Composite dynamic movement primitives based on neural networks for human–robot skill transfer

Weiyong Si[1] · Ning Wang[1] · Chenguang Yang[1]

Abstract

In this paper, composite dynamic movement primitives (DMPs) based on radial basis function neural networks (RBFNNs) are investigated for robots' skill learning from human demonstrations. The composite DMPs could encode the position and orientation manipulation skills simultaneously for human-to-robot skills transfer. As the robot manipulator is expected to perform tasks in unstructured and uncertain environments, it requires the manipulator to own the adaptive ability to adjust its behaviours to new situations and environments. Since the DMPs can adapt to uncertainties and perturbation, and spatial and temporal scaling, it has been successfully employed for various tasks, such as trajectory planning and obstacle avoidance. However, the existing skill model mainly focuses on position or orientation modelling separately; it is a common constraint in terms of position and orientation simultaneously in practice. Besides, the generalisation of the skill learning model based on DMPs is still hard to deal with dynamic tasks, e.g., reaching a moving target and obstacle avoidance. In this paper, we proposed a composite DMPs-based framework representing position and orientation simultaneously for robot skill acquisition and the neural networks technique is used to train the skill model. The effectiveness of the proposed approach is validated by simulation and experiments.

Keywords Position and orientation skill learning framework · Composite dynamic movement primitive · Learning from demonstration · Human–robot skill transfer · Radial basis function NNs (RBFNNs)

1 Introduction

Robot manipulator has been widely used in a number of fields, such as industrial assembly [1], space exploration [2], medical surgery [3] and so on. Specifically, it has been utilised to perform tasks in specific and structured environments due to the advantages of low-cost, efficiency and safety. However, it is hard to program robots for various scenarios, and it is also time-consuming to program each robot manually. As the fast development of machine learning techniques, robot skill learning has attained increasing attention. Several machine learning techniques, e.g., reinforcement learning, imitation learning and deep learning [4, 5], have been successfully employed in robotic skill learning. Among the various learning methods, the learning from demonstration (LfD) (also named programming by demonstration, PbD) has been proved as an effective way to transfer manipulation skills from humans to robots easily [6]. Also, human often has substantial advantages over robots in terms of complex manipulation skill. In contrast to the traditional robot programming methods which require expertise in coding and significant time investment, the attractive aspect of LfD is its capability to facilitate nonexpert robot programming. Thus, the LfD has the potential to significantly benefit a variety of industries, such as manufacturing and health care.

Currently, it is very common for industrial robots to perform accurate position control tasks. However, it is time-consuming to prepare the work environment and robot programs carefully. It often needs to replan the trajectory when any variation happens, such as the changes of

✉ Chenguang Yang
  Charlie.Yang@uwe.ac.uk

[1] Bristol Robotics Laboratory, University of the West of England, Bristol, UK

object positions, the deviation between the real object and the programmed position, limiting the application of automation, such as assembly tasks in the industrial plant. For example in [7], object handover is a common task in human–robot interaction/collaboration, and it is still very challenging on the generalisation, temporal and spatial scaling. In [8], the proposed method can be used to generate a trajectory for the handover task, which could satisfy the shape-driven and goal-driven requirement. It is ensured to achieve the goal and also try to maintain the demonstrated trajectory shape.

The LfD process consists of three phases: the human demonstration, the model learning and skill reproduction. In the demonstration stage, humans teach robots how to execute the tasks with various approaches, such as kinesthetic teaching, teleoperation and passive observation, and the movement profiles of robots and humans will be recorded. In the next learning stage, the manipulation skill models will be trained, which has a significant impact on the performance of robot skill learning and generalisation in practice. The skill model is expected to be modular, compact and adaptive for robotic manipulation skills. There already exists much work to deal with skill modelling for human–robot skills transfer, such as dynamic movement primitives (DMPs) [9, 10], Gaussian mixture model (GMM) [11], the stable estimator of dynamical systems (SEDS) [12], kernelised movement primitives (KMPs) [13], probabilistic movement primitives (ProMPs) [14] and hidden semi-Markov model (HSMM) [15]. And some of these approaches are combined, such as integrating the HSMM with GMM to model the robot skills and perception mechanism [16]. Generally, based on the modelling principle, they can be divided into two branches: dynamic system method and statistic approach. The statistic-based methods include GMM, KMP, ProMPs and HSMM, which could easily represent multimodal sensory information. However, DMP is a general framework to realise the movement planning, online trajectory modification for LfD, which was originally proposed by Ijspeert et al. [17]. As the DMPs have several good characteristics, such as resistance to perturbation and uncertainties, spatial and temporal scaling, they have been gaining much attention. The DMPs approach has the property of generalising the learnt skills to new initial and goal positions, maintaining the desired kinematic pattern. Since the original version of DMPs was proposed, a number of modified versions had been studied to improve the performance of DMPs. Most of these works mainly focus on the two issues, how to improve the generalisation ability of DMPs and how to overcome the inherent drawbacks of DMPs. More recently, it also has been further used to encode different modalities, such as stiffness and force profiles. For example, DMPs with the perceptual term have been proposed to execute physical interaction tasks, which require robots to regulate the contact force, torque, as well as the desired trajectory [18]. Besides, some researchers proposed coupling DMPs to realise obstacle avoidance, interaction with objects and bimanual manipulation by modifying the formulation of DMPs model or adding control methods [19]. The reinforcement learning technique has been used to optimise the parameters of DMPs, which could further improve the generalisation ability of DMPs. RL-based DMPs were proposed to increase the generalisation of the original DMPs [20].

An essential aspect of LfD is how to generalise the learnt skills to novel environments and situations. Since the demonstration cannot cover all the robot working environments and situations, robots need to own the ability to adapt their behaviours according to the changes in environments. The adaptability of robot skills often refers to spatial and temporal scaling, adjusting their behaviours based on the perception information. Such as tracking moving tasks, the robot needs to modify its trajectory based on the position and velocity of the moving target [21]. Besides, many specific tasks require the generalisation of robot skills, such as obstacle avoidance and performing tasks in dynamic environments and situations. Heiko et al. modified the original DMP framework using biologically inspired dynamical systems to increase the generalisation, achieving the real-time goal adaptation and obstacle avoidance [22]. The sensory information has been integrated into the DMP framework to increase the online generalisation, which could generate a robust trajectory account for external perturbations and perception uncertainty [23]. The neural network technique has been utilised to learn the perception term in DMP to realise the reactive planning and control, which can pave the path for robots working in dynamical environments. Further, the modulation of DMP has been exploited using force and tactile feedback to increase the interaction ability and execute bimanual tasks [23]. In addition, a task-oriented regression algorithm with radial basis functions has been proposed to increase the generalisation of DMPs. For dynamic tasks, such as tracking moving targets, it can be seen that many researchers proposed modified versions of DMPs to deal with moving goals. For example in [21], the authors modified the DMP by adjusting the temporal scaling parameter online to follow a moving goal, although it only focused on the position trajectory in Cartesian space. To improve the generalisation, single DMP could not produce complex behaviour. Merging the different DMP is very important to deal with this challenge [24]. It also pointed out that building a motion skill library for robots to produce complex behaviour is a useful tool. Also, the merging sequential motion primitives have been studied to produce complex behaviours. Complex trajectories involving

several actions can be reproduced by sequencing multiple motion primitives. Each motion primitive is represented as DMP; various approaches were investigated to connect the motion primitives seamlessly [24]. However, most of the current work focused on the position motion primitives in Cartesian or joint space, and there is a lack of research on the orientation primitive trajectories in Cartesian.

Most recently, various versions of DMP have been proposed to increase the online adaptability to uncertainties and novel tasks. However, the spatial scaling is limited due to encoding the position trajectory for each coordinate. Most of the existing work focused on DMPs representing position skills in Cartesian space, ignoring the orientation requirements in some application, such as obstacle avoidance [22, 25], picking and placing [10], cutting task [9]. However, for some tasks, such as ultrasound scanning in medical application, the probe orientation has a significant impact on the image quality in robot-assisted ultrasonography; hence the orientation and position need to be considered simultaneously. The forcing term in DMPs often is approximated by the Gaussian functions, and the locally weighted regression (LWR) technique to be used to learn the weights of each basis functions. However, [26] stated that forcing term approximation could influence the accuracy and performance of DMP. And different basis functions have been studied to improve the performance of DMPs.

In this work, a composite DMPs-based skill learning framework is studied, which considers not only the position constraints but also the orientation requirement. Both temporal and spatial generalisation capability has been increased. Besides, the DMP-based framework can be adapted temporally to moving targets with the specific requirement of orientation. Further, the RBFNNs are utilised to learn the nonlinear functions in composite DMPs.

The contributions in this work are (1) combining the DMPs and RBFNNs to improve the generalisation of robot manipulation skills. The radial basis function NN is employed to approximate the force term in the composite DMPs. (2) A basic skill associated with position and orientation could be modelled by the composite DMP simultaneously, coupled with the temporal parameter. (3) The composite DMP could reach the moving goals with generalisation in terms of temporal and spatial scaling. The composite DMP-based framework can guarantee to converge to moving goals while being perturbed to obstacles.

The rest of the paper is organised as follows. Section 2 provides an overview of the position and orientation DMP in Cartesian space and its limitations. The composite DMPs framework based on RBFNNs is presented in Sect. 3. The stability analysis for the DMP-based model is presented.

Section 4 presents the simulation and experimental results to validate the temporal and spatial generalisation. RBFNNs have been utilised to learn the nonlinear functions associated with the combined DMPs. Section 6 concludes the paper finally.

2 Preliminaries and motivations

2.1 Radial basis function neural networks (RBFNNs)

The neural network has been proved to an effective approach to robot applications, and much work on the neural network has been studied, such as the stability of neural network [27, 28]. RBFNNs are a useful tool to approximate nonlinear functions for robot control and robot skills learning. For instance, RBFNNs is combined with the broad learning framework to learn and generalise the basic skills [29]. RBFNNs are employed to approximate the nonlinear dynamics of the manipulator robot to improve tracking performance [16, 30]. Therefore, RBFNNs can approximate the nonlinear forcing term in the DMP framework. Radial basis function networks consist of three layers: an input layer, a hidden layer with a nonlinear RBF activation function and a linear output layer. It is an effective approach to approximate any continuous function $h : R^n \rightarrow R$,

$$h(x) = W^T S(x) + \varepsilon(x) \tag{1}$$

where $x \in R^n$ is the input vector, $W = [\omega_1, \omega_2, \ldots, \omega_N]^T \in R^N$ denotes the weight vector for the N neural network nodes. The approximation error $\varepsilon(x)$ is bound. $S(x) = [s_1(x), s_2(x), \ldots, s_N(x)]^T$ is a nonlinear vector function, where $s_i(x)$ can be defined as a radial basis function,

$$s_i(x) = \exp(-h_i(x - c_i)^T (x - c_i)) \qquad i = 1, 2, \ldots, N \tag{2}$$

where $c_i = [c_{i1}, c_{i2}, \ldots, c_{in}]^T \in R^n$ denotes the centres of the Gaussian function and $h_i = 1/\chi_i^2$, χ_i denotes the variance. The ideal weight vector W is defined as,

$$W = \underset{\hat{W} \in R^N}{\arg \min} \left\{ \sup \left| h(x) - \hat{W}^T S(x) \right| \right\} \tag{3}$$

which minimises the approximation error of nonlinear function. The nonlinear functions in DMPs can be learnt by RBFNNs from demonstration data. In this work, RBFNNs will be utilised to parameterise the nonlinear functions in DMPs.

2.2 Position and orientation DMP in Cartesian space

DMP is a useful tool to encode the movement profiles via a second-order dynamical system with a nonlinear forcing term. Robots skills learning by DMPs aims to model the forcing term in such a way to be able to generalise the trajectory to a new start and goal position while maintaining the shape of the learnt trajectory. DMPs can be used to model both periodic and discrete motion trajectories. However, in this work, we will focus on the discrete motion trajectories. Currently, the most research on DMPs mainly focuses on the position DMPs and its modifications, which can be used to represent arbitrary movements for robots in Cartesian or joint space by adding a nonlinear term to adjust the trajectory shape. For one degree of multiple-dimensional dynamical systems, the transformation system of position DMP can be modelled as follows [31],

$$\tau_s \dot{v} = \alpha_z(\beta_z(p_g - p) - v) + F_p(x) \tag{4}$$

$$\tau_s \dot{p} = v \tag{5}$$

where the p_g is the desired position, p is the current position, the v is the scaled velocity, τ_s is the temporal scaling parameter, α_z, β_z are the design parameters, usually, $\alpha_z = 4\beta_z$. $F_p(x)$ is the nonlinear forcing term responsible for tuning the shape of trajectory. The $F_p(x)$ can be approximated by a set of radial basic functions,

$$F_p(x) = \frac{\sum_{i=1}^{N} \psi_i(x) w_i}{\sum_{i=1}^{N} \psi_i(x)} x(p_g - p_0) \tag{6}$$

$$\psi_i(x) = \exp(-h_i(x - c_i)^2) \tag{7}$$

where $\psi_i(x)$ is a Gaussian radial basis function with the centre c_i and width h_i; p_0 is the initial position, and w_i is the weight learning from demonstration. The phase variable x is determined by the canonical system, which can be represented as follows,

$$\tau_s \dot{x} = -\alpha_x x, \qquad x \in [0, 1]; \qquad x(0) = 1 \tag{8}$$

where α_x is a positive gain coefficient, τ_s is the temporal scaling parameter and the $x_0 = 1$ is the initial value of x, which can converge to 0 exponentially. For the multiple-degree-of-freedom (DoF) dynamic system, each dimension can be modelled by a transformation system, but they share a common canonical system to synchronise them.

The orientation DMP has been first proposed by [32], which is vital to robot learning and control. The orientation in DMP is often represented by rotation matrix or quaternions. For example in [33], the unit quaternions are used to model the orientation, and the unit quaternion set minus one single point also has been proved to be contractible

[34]. This property of the unit quaternion set could guarantee the convergence of orientation DMPs. In addition, as the quaternion formulation has less variable than the rotation matrix, it has been used widely in the orientation representation for robot learning and control. In [32], the unit quaternion-based transformation system can be described as,

$$\tau_s \dot{z} = -\alpha_z(\beta_z 2 \log(q_g * \bar{q}) - z) + F_o(x) \tag{9}$$

$$\tau_s \dot{q} = \frac{1}{2} \begin{bmatrix} 0 \\ z \end{bmatrix} * q \tag{10}$$

where $q \in S^3$ denotes the orientation as a unit quaternion, $q_g \in S^3$ represents the final orientation. ω denotes the angular velocity, $z = \tau_s \omega \in R^3$ is the scaled angular velocity, '*' denotes the quaternion product, $\bar{q}$ represents the quaternion conjugate which is equal to the inverse quaternion for unit quaternions and $2 \log(q_2 * \bar{q}_1) \in R^3$ denotes the rotation of q_1 around a fixed axis to reach q_2. The forcing term $F_o(x) \in R^3$ for each orientation coordinate will learn the desired orientation skills from the demonstration data.

3 Composite position and orientation dynamic movement primitives

Currently, the separate position or orientation DMP has been studied widely [35]; however, research on the composite DMPs, modelling the position and orientation simultaneously, is not common. In real practice, most manipulation skills often mix the position and orientation skills, which requires robots to satisfy the specific position constraints as well the orientation for many tasks, such as polishing, spraying, assembly [36, 37]. In addition, for human–robot interaction tasks, such as two partners collaborating an object handover interaction, the target position is always changing. It is still open to guarantee various orientation requirements. Inspired by the improvement in the orientation DMP, the proposed framework has great generalisation and adaptability to novel tasks and situations. Studying on the DMPs to handle the moving goals is also vital to the practical application. Therefore, we propose the composite DMPs, coupling the position and orientation modelling in a framework, and the RBFNNs are used to learn the nonlinear term in models.

3.1 The composite DMP formulation

As shown in Fig. 1, the manipulation skill modelled by position and orientation DMPs consisted of recoding the demonstration data, training the RBFNNs and reproducing the skills. The demonstration data include the position and
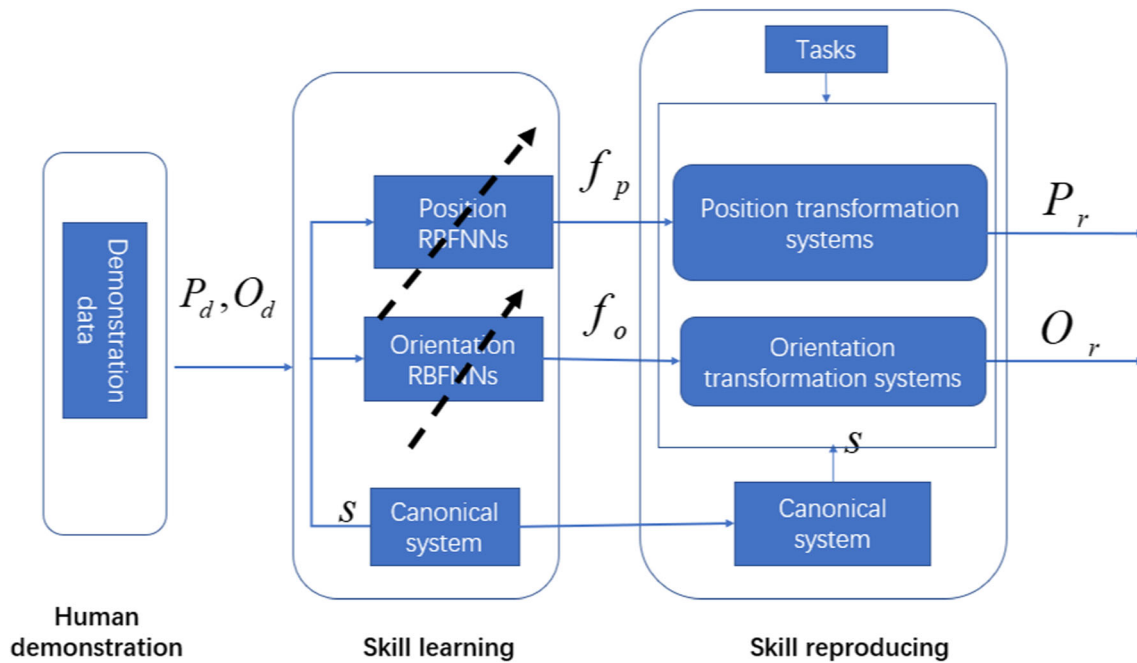
Fig. 1 Structure of human–robot skill transfer using the composite DMP model

orientation trajectories, and the output of skills reproducing is the reference of position and orientation trajectories associated with specific tasks. The canonical system is used to coordinate the position and orientation constrains in the composite DMPs. The nonlinear forcing terms associated with each DMP are trained by using RBFNNs from the position and orientation demonstration data. Six RBFNNs are used to parameterise the nonlinear functions for position and orientation DMPs, respectively. After the DMPs have learned the demonstration, the dynamic and multiple constraints can be guaranteed: (1) the goal and initial position and orientation can be changed; (2) the targets can be moved, the velocity profiles of DMP output will keep in a safe bound; and (3) the requirements of position and orientation can be achieved simultaneously.

The position DMP formulation can be described as,

$$\tau_s \dot{v} = -\alpha_z(\beta_z e_p + v) + diag(p_g - p_0)f_p(x) \tag{11}$$

$$\tau_s \dot{e}_p = v \tag{12}$$

where $e_p = p_g - p \in R^3$ is the position error and $v \in R^3$ is the scaled velocity error. The α_z, β_z are positive gains, and $f_p(x)$ is trained by RBFNNs for each DMP. The system is trained using a demonstration from the initial position $p_{0,d}$ to the stationary goal $p_{g,d}$ with temporal scaling τ_d.

The orientation DMP formulation can be described as [33],

$$\tau_s \dot{z} = -\alpha_z(\beta_z e_o + z) + diag(q_g * \bar{q}_0)f_o(x) \tag{13}$$

$$\tau_s \dot{e}_o = z \tag{14}$$

$$e_o = 2\log(q_g * \bar{q}) \tag{15}$$

where the e_o is the quaternion error, z is the scaled quaternion error velocity. To obtain the orientation, we solve equation (15),

$$q = \overline{\exp(\frac{1}{2}e_o)} * q_g \tag{16}$$

The angular velocity is

$$\omega = 2vec(\dot{q} * \bar{q}) \tag{17}$$

where the $\dot{q}$ can be obtained by the following equations,

$$\dot{q} = -\frac{1}{2}q * \bar{q}_g * J_{\log q}(q_g * \bar{q})\dot{e}_o * q \tag{18}$$

$$\dot{e}_o = -2J_q(q_g * \bar{q})(q_g * \bar{q} * \dot{q} * \bar{q}) \tag{19}$$

Inspired by the work [38], the temporal scaling can be adjusted based on the task and the velocity constraints. The target position and velocity update the shared temporal parameter in the position and orientation DMPs, and it may be described as [21],

$$\dot{\tau}_s = -\gamma(\tau_s - \tau_a) + \dot{\tau}_a \tag{20}$$

where the γ is a design parameter, τ_a is determined by

$$\tau_a = \frac{\|e\|}{\|e_d\|} * \tau_d \tag{21}$$

$$e = \begin{bmatrix} e_p^T, & e_o^T \end{bmatrix}^T$$
$$e_d = \begin{bmatrix} e_{p,d}^T, e_{o,d}^T \end{bmatrix}^T \tag{22}$$

where the e_p is the position error between the goal and the initial point, e_o is the orientation error between goal and start. The $e_{p,d}$ is the position error between the goal and the initial point in the demonstration, $e_{o,d}$ is the orientation error between the goal and start in the demonstration. τ_d is the temporal scaling coefficient in the demonstration. The temporal parameter update law has been proved to converge to the moving goals in [21].

3.2 The training of DMPs by RBFNNs

Take one dimension for position and orientation DMP as examples. The nonlinear forcing terms of position and orientation DMP can be approximated by RBFNNs respectively,

$$f_p(s) = \sum_i w_p^i \psi^i(s) \tag{23}$$

$$f_o(s) = \sum_j w_o^j \psi^j(s) \tag{24}$$

w_p^i, w_o^j are the weight coefficients, $\psi^i(s)$ and $\psi^j(s)$ are the Gaussian activation functions, defined as,

$$\psi^i(s) = \exp(-h^i(s - c^i)^2) \tag{25}$$

$$\psi^j(s) = \exp(-h^j(s - c^j)^2) \tag{26}$$

In the demonstration phase, one position trajectory $p_d, \dot{p}_d, \ddot{p}_d$ is recorded, from starting position $p_{0,d}$, to the target position $p_{g,d}$. According to the position DMP transformation system Eqs. (11), (12) and the demonstration data, the desired force function is

$$f_p^d(s) = \frac{1}{p_{g,d} - p_{0,d}} (\tau_d^2 \ddot{p}_d - \alpha_z(\beta_z(p_{g,d} - p_d) - \tau_d \dot{p}_d)) \tag{27}$$

where the τ_d is the temporal scaling during demonstration. Similarly, the force term in the orientation DMPs can be described as,

$$f_o^d(s) = (diag(2\log(q_{g,d} * \bar{q}_{0,d})))^{-1} \\ *(\tau_d^2 \dot{\omega}_d - \alpha_z(\beta_z(2\log(q_{g,d} * \bar{q}_d) - \tau_d \omega_d))) \tag{28}$$

The following error function between the desired force term and the approximated value is the objective function of the optimisation problem, which will be minimised for learning the parameters of RBFNNs in the DMPs:

$$E = \frac{1}{2}((f_p^d(s^t) - f_p(s^t))^2 + (f_o^d(s^t) - f_o(s^t))^2) \tag{29}$$

s^t is the value of s. A gradient descent approach is used to derive the weight update law as [39],

$$w_p^i(t+1) = w_x^i(t) - \lambda_1 \frac{\partial E}{\partial w_p^i} \tag{30}$$

$$w_o^j(t+1) = w_o^j(t) - \lambda_2 \frac{\partial E}{\partial w_o^j} \tag{31}$$

$$\frac{\partial E}{\partial w_p^i} = (f_p^d(s^t) - f_p(s^t)) \frac{\partial}{\partial w_p^i}(-\sum_i w_p^i \psi^i(s)) \\ = (f_p^d(s^t) - f_p(s^t))(-\psi^i(s^t)) \tag{32}$$

The weight update law of w_p^i is given as,

$$w_p^i(t+1) = w_p^i(t) + \lambda_1(f_p^d(s^t) - f_p(s^t))\psi^i(s^t) \tag{33}$$

Similarly, the weight w_o^i is updated by,

$$w_o^j(t+1) = w_o^j(t) + \lambda_2(f_o^d(s^t) - f_o(s^t))\psi^j(s^t) \tag{34}$$

The weights in the RBFNNs can be attained through the gradient descent approach and demonstration data.

Fig. 2 Six-DoF Omni Phantom

Table 1 Parameters in DMP

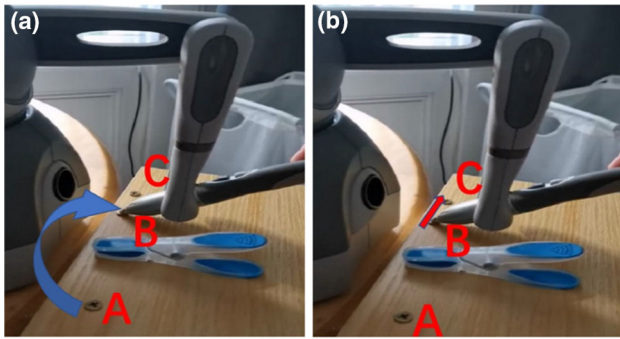| Parameter	Descriptions	Value
N	Number of RBFNNs nodes	10
α_z	Coefficient of DMP	60
β_z	Coefficient of DMP	15
λ_1	Learning rate for position	0.1
λ_2	Learning rate for orientation	0.15
γ	Coefficient for tracking target	2

Fig. 3 **a** Represents the human demonstration from Point A to Point B; **b** represents the target goal moving from Point B to Point C

4 Experimental results

As a complex task can be hierarchically decomposed into different subtasks involving multiple primitive actions and manipulated objects, several basic motion skills could be synthesised to complex tasks. Thus, in the paper, we will conduct several typical motion skills through simulation and experiments. As shown in Fig. 2, Omni Phantom is an input device for human–robot skill transfer, which has been used in the teleoperation applications. This haptic device could provide the operator force feedback when interacting with the objects or the environments. In this paper, the Omni Phantom is used to acquire training data of human demonstration in 3D Cartesian space for training the DMP model.
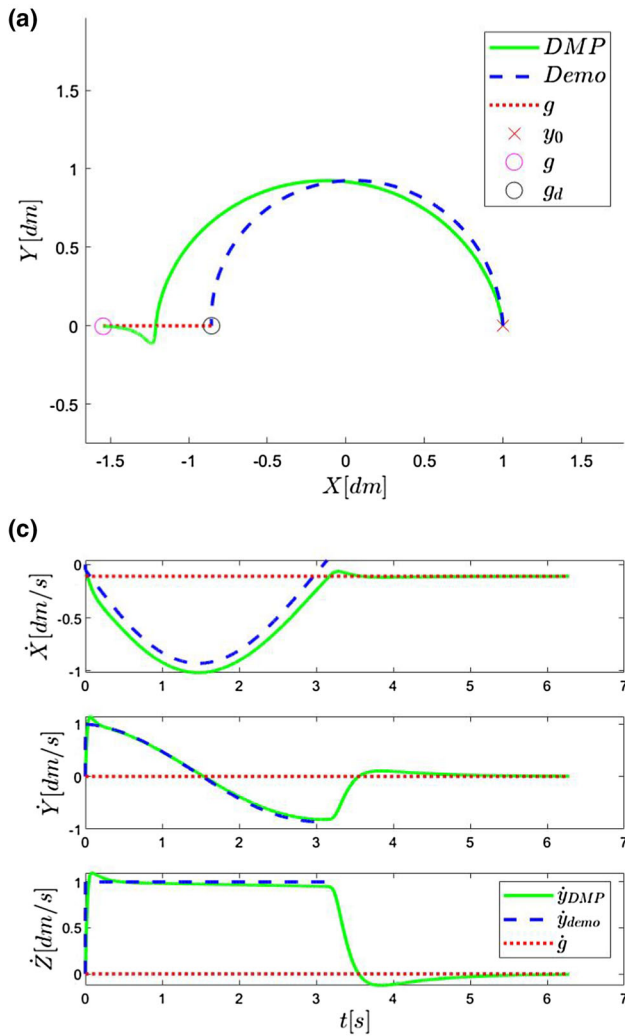
Fig. 4 In (**a**), the blue line is the human demonstration trajectory; the green line is the reproduced trajectory by DMPs; the red dash line represents the moving target. **b** The position trajectory generated by demonstration and DMP; the red dash line is the goal's position trajectory. **c** The velocity trajectory generated by demonstration and DMP; the red dash line is the goal's velocity trajectory. **d** Provides the evolution of the temporal coefficient and its derivative
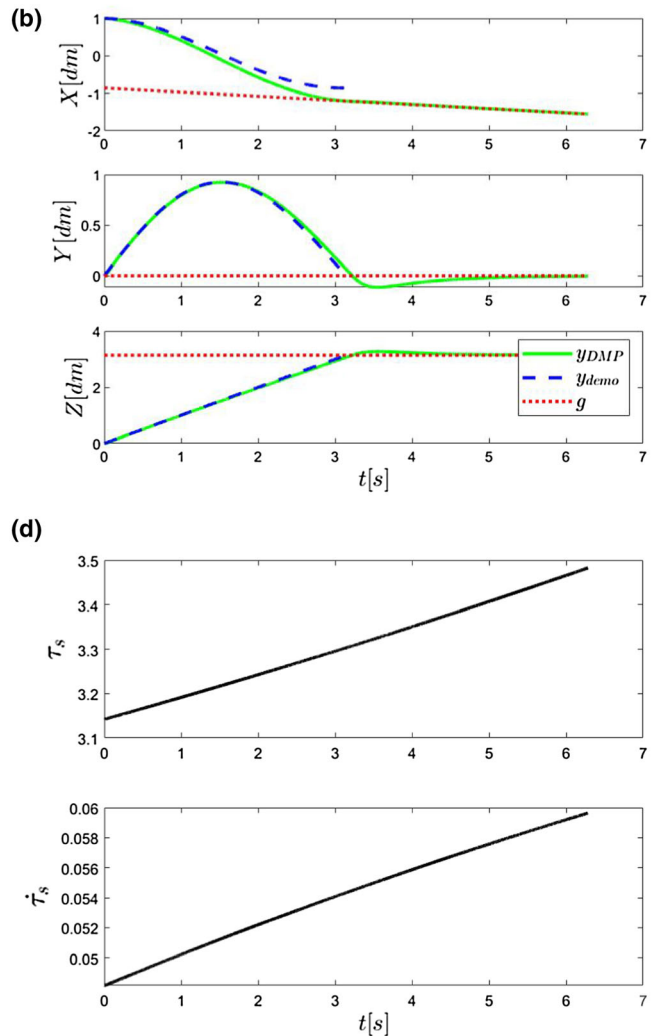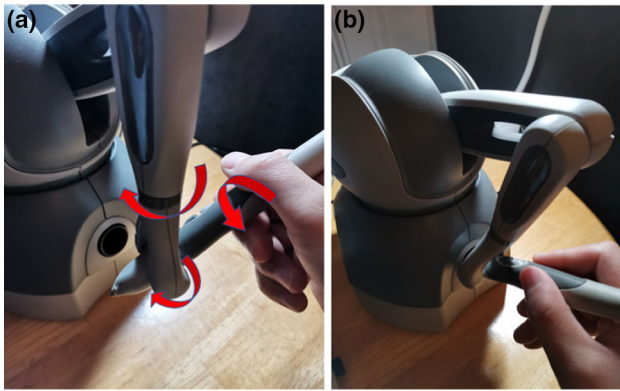
Fig. 5 Human demonstrating to change the pose of the Omni from (a) to (b) through the three orientation joints (red arrow)

We use Omni Phantom to demonstrate trajectories, including the position and orientation in Cartesian space. We then used the demonstrated data to train all DMPs and executed the DMPs with new start and goal position and orientation. Omni Phantom can record the position and pose of the end. During the demonstration, both the position and orientation trajectories are recorded, and used to train the skill model. In the execution, we modify the desired task to test the generalisation performance. The parameters in DMP are shown in Table 1.

4.1 Spatial scaling of composite DMP

To demonstrate the spatial generalisation ability, we carried out simulation experiments to test the composite DMPs. When the DMP reproduces trajectory, we set a new goal position; the proposed DMPs could converge to the

desired position. We test the spatial generalisation of DMPs through the task shown in Fig. 3, simulating the picking and placing skill in the industrial case. First, we demonstrate an obstacle-free trajectory from point A to point B for robots. However, when the robot performs the task, the target moves from Point B to Point C. Our experiment assumes the target moves from B to C at a constant velocity, which is known. The position DMP could generate one trajectory online to adapt the dynamic tasks.

In Fig. 4, the trajectory generated by DMPs could reach the desired position of the moving goal even when we learn the DMP using a static goal. (a) shows the human demonstration trajectory and trajectory reproduced by DMP. Although the goal is moving, the trajectory generated by DMP maintains the shape of the demonstration. The red dash line in (c) represents the target velocity, and the green line is the velocity trajectory generated by DMP. From the (d), it can be seen that the temporal scaling parameter τ_s is increasing. In the beginning, since the target velocity is relatively high, the rate of change of τ_s is also relatively large, until it decreases to zero. When the target does not move, the τ_s does not change. Since the temporal scaling coefficient is tuned based on the goal's position and velocity, it could achieve the target and maintain the demonstrated shape. In original DMP, the temporal scaling parameter is fixed; hence, it is hard to deal with the dynamic perturbance, such as the moving target and the stopping by an obstacle. Therefore, the composite DMP could adapt to a dynamic environment and tasks based on the position and velocity of the goal.
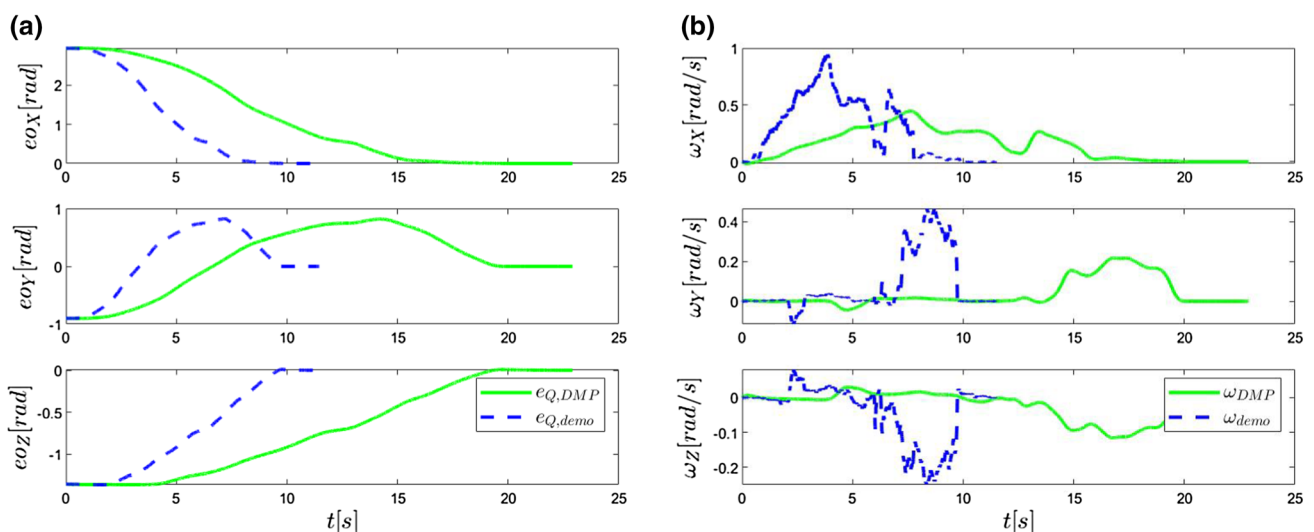

Fig. 6 **a** Pose of Omni: the blue line is the demonstration trajectory; the green one is the output of DMP when the execution time is set twice the demonstration one; **b** provides the angular velocity generated by demonstration and DMP

(a)

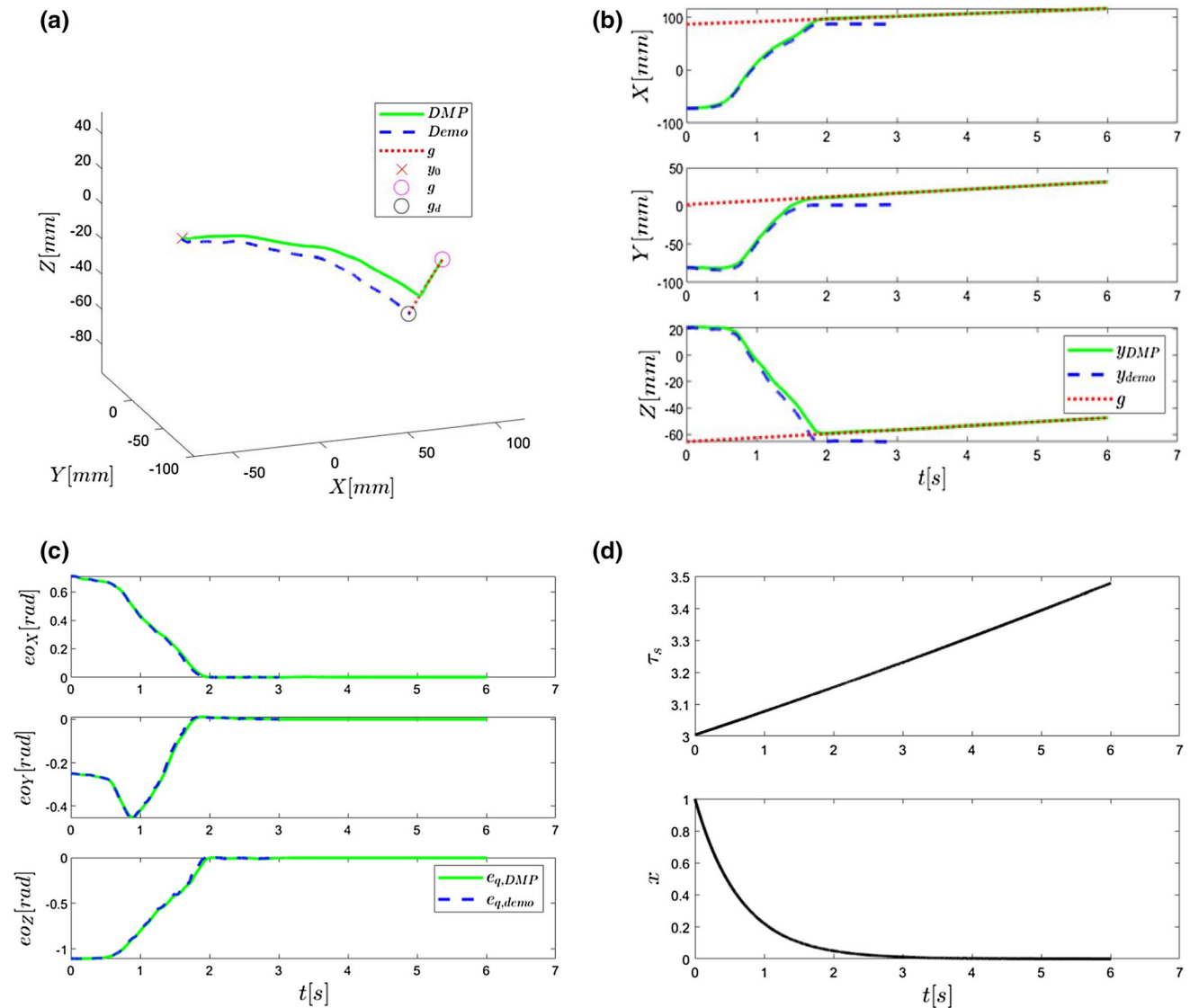**(b)**

(c)

(d)

Fig. 7 **a** 3D trajectory. The blue and green lines in **b** show the demonstration and DMP trajectory in each direction; the red dash line is the goal trajectory in XYZ directions. **c** The orientation error between the current orientation and the goal orientation. **d** provides the evolution of the temporal coefficient and the phase variable x with time

4.2 The temporal scaling of orientation DMP

It often requires robots to satisfy the orientation requirements when the robot coordinates with humans or other robots in the industrial application. To demonstrate the temporal scaling ability in the orientation of composite DMP, we modify the execution duration when DMP is reproducing the trajectory. As shown in Fig. 5, a human demonstrates how to change Omni's orientation from (a) to (b) and the demonstration data are recorded for training the orientation DMP. In reproduced period, we set the duration time as twice, and the result is shown in Fig. 6.

From the (a) in Fig. 6, we can find the orientation DMPs can be scaled temporally, and since the execution time is

longer, the angular velocity is slower than the demonstration one. The orientation scaling could be achieved by adjusting the temporal parameter. When the temporal coefficient τ_s is twice, the execution time is double, and the trajectory shape is maintained. Also, from the (b), the angular velocity trajectory has the same pattern with the demonstrated one, when modifying the execution time. The trajectory is also smooth and can be adjusted temporally based on the task requirement and perception information on the external environment. This property could be used to adjust the orientation dynamically and satisfy the orientation requirement. When the DMP couples the position and orientation, the temporal coefficient is adjusted based

on the position and orientation tasks, and the external environment.

4.3 The performance of composite DMP for a moving goal

For the tasks with position and orientation constraints, the composite DMPs between the position and orientation are necessary. Test the performance of composite DMPs to the tasks requiring the position and orientation simultaneously. For this case, we first demonstrate a trajectory involving the position and orientation and then train the composite DMPs using the demonstration data. During the reproducing stage, the DMPs need to generate position and orientation trajectory for the moving goal and satisfy the orientation constrains. The performance of reaching a moving target with orientation constraint can be found in Fig. 7.

Through (a) and (b) in Fig. 7, the trajectory generated by DMPs could reach the moving goal with the desired orientation. Due to the moving goal, the temporal scaling τ_s is increasing. Although the target has a constant velocity, the shape of position and orientation is consistent with the demonstration. Due to the goal's velocity, the temporal scaling is increasing, which guarantees the velocity shape is similar to the learned pattern. The position and orientation constraints are satisfied simultaneously. For the composite DMP, because the goal's motion information could influence the temporal scaling and phase variable, it could influence the shape of trajectory. The position and orientation could be coupled and adjusted based on the task and the external environments through the temporal scaling. The proposed composite DMP considers the moving goal and the orientation requirements simultaneously.

5 Conclusion

This paper proposed composite DMPs, coupling the position and orientation representation simultaneously and using the RBFNNs to approximate the nonlinear forcing term in DMPs. The composite DMPs can track moving goals and guarantee the velocity stays in a safe range. The generalisation performance of temporal and spatial scaling is validated through several primitive skills. In the future, we will consider extending the DMPs to model various pieces of sensory information, making the DMPs interact with the environment. It also can be used in the cooperation manipulation tasks for the bimanual manipulator.

Compliance with ethical standards

References

1. Qiao H, Wang M, Su J, Jia S, Li R (2014) The concept of "attractive region in environment" and its application in high-precision tasks with low-precision systems. IEEE/ASME Trans Mechatron 20(5):2311–2327
2. Park IW, Smith T, Sanchez H, Wong SW, Piacenza P, Ciocarlie M (2017) Developing a 3-dof compliant perching arm for a free-flying robot on the international space station. In: 2017 IEEE international conference on advanced intelligent mechatronics (AIM). IEEE, pp 1135–1141
3. Su H, Yang C, Ferrigno G, De Momi E (2019) Improved human–robot collaborative control of redundant robot for teleoperated minimally invasive surgery. IEEE Robot Autom Lett 4(2):1447–1453
4. Su H, Qi W, Hu Y, Karimi HR, Ferrigno G, De Momi E (2020) An incremental learning framework for human-like redundancy optimization of anthropomorphic manipulators. IEEE Trans Ind Inform. https://doi.org/10.1109/TII.2020.3036693
5. Qi W, Su H, Aliverti A (2020) A smartphone-based adaptive recognition and real-time monitoring system for human activities. IEEE Trans Hum Mach Syst 50(5):414–423
6. Ravichandar H, Polydoros AS, Chernova S, Billard A (2020) Recent advances in robot learning from demonstration. Annu Rev Control Robot Autonom Syst 3(1):297–330
7. Prada M, Remazeilles A, Koene A, Endo S (2013) Dynamic movement primitives for human–robot interaction: comparison with human behavioral observation. In: 2013 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 1168–1175
8. Prada M, Remazeilles A, Koene A, Endo S (2014) Implementation and experimental validation of dynamic movement primitives for object handover. In: 2014 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 2146–2153
9. Yang C, Zeng C, Fang C, He W, Li Z (2018) A dmps-based framework for robot learning and generalization of humanlike variable impedance skills. IEEE/ASME Trans Mechatron 23(3):1193–1203

10. Yang C, Zeng C, Cong Y, Wang N, Wang M (2018) A learning framework of adaptive manipulative skills from human to robot. IEEE Trans Ind Inform 15(2):1153–1161

11. Calinon S, Guenter F, Billard A (2007) On learning, representing, and generalizing a task in a humanoid robot. IEEE Trans Syst Man Cybern Part B (Cybernetics) 37(2):286–298

12. Khansari-Zadeh SM, Billard A (2011) Learning stable nonlinear dynamical systems with gaussian mixture models. IEEE Trans Robot 27(5):943–957

13. Huang Y, Rozo L, Silvério J, Caldwell DG (2019) Kernelized movement primitives. Int J Robot Res 38(7):833–852

14. Paraschos A, Daniel C, Peters JR, Neumann G (2013) Probabilistic movement primitives. In: Advances in neural information processing systems, pp 2616–2624

15. Zeng C, Yang C, Cheng H, Li Y, Dai SL (2020) Simultaneously encoding movement and semg-based stiffness for robotic skill learning. IEEE Trans Ind Inform 17(2):1244–1252

16. Yang C, Chen C, He W, Cui R, Li Z (2018) Robot learning system based on adaptive neural control and dynamic movement primitives. IEEE Trans Neural Netw Learn Syst 30(3):777–787

17. Ijspeert AJ, Nakanishi J, Schaal S (2002) Movement imitation with nonlinear dynamical systems in humanoid robots. In: Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292). IEEE, vol 2, pp 1398–1403

18. Pastor P, Righetti L, Kalakrishnan M, Schaal S (2011) Online movement adaptation based on previous sensor experiences. In: 2011 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 365–371

19. Gams A, Nemec B, Ijspeert AJ, Ude A (2014) Coupling movement primitives: interaction with the environment and bimanual tasks. IEEE Trans Robot 30(4):816–830

20. Tamosiunaite M, Nemec B, Ude A, Wörgötter F (2011) Learning to pour with a robot arm combining goal and shape learning for dynamic movement primitives. Robot Autonom Syst 59(11):910–922

21. Koutras L, Doulgeri Z (2020) Dynamic movement primitives for moving goals with temporal scaling adaptation. In: 2020 IEEE international conference on robotics and automation (ICRA). IEEE, pp 144–150

22. Hoffmann H, Pastor P, Park DH, Schaal S (2009) Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance. In: 2009 IEEE international conference on robotics and automation. IEEE, pp 2587–2592

23. Chebotar Y, Kroemer O, Peters J (2014) Learning robot tactile sensing for object manipulation. In: 2014 IEEE/RSJ international conference on intelligent robots and systems. IEEE, pp 3368–3375

24. Saveriano M, Franzel F, Lee D (2019) Merging position and orientation motion primitives. In: 2019 international conference on robotics and automation (ICRA). IEEE, pp 7041–7047

25. Ginesi M, Meli D, Calanca A, Dall'Alba D, Sansonetto N, Fiorini P (2019) Dynamic movement primitives: volumetric obstacle avoidance. In: 2019 19th international conference on advanced robotics (ICAR). IEEE, pp 234–239

26. Wang R, Wu Y, Chan WL, Tee KP (2016) Dynamic movement primitives plus: for enhanced reproduction quality and efficient trajectory modification using truncated kernels and local biases. In: 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, pp 3765–3771

27. Qiao H, Peng J, Xu ZB (2001) Nonlinear measures: a new approach to exponential stability analysis for hopfield-type neural networks. IEEE Trans Neural Netw 12(2):360–370

28. Peng J, Qiao H, Zb Xu (2002) A new approach to stability of neural networks with time-varying delays. Neural Netw 15(1):95–103

29. Huang H, Zhang T, Yang C, Chen CP (2019) Motor learning and generalization using broad learning adaptive neural control. IEEE Trans Ind Electron 67(10):8608–8617

30. Wang N, Chen C, Yang C (2020) A robot learning framework based on adaptive admittance control and generalizable motion modeling with neural network controller. Neurocomputing 390:260–267

31. Ijspeert AJ, Nakanishi J, Hoffmann H, Pastor P, Schaal S (2013) Dynamical movement primitives: learning attractor models for motor behaviors. Neural Comput 25(2):328–373

32. Ude A, Nemec B, Petrić T, Morimoto J (2014) Orientation in cartesian space dynamic movement primitives. In: 2014 IEEE international conference on robotics and automation (ICRA). IEEE, pp 2997–3004

33. Koutras L, Doulgeri Z (2020) A correct formulation for the orientation dynamic movement primitives for robot control in the cartesian space. In: Conference on robot learning, pp 293–302

34. Karlsson M, Robertsson A, Johansson R (2018) Convergence of dynamical movement primitives with temporal coupling. In: 2018 European control conference (ECC). IEEE, pp 32–39

35. Liu N, Zhou X, Liu Z, Wang H, Cui L (2020) Learning peg-in-hole assembly using cartesian dmps with feedback mechanism. Assembly Autom 40(6):895–904

36. Wen X, Chen H (2020) 3d long-term recurrent convolutional networks for human sub-assembly recognition in human–robot collaboration. Assembly Autom 40(4):655–662

37. Huang Y, Zheng Y, Wang N, Ota J, Zhang X (2020) Peg-in-hole assembly based on master-slave coordination for a compliant dual-arm robot. Assembly Autom 40(2):189–198

38. Dahlin A, Karayiannidis Y (2019) Adaptive trajectory generation under velocity constraints using dynamical movement primitives. IEEE Control Syst Lett 4(2):438–443

39. Sharma RS, Shukla S, Karki H, Shukla A, Behera L, Venkatesh K (2019) Dmp based trajectory tracking for a nonholonomic mobile robot with automatic goal adaptation and obstacle avoidance. In: 2019 international conference on robotics and automation (ICRA). IEEE, pp 8613–8619