24].

4.2. *High Stiffness Peg-in-Hole Task*

4.2.1. *Searching Stage*

To verify the efficiency and robustness of the strategy proposed in Section 2.1 for searching holes, the peg moves along the z axis in the tool coordinate system, while the $x - y$ plane position is adjusted by admittance control until it contacts the edge of the hole. We conduct 50 sets of test experiments, and the initial pose is set randomly. The experiment is successful if the peg can achieve the three-point contact state. Finally, the success rate can reach 88%. The experimental process and snapshots are shown in Fig. 14.

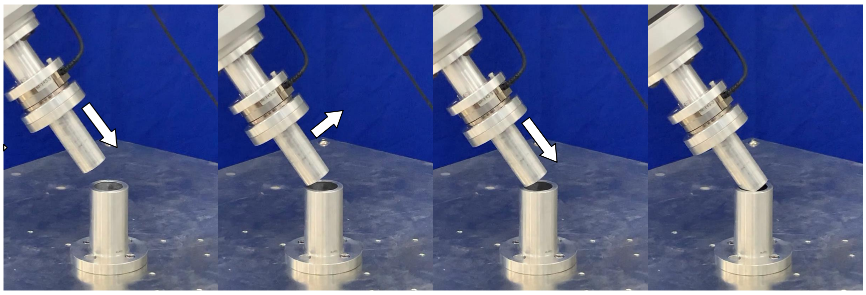
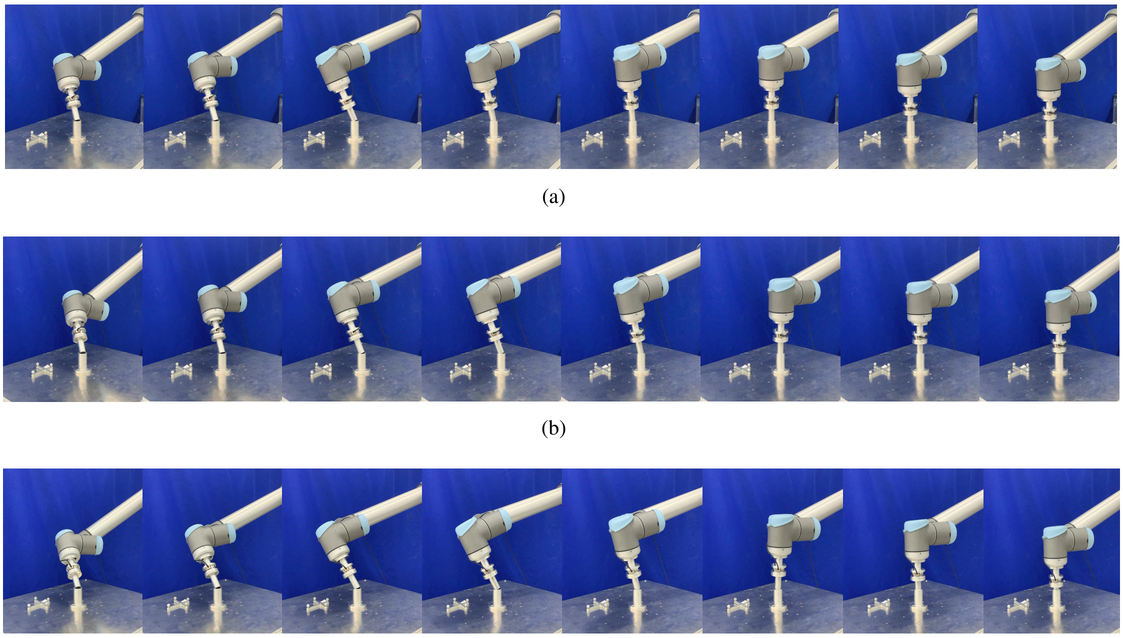


Fig. 14. Screenshots of the proposed hole searching method procedure, where white arrows represent the corrected direction of peg movement

4.2.2. *Inserting Stage*

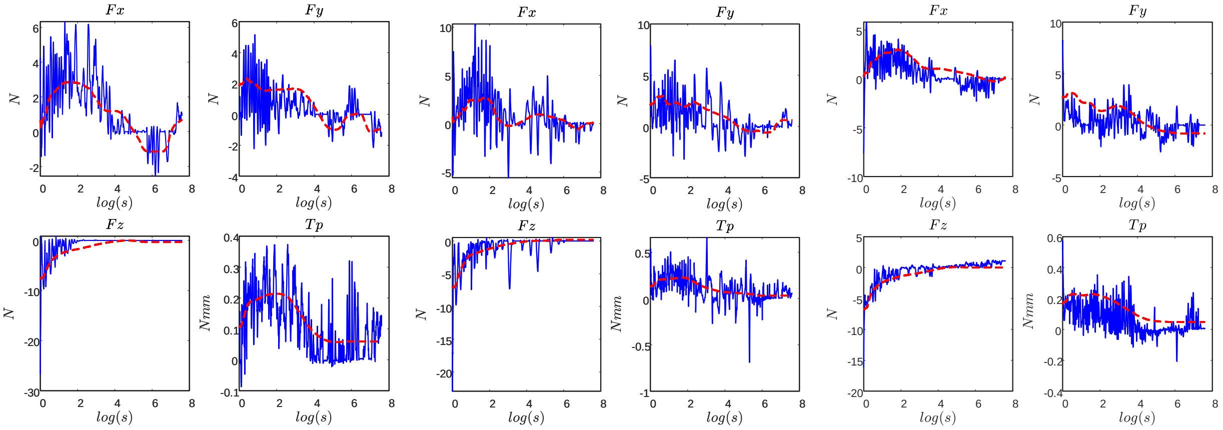
The robot starts with three levels of deviation between the peg and hole axis, from small 20° to medium 40° and then large 50° to verify the generalization ability of our method for variable deviations of the pose, as Fig. 15 shows. The robot rotates the peg along the inner edge of the hole while executing the insertion action, and the generalized trajectory is continuously adjusted online according to the current contact state until the peg reaches the desired hole depth. As we can see from the supplemental video, the performance of the rotational insertion movement is superior.


(a)

(b)

(c)

Fig. 15. Screenshots of the insertion process in the condition of angle deviation of 20°(a), 40°(b) and 50°(c) in the high stiffness assembly task

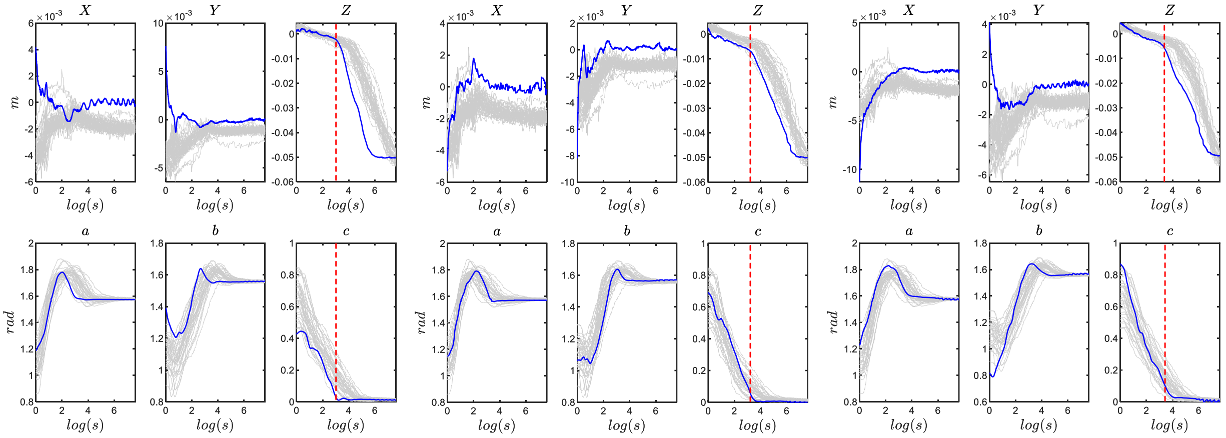


(a) 20° deviation

(b) 40° deviation

(c) 50° deviation

Fig. 16. Desired and sampled force-torque profiles of the end-effector. The red dotted line represents the desired force-torqe data, and the blue line represents the sampled force-torque data. (a) represents the data of 20° task. (b) represents the data of 40° task. (c) represents the data of 50° task



(a) 20° deviation (b) 40° deviation (c) 50° deviation

Fig. 17. Position and assembly-angle trajectory of the high stiffness assembly compared with demonstration

The contact force and torque data sampled from three kinds of initial angles are shown in Fig. 16, where the red dotted line is the reference force and torque generalized from the network and the blue line is the actual contact force and torque. The horizontal axis represents the log form of phase variable s. The trend of the actual interaction force is consistent with the reference trajectory, which further improves the accuracy of the correction of the movement. In the first stage, the peg is rotating along the hole edge while keeping the contact force vertical to the peg constant, so the magnitude of the force along the x and y axes is alternating. Meanwhile, the peg is gradually inserted into the hole so that the force along the peg gradually decreases until it is vertical to the hole plane.

In Fig. 17, we can see the position and orientation trajectory of the insertion movement, where the blue line is the generalization movement and the multiple grey lines are the demonstrations. The 20° and 50° initial cases are out of the demonstration region, which verifies the generalization ability of our method. The trends of the assembly-angle trajectory from different initial states are similar to demonstration, and the angle c eventually converges to 0. The area where the position changes gently in the z-axis is the stage of rotational insertion of the peg before the red dotted line, as shown in the Z and c figures in Fig. 17. As the assembly-angle c gradually decreases to 0, it approaches vertical insertion. We can also find that during vertical insertion, the assembly angles a and b are approximately 90°, so it can be proved that the peg is in the correct pose for vertical insertion.

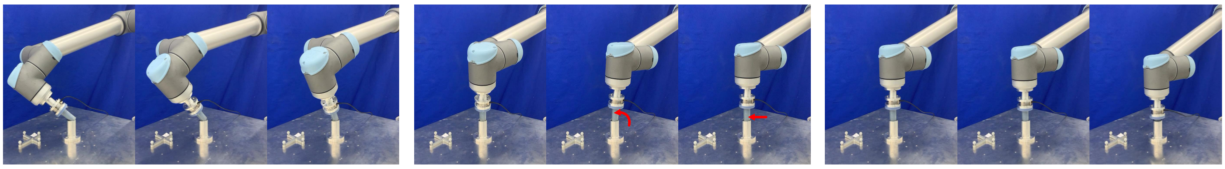
From the above analysis of the experimental results, the orientation representation proposed in this paper can allow an effective description of the assembly process. It can also be taken into the DC-DMPs framework to realize the constrained generalization of orientation. We conducted 80 sets of experiments to test the method proposed in different quadrants with different initial angles, and the success rate reaches 93.75%. We also compare the results with other typical methods used in the peg-in-assembly task. In Table 1, we can see the advantage of our approach.

Table 1: Comparison of different assembly strategies

	Any initial pose	**Big angle deviation**	**Simple orientation representation**
Method [22]	×	✓	×
Method [1, 24, 30]	✓	×	×
Proposed method	✓	✓	✓

4.3. *Low Stiffness Peg-in-Hole task*

We chose the soft peg assembly task to verify that our method can also be generalized to different materials. We made a plastic peg base using the 3D printing technique and place a piece of silicone over the top of the peg. The dif-



(a) Rotation Stage (b) Adjustment Stage (c) Insertion Stage

Fig. 18. Screenshots of the insertion process in the low stiffness assembly task. The whole process is divided into three stages: rotation, adjustment and insertion

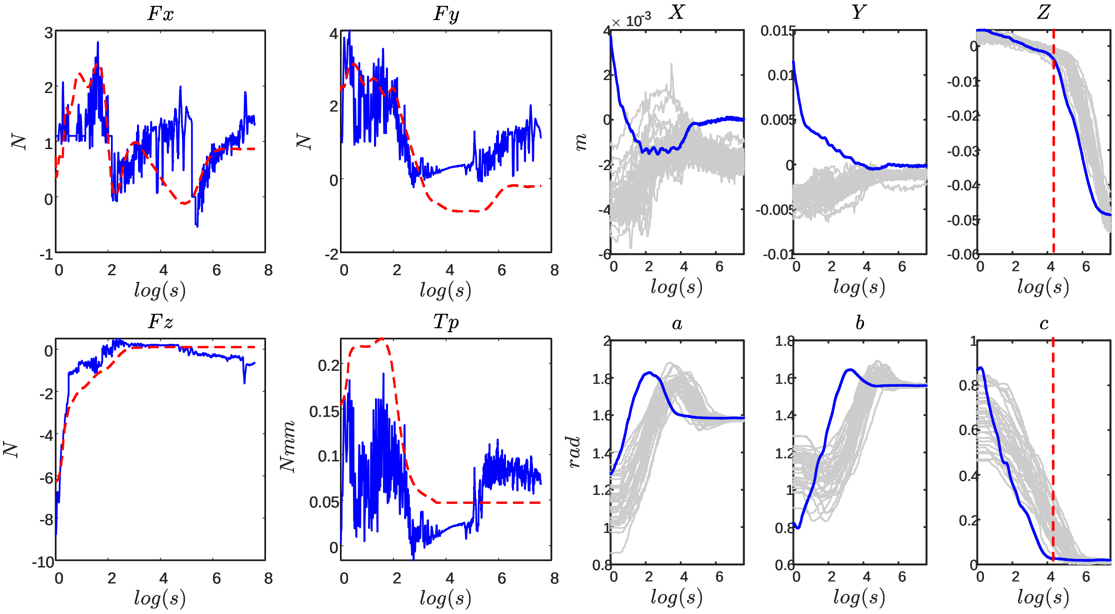


Fig. 19. The force-torque and pose data sampled from the low assembly task compared with demonstration

ficulty of this task is that the soft peg is subject to a certain amount of deformation, resulting in inaccurate positioning. However, the advantage is that the deformation forces due to positional errors are small compared to an aluminium peg because of its soft appearance, making the control frequency requirements for the force tracking adjustment process less demanding. To solve this part of the problem, we adjust the sensitivity force feedback parameters of DMPs and DC-DMPs to make the system more sensitive to the force/torque error; this will eliminate the pose errors caused by the flexible peg surface.

In this part, the angle of the z axis is set as 50° and uses the same demonstration as the high stiffness task to construct the assembly motion model. We omit the hole searching process, because it is the same as the high stiffness part. Fig. 18 shows the whole assembly process. In the beginning, the peg head has been deformed. The robot constantly decreases the error due to deformation as it rotates and inserts. The force-torque profiles are shown as Fig. 19. Since the stiffness of the peg is low, the force and torque range is smaller than those of high stiffness. We also perform 80 sets of experiments to verify our method in different quadrants at different initial angles, and the rate of success is as low as 86.25%. Due to the adhesion and deformation of the silicone surface, it becomes an unknown factor in the assembly process, so these disturbances cannot be taken into account in the DMPs generalization framework.

5. Discussion

This paper introduces a new assembly policy using the learning from demonstration method. The proposed assembly orientation expressed by cosine direction and generalized by DC-DMPs can significantly simplify the generation of the assembly motion in varying task situations.

Although there are important discoveries revealed by studies, it still has some limitations. In the orientation representation we mention in the Section 2.3 that the rotation is simplified by setting the value of γ to 0, which omits the shape of peg or hole. However, it is still valid if we choose square or another polygon object and only additional demonstrations are required. The assembly task based on imitation learning strategy is hard to be applied in higher precision assembly tasks as introduced in [22, 24, 30]. Although the tolerance in our experiment is similar to theirs, our method can be applied to the scenarios with more significant pose deviation, which can meet the needs of most assembly tasks.

Our future research will focus on constructing the correlation model between the shape factor and generalization model, and our plan is to corresponding to the different shapes of hole. Then, we will try to optimize the feedback term to modify the online generalization using DC-DMPs, which may improve the assembly tolerance compared with some higher assembly strategies. We also plan to sample fewer demonstrations to construct the force/torque prediction model to simplify the demonstration part.

6. Conclusion

In this paper, the human assembly skill typically applied in industrial manufacturing is transferred to robots by imitation learning. An assembly-angle representation is proposed to encode the orientation of the round peg-in-hole task. Position and orientation generalization is realized based on the DMPs framework. To consider the relationship among assembly-angle in the DMPs framework, we propose the DC-DMPs method to generalize the movement under variable constraints. We also apply the RBF network to improve assembly force and torque prediction accuracy and utilise force feedback to adjust the generalized motion online. We demonstrate the assembly skill using a particular device for conveniently manipulating and keeping the data authenticity and accuracy. In the search stage, regardless of the initial position, the proposed search strategy can guide the peg to reach the three-point contact state, which is the state of assembly preparation. Finally, in generalization, the relative angle between the axis of peg and hole is set from 20° to 50°. The clearance of peg and hole is 0.3 mm with different materials, achieving a success rate of 93.75% for aluminium and 86.25% for silicone. It is still necessary to optimize the pose representation for different shapes of the peg. In the future, we need to explore the possibility of the assistance of humans for more complex assembly scenarios.

References

[1] 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 Transactions on Industrial Electronics 64 (2017) 6299–6309.

[2] S. R. Chhatpar, M. S. Branicky, Search strategies for peg-in-hole assemblies with position uncertainty, IEEE International Conference on Intelligent Robots and Systems 3 (2001) 1465–1470.

[3] J. A. Marvel, R. Bostelman, J. Falco, Multi-robot assembly strategies and metrics, ACM Computing Surveys 51 (2018).

[4] K. Sharma, P. K. Pal, V. Shirwalkar, Peg-In-Hole search using convex optimization techniques, Industrial Robot 44 (2017) 618–628.

[5] M. Engineering, L. Le, Robotic assembly: chamferless peg-hole assembly W. Haskiya, K. Maycock and J. Knight 17 (1999) 621–634.

[6] K. Sathirakul, R. H. Sturges, Jamming conditions for multiple peg-in-hole assemblies, Robotica 16 (1998) 329–345.

[7] M. Parigi Polverini, A. M. Zanchettin, P. Rocco, A constraint-based programming approach for robotic assembly skills implementation, Robotics and Computer-Integrated Manufacturing 59 (2019) 69–81.

[8] M. W. Abdullah, H. Roth, M. Weyrich, J. Wahrburg, An approach for peg-in-hole assembling using intuitive search algorithm based on human behavior and carried by sensors guided industrial robot, IFAC-PapersOnLine 28 (2015) 1476–1481.

[9] K. Zhang, J. Xu, H. Chen, J. Zhao, K. Chen, Jamming analysis and force control for flexible dual peg-in-hole assembly, IEEE Transactions on Industrial Electronics 66 (2019) 1930–1939.

[10] R. Zhang, Q. Lv, J. Li, J. Bao, T. Liu, S. Liu, A reinforcement learning method for human-robot collaboration in assembly tasks, Robotics and Computer-Integrated Manufacturing 73 (2022) 102227.

[11] 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, arXiv (2019) 8943–8950.

[12] T. Ren, Y. Dong, D. Wu, K. Chen, Learning-based variable compliance control for robotic assembly, Journal of Mechanisms and Robotics 10 (2018) 1–8.

[13] F. Li, Q. Jiang, S. Zhang, M. Wei, R. Song, Robot skill acquisition in assembly process using deep reinforcement learning, Neurocomputing 345 (2019) 92–102.

[14] Y. Fan, J. Luo, M. Tomizuka, A Learning Framework for High Precision Industrial Assembly, arXiv (2018) 0–6.

[15] P. Zou, Q. Zhu, J. Wu, R. Xiong, Learning-based optimization algorithms combining force control strategies for peg-in-hole assembly, IEEE International Conference on Intelligent Robots and Systems (2020) 7403–7410.

[16] Y.-L. Kim, K.-H. Ahn, J.-B. Song, Reinforcement learning based on movement primitives for contact tasks, Robotics and Computer-Integrated Manufacturing 62 (2020) 101863.

[17] S. Schaal, Is imitation learning the route to humanoid robots?, Trends in cognitive sciences 3 (1999) 233–242.

[18] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, J. B. Mouret, A Survey on Policy Search Algorithms for Learning Robot Controllers in a Handful of Trials, IEEE Transactions on Robotics 36 (2020) 328–347.

[19] S. Schaal, Dynamic movement primitives-a framework for motor control in humans and humanoid robotics, in: Adaptive motion of animals and machines, Springer, 2006, pp. 261–280.

[20] P. Pastor, L. Righetti, M. Kalakrishnan, S. Schaal, Online movement adaptation based on previous sensor experiences (2011) 365–371.

[21] F. Stulp, E. Theodorou, M. Kalakrishnan, P. Pastor, L. Righetti, S. Schaal, Learning motion primitive goals for robust manipulation, 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (2011) 325–331.

[22] A. Kramberger, A. Gams, B. Nemec, D. Chrysostomou, O. Madsen, A. Ude, Generalization of orientation trajectories and force-torque profiles for robotic assembly, Robotics and Autonomous Systems 98 (2017) 333–346.

[23] F. J. Abu-Dakka, B. Nemec, A. Kramberger, A. G. Buch, N. Krüger, A. Ude, Solving peg-in-hole tasks by human demonstration and exception strategies, Industrial Robot 41 (2014) 575–584.

[24] F. J. Abu-Dakka, B. Nemec, J. A. Jørgensen, T. R. Savarimuthu, N. Krüger, A. Ude, Adaptation of manipulation skills in physical contact with the environment to reference force profiles, Autonomous Robots 39 (2015) 199–217.

[25] K. Kronander, E. Burdet, A. G. Billard, Task transfer via collaborative manipulation for insertion assembly, Workshop on Human-Robot Interaction, Robotics, Science and Systems (2014).

[26] D. A. Duque, F. A. Prieto, J. G. Hoyos, Trajectory generation for robotic assembly operations using learning by demonstration, Robotics and Computer-Integrated Manufacturing 57 (2019) 292–302.

[27] H.-I. Lin, Design of an intelligent robotic precise assembly system for rapid teaching and admittance control, Robotics and Computer-Integrated Manufacturing 64 (2020) 101946.

[28] T. Tang, H. C. Lin, Y. Zhao, Y. Fan, W. Chen, M. Tomizuka, Teach industrial robots peg-hole-insertion by human demonstration, IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM 2016-Septe (2016) 488–494.

[29] Z. Hou, M. Philipp, K. Zhang, Y. Guan, K. Chen, J. Xu, The learning-based optimization algorithm for robotic dual peg-in-hole assembly, Assembly Automation 38 (2018) 369–375.

[30] J. Song, Q. Chen, Z. Li, A peg-in-hole robot assembly system based on Gauss mixture model, Robotics and Computer-Integrated Manufacturing 67 (2021) 101996.

[31] A. Ude, B. Nemec, T. Petrić, J. Morimoto, Orientation in cartesian space dynamic movement primitives, in: 2014 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 2997–3004.

[32] B. Nemec, F. J. Abu-Dakka, B. Ridge, A. Ude, J. A. Jørgensen, T. R. Savarimuthu, J. Jouffroy, H. G. Petersen, N. Kruger, Transfer of assembly operations to new workpiece poses by adaptation to the desired force profile, 2013 16th International Conference on Advanced Robotics, ICAR 2013 (2013).

[33] N. Liu, X. Zhou, Z. Liu, H. Wang, L. Cui, Learning peg-in-hole assembly using Cartesian DMPs with feedback mechanism, Assembly Automation 40 (2020) 895–904.

[34] D. Kay, K. David, Schaum's Outline of Tensor Calculus, McGraw Hill Professional, 1988.

[35] H.-C. Lin, T. Tang, M. Tomizuka, W. Chen, Remote lead through teaching by human demonstration device, in: Dynamic Systems and Control Conference, volume 57250, American Society of Mechanical Engineers, p. V002T30A003.

[36] A. Duan, R. Camoriano, D. Ferigo, D. Calandriello, L. Rosasco, D. Pucci, Constrained DMPs for Feasible Skill Learning on Humanoid Robots, IEEE-RAS International Conference on Humanoid Robots 2018-Novem (2019) 1032–1038.