Robot skill learning system of multi-space fusion based on dynamic movement primitives and adaptive neural network control

Chengguo Liu [a,c], Guangzhu Peng [b], Yu Xia [a], Junyang Li [a], and Chenguang Yang [c*]

[a] *State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing 400044, China.*

[b] *School of Automation, Nanjing University of Information Science and Technology, Nanjing 210044, China.*

[c] *Bristol Robotics Laboratory, University of the West of England, Bristol BS16 1QY, United Kingdom.*

ARTICLE INFO

Keywords:
Multi-space skill
Riemannian manifold
Dynamic movement primitive
Adaptive neural network
Variable admittance control

ABSTRACT

This article develops a robot skill learning system with multi-space fusion, simultaneously considering motion/stiffness generation and trajectory tracking. To begin with, surface electromyography (sEMG) signals from the human arm is captured based on the MYO armband to estimate endpoint stiffness. Gaussian Process Regression (GPR) is combined with dynamic movement primitive (DMP) to extract more skills features from multi-demonstrations. Then, the traditional DMP formulation is improved based on the Riemannian metric to encode the robot's quaternions with non-Euclidean properties. Furthermore, an adaptive neural network (NN)-based finite-time admittance controller is designed to track the trajectory generated by the motion model and to reflect the learned stiffness characteristics. In this controller, radial basis function neural network (RBFNN) is used to compensate for the uncertainty of the robot dynamics. Finally, experiments are performed using the ROKAE collaborative robot, and the results confirm the effectiveness of the proposed approach. In summary, the proposed framework is suitable for human-robot skill transfer method that require simultaneous consideration of position and stiffness in Euclidean space and orientation on Riemannian manifolds.

1. Introduction

In recent years, robots are becoming ubiquitous with the development of robotics and artificial intelligence technologies [1]. There is an increasing requirement for robot adaptability due to the current market demand for customized services and rapid reorganization, always expecting them to operate as naturally and smoothly as humans. Therefore, it is necessary to develop methods to enhance the learning capabilities of robots [2]. Learning from demonstration (LfD) [3], a strategy for mapping from example states to actions has attracted significant attention. LfD has the advantages of simple and fast demonstration compared with traditional manual programming methods, which can significantly reduce labor and time costs and play a key role in improving the flexibility and intelligence of robots. On the other hand, LfD focuses mainly on point-to-point motion trajectory planning, but does not consider dynamical skills. Humans benefit from the inherent properties of the musculoskeletal system that naturally and intuitively balance the relationship between interactive force and position [4, 5]. Hogan [6] defined this dynamic relationship as an impedance/admittance model and applied it to robotics to achieve the compliance characteristics exhibited by humans. However, the traditional constant

* Corresponding author.
E-mail address: cyang@ieee.org (C. Yang).

impedance control is not in line with human operation habits, which cannot take into account the relationship between accuracy of motion and compliance of interaction, so it is an effective way to address this problems by equipping the robot with human-like variable impedance characteristics [7, 8].

The premise of LfD is skill modeling, where the extraction of human arm endpoint stiffness is the challenging task, and it is also the main factor affecting the performance of human interaction with the physical environment [9]. Abu-Dakka *et al.* [10] estimated the stiffness symmetric positive definition (SPD) matrix from the collected forces and the robot's state based on the impedance model and the sliding window method. However, the force signal usually lags behind the limb action, which affects the accuracy of the estimated stiffness. In contrast, the sEMG signal contains rich information and the signal is generated earlier than the limb action, which is more suitable as a decoding signal for arm stiffness information. Zeng *et al.* [11] collected sEMG signals through the MYO armband and performed endpoint stiffness estimation based on the conservative congruence transformation between the end-effector stiffness of the human arm and the joint stiffness. A simplified geometric model was proposed in [12] to reduce the complexity of the endpoint stiffness representation of the human arm, where the triangle formed by the human arm was modeled to determine the pose of the endpoint stiffness ellipsoid and the sEMG signal of the antagonist muscle to reflect the volume of the endpoint stiffness ellipsoid.

There are numerous types of research on robot skill learning, including Gaussian mixture model (GMM) [13], DMP [14], kernelized movement primitive (KMP) [15], gaussian process movement primitive (GP-MP) [16] and compliant movement primitive (CMP) [17]. DMP is constructed based on second-order spring-damped system, which has been widely noticed for its robustness to perturbations, convergence to attractors, and adaptability to new targets. Some researchers have applied DMP to trajectory tracking [18], obstacle avoidance [19], and state prediction [20], etc. However, traditional DMP is unsuitable for learning multi-demonstration data due to the particular structure of the model. Therefore, *Yang et al.* [21] modeled the nonlinear function of the DMP with GMM and its estimation was retrieved by Gaussian mixture regression (GMR). This modification allows the robot to extract more motion features from multi-demonstration and to generate motions that synthesize these features. On the other hand, traditional DMP also has a dependence on Euclidean spatial data, which leads to its inability to directly encode high-dimensional non-Euclidean skills, such as orientation and manipulability of the robot, which need to be represented on Riemannian manifolds. Therefore, in recent years, many studies [22, 23] have introduced Riemannian manifolds to properly represent Riemannian manifold skills and improve the learning accuracy.

Several types of manifolds including: the sphere manifold S^d, the special orthogonal group $SO(d)$ and the special Euclidean group $SE(3)$ are explored in [18], along with some basic operations such as *geodesics*, *logarithmic map* and *exponential map*. Ude *et al.* [24] proposed an improved DMP framework with the rotation matrix $SO(3)$ and the quaternion S^3 representation, and theoretical and experimental analysis showed that the *logarithmic map* defined on the unit quaternion space does not have discontinuity boundaries making it easier to use than the rotation matrix. Although these methods provide an effective way to learn orientation of robots, it would be better if the authors could simplify the representation by using Riemannian manifold tools and involve endpoint stiffness learning. To address this gap, Yu *et al.* [9] used DMP to learn translational motion and stiffness simultaneously to control the robot for pumping tasks. However, in real operation of the robot such as opening the door and other tasks, both translational and rotational motions are required to be considered, which limits the application of this method. Therefore, it is necessary to combine the Riemannian manifold to refine the formulation of the DMP and improve the learning accuracy, while considering the variable stiffness properties comprehensively.

In addition, the combination of DMP and robot control is another topic of great interest. Schaal *et al.* [25] proposed the motion control framework incorporating DMP. There is always uncertainty in the dynamics model of the robot leading

to poor tracking accuracy, while function approximation tools such as RBFNN are often used to deal with nonlinear properties [26] and have a faster convergence rate. On the other hand, many specific tasks nowadays put higher demands on the transient response and steady-state accuracy of control systems. However, most controllers can only ensure that the tracking error is asymptotically stable or consistently and eventually bounded stable, which means that as time approaches infinity, the error converges asymptotically to zero or to a neighborhood near zero. And that usually imposes some penalties on the transient performance aspects of the system, including a certain amount of overshoot, etc. The finite time control (FTC) technique [27, 28] has attracted wide attention due to its advantages of faster convergence, stronger resistance to disturbance and higher control accuracy.

Inspired by them, this research designs a robot skill learning system with multi-space fusion as shown in **Fig. 1**. It includes a skill generation model and an adaptive neural network-based finite-time admittance controller. The proposed framework has the following contributions compared to the DMP-based trajectory planning and various control methods:

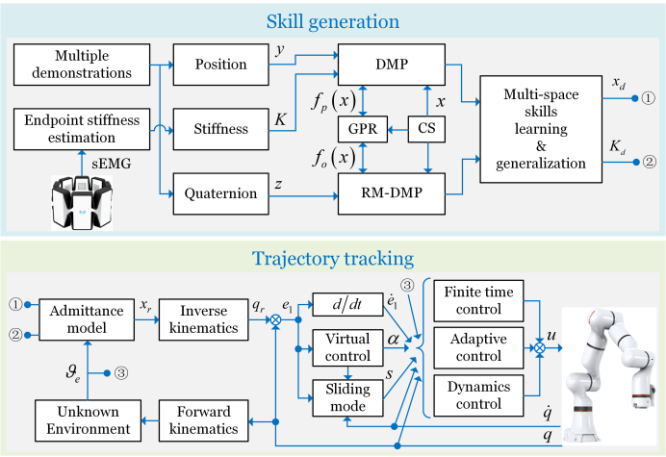


Fig. 1. Diagram of trajectory tracking controller and multi-space fusion robot skill learning and generalization.

1) A method to estimate endpoint stiffness based on sEMG signal is proposed, and the estimated stiffness is associated with variable admittance controller based on the robot dynamic model to maintain both compliance and safety in interaction tasks.

2) A novel robot skill learning framework is proposed. The nonlinear forcing term of the traditional DMP is approximated using GPR to model multi-demonstrations data. The DMP formulation is reconstructed based on the S^3 Riemannian metric (RM-DMP) to learn, reproduce and generalize the orientation properties of the robot represented by quaternions. Simultaneous encoding of position, orientation and stiffness to transfer human-like multi-space skills to the robot.

3) The FTC, sliding mode control (SMC) and Log-Barrier Lyapunov Function (Log-BLF) are integrated on the basis of the backstepping control technique to design a robot trajectory tracking controller with faster error convergence and higher steady-state accuracy.

The rest of this article is organized as follows: **Section 2** introduces the preliminaries. **Section 3** shows the robot skill learning method with multi-space fusion of this research. **Section 4** shows the adaptive neural network-based finite-time trajectory tracking control and **Section 5** gives the experimental results that are obtained by using the collaborative robot ER3 Pro developed by ROKAE. Finally, **Section 6** concludes the research. We summarize key notations used throughout this paper in Table 2.

2. Preliminaries

2.1. Dynamic movement primitive

The basic idea of DMP is to construct an "attractor model" using a second-order spring-damped system with self-stability, and to adjust the final state of the system by changing this "attractor" to converge to a specified motion [29]. It can be used to represent the evolution of various state variables in skill transfer, e.g., position [9] and stiffness [7, 21]. The DMP of 1-degree-of-freedom (DOF) discrete trajectory y is defined by the following nonlinear differential equation:

$$\tau^2\ddot{y}=\alpha_p\left(\beta_p\left(y_g-y\right)-\tau\dot{y}\right)+\left(y_g-y_0\right)f_p\left(x\right) \tag{1}$$

$$\tau\dot{x}=-\alpha_x x,\quad x\in[0,1],\quad x(0)=1 \tag{2}$$

where α_p and β_p are the designed parameters, τ is the time scaling parameter, y is the motion trajectory encoded

by DMP, y_0 and y_g are the initial and target positions, respectively. And both can be adjusted to achieve trajectory generalization according to the task requirements. The phase variable x is determined by the canonical system (CS) of Eq. (2), and $\alpha_x > 0$ is the decay rate. $f_p(x)$ is the nonlinear forcing term composed by N Gaussian basis functions.

$$f_p(x) = \frac{\sum_{i=1}^{N} \varphi_i(x)\omega_{pi}}{\sum_{i=1}^{N} \varphi_i(x)} x \tag{3}$$

$$\varphi_i(x) = \exp\left(-h_i(x-c_i)^2\right) = \exp\left(-\frac{1}{2\sigma_i^2}(x-c_i)^2\right) \tag{4}$$

where c_i and h_i are the center and width of the basis function $\varphi_i(x)$, respectively. ω_{pi} is the learning weight.

The transform system of Eq. (1) is used to represent the motion in each dimension for the learning of multi-dimensional motion trajectories, and the CS of Eq. (2) is used for synchronization.

2.2. S^3 Riemannian manifold

Using quaternions to represent the robot's orientation is a relatively common approach currently. However, the set S^3 composed of it does not belong to the vector space because it is not closed under the addition and scalar product, so it is not enough to treat and analyze it by the classical Euclidean space method. Combining quaternion with Riemannian metrics to form Riemannian manifolds is an effective way to solve the above problem. A Riemannian manifold M is a topological space whose local behavior is similar to that of the Euclidean space $\mathbb{R}^3$ as show in **Fig. 2**. M is a smooth and differentiable manifold with a positive definite metric tensor. There exists a locally linearized tangent space $\mathrm{T_pM}$ for each point $\mathrm{p} \in \mathrm{M}$, yields

$$\mathrm{p} = \lambda + \vec{\eta} = \lambda + \eta_x i + \eta_y j + \eta_z k \ : \ \mathrm{p} \in S^3 \tag{5}$$

S^3 is a unit sphere in $\mathbb{R}^4$. The quaternion conjugation of $\overline{\mathrm{p}}$ is defined as $\overline{\mathrm{p}} = \lambda - \eta_x i - \eta_y j - \eta_z k$.

A mapping system is needed to switch between $\mathrm{T_pM}$ and M to operate on the tangent space, which is defined by:

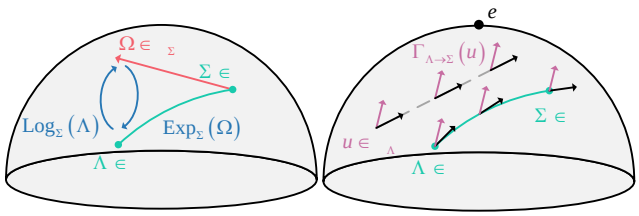


Fig. 2. Left: *exponential/logarithmic* mapping between the S^2 manifold embedded in $\mathbb{R}^3$ and the tangent space. Right: Parallel transport of a vector along a geodesic, e denotes the alignment base.

The *exponential map* $\mathrm{Exp}_\Sigma(\Omega) : \mathrm{T}_\Sigma\mathrm{M} \to \mathrm{M}$ maps a point $\Omega \in \mathrm{T}_\Sigma\mathrm{M}$ to a point $\Lambda \in \mathrm{M}$, so that it lies on the geodesic starting from $\Sigma \in S^3$ in the direction of Ω.

$$\Lambda = \mathrm{Exp}_\Sigma(\Omega) = \begin{cases} \left(\cos\left(\|\Omega\|\right), \sin\left(\|\Omega\|\right)\frac{\Omega}{\|\Omega\|}\right) * \Sigma, & \Omega \neq \vec{0} \\ 1 + \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\cdot} * \Sigma, & \text{otherwise} \end{cases} \tag{6}$$

where "$*$" is quaternion multiplication, it is defined by

$$\mathrm{p}_1 * \mathrm{p}_2 = \left(\lambda_1\lambda_2 - \eta_1^{\cdot}\eta_2\right) + \left(\lambda_1\eta_2 + \lambda_2\eta_1 + \eta_1 \times \eta_2\right) \tag{7}$$

The *logarithmic map* $\mathrm{Log}_\Sigma(\Lambda) : \mathrm{M} \to \mathrm{T}_\Sigma\mathrm{M}$ maps a point $\Sigma \in \mathrm{M}$ to a point $\Omega \in \mathrm{T}_\Sigma\mathrm{M}$.

$$\Omega = \mathrm{Log}_\Sigma(\Lambda) = \begin{cases} \arccos(q_\omega)\frac{\begin{bmatrix} q_x & q_y & q_z \end{bmatrix}}{\left\| q_x \ q_y \ q_z \right\|}, & \begin{bmatrix} q_x & q_y & q_z \end{bmatrix} \neq \vec{0} \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\cdot}, & \text{otherwise} \end{cases} \tag{8}$$

where $(\Lambda) * \bar{\Sigma} = \left(q_\omega, q_x i, q_y j, q_z k\right) \in S^3$.

Another useful operation on manifolds is the *parallel transport* $\Gamma_{\Sigma\to\Lambda}(\Omega) : \mathrm{T}_\Sigma\mathrm{M} \to \mathrm{T}_\Lambda\mathrm{M}$, which moves elements between tangent spaces as a function from Σ to Λ on the geodesic such that the angle between two elements in the tangent space remains constant, thus making the inner product conserved [22, 30]. It is given by

$$\Gamma_{\Sigma\to\Lambda}(\Omega) = B^{\cdot} R_{\Sigma\to e} R_{\Sigma\to\Lambda} R_{e\to\Lambda} B\Omega \tag{9}$$

where B contains the direction of the base coordinates at the origin, which in the S^3 manifold is given by

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{\cdot} \tag{10}$$

$$R_{\Sigma\to\Lambda} = I_4 - \sin(m)\Sigma u^{\text{唯}} + \left(\cos(m) - 1\right)uu \tag{11}$$

where $u = \left[\kappa^{\cdot}, 0\right]^{\cdot}$ is the direction of transport,

$\kappa = R_{e\to\Sigma}\mathrm{Log}_{\Sigma}(\Lambda)/m$ with $m = \mathrm{Log}_{\Sigma}(\Lambda)$ is the angle of transport.

2.3. Gaussian process regression

Gaussian process regression (GPR) is a nonparametric regression method based on Bayesian inference [31], which predicts output variables by modeling the Gaussian process on the input data. Given a set of noise observations $D = \left\{ \left(x_i^{\mathrm{I}}, x_i^{\mathrm{O}} \right) \mid i = 1, \cdots, M \right\}$ in which $x_i^{\mathrm{O}} = G\left(x_i^{\mathrm{I}} \right) + \varepsilon$, $\varepsilon \sim N\left(0, \sigma_n^2\right)$, GPR is implemented as:

$$G(x) \sim GP\left(\mu\left(x^{\mathrm{I}}\right), K\left(x^{\mathrm{I}}, x^{\mathrm{I}}\right) \right) \tag{12}$$

where $\mu\left(x^{\mathrm{I}}\right) = E\left[G(x)\right]$ is mean function, $K\left(x^{\mathrm{I}}, x^{\mathrm{I}}\right)$ is the covariance function of the process

$$K\left(x^{\mathrm{I}}, x^{\mathrm{I}}\right) = \begin{bmatrix} k\left(x_1^{\mathrm{I}}, x_1^{\mathrm{I}}\right) & k\left(x_1^{\mathrm{I}}, x_2^{\mathrm{I}}\right) & \cdots & k\left(x_1^{\mathrm{I}}, x_M^{\mathrm{I}}\right) \\ k\left(x_2^{\mathrm{I}}, x_1^{\mathrm{I}}\right) & k\left(x_2^{\mathrm{I}}, x_2^{\mathrm{I}}\right) & \cdots & k\left(x_2^{\mathrm{I}}, x_M^{\mathrm{I}}\right) \\ \vdots & \vdots & \ddots & \vdots \\ k\left(x_M^{\mathrm{I}}, x_1^{\mathrm{I}}\right) & k\left(x_M^{\mathrm{I}}, x_2^{\mathrm{I}}\right) & \cdots & k\left(x_M^{\mathrm{I}}, x_M^{\mathrm{I}}\right) \end{bmatrix} \tag{13}$$

where $k\left(x_i^{\mathrm{I}}, x_j^{\mathrm{I}}\right) = E\left[G\left(x_i^{\mathrm{I}} - \mu\left(x_i^{\mathrm{I}}\right)\right) G\left(x_j^{\mathrm{I}} - \mu\left(x_j^{\mathrm{I}}\right)\right) \right]$ and it is usually defined as $k\left(x_i^{\mathrm{I}}, x_j^{\mathrm{I}}\right) = \sigma_f^2 \exp\left(-\frac{1}{2L} \left\| x_i^{\mathrm{I}} - x_j^{\mathrm{I}} \right\|^2 \right)$, L is diagonal matrix defined by the length scale and σ_f^2 is the variance.

Specify a new set of inputs $x^{\mathrm{I}*}$, and the joint distribution of the outputs is given:

$$\begin{bmatrix} x^{\mathrm{O}} \\ x^{\mathrm{O}*} \end{bmatrix} \sim N\left(\begin{bmatrix} \mu\left(x^{\mathrm{I}}\right) \\ \mu\left(x^{\mathrm{I}*}\right) \end{bmatrix}, \begin{bmatrix} K\left(x^{\mathrm{I}}, x^{\mathrm{I}}\right) + \sigma_n^2 I & K\left(x^{\mathrm{I}}, x^{\mathrm{I}*}\right) \\ K\left(x^{\mathrm{I}*}, x^{\mathrm{I}}\right) & K\left(x^{\mathrm{I}*}, x^{\mathrm{I}*}\right) \end{bmatrix} \right) \tag{14}$$

Then, the conditional probability property of the Gaussian distribution is used to evaluate the posterior distribution over $x^{\mathrm{O}*}$, yielding a Gaussian

$$p\left(x^{\mathrm{O}*} \mid x^{\mathrm{O}}\right) \sim N\left(\mu^*, \Sigma^*\right) \tag{15}$$

with mean and covariance

$$\mu^* = \mu\left(x^{\mathrm{I}*}\right) + K\left(x^{\mathrm{I}*}, x^{\mathrm{I}}\right)\left(K\left(x^{\mathrm{I}}, x^{\mathrm{I}}\right) + \sigma_n^2 I\right)^{-1}\left(x^{\mathrm{O}} - \mu\left(x^{\mathrm{I}}\right)\right) \tag{16}$$

$$\Sigma^* = K\left(x^{\mathrm{I}*}, x^{\mathrm{I}*}\right) - K\left(x^{\mathrm{I}*}, x^{\mathrm{I}}\right)\left(K\left(x^{\mathrm{I}}, x^{\mathrm{I}}\right) + \sigma_n^2 I\right)^{-1} K\left(x^{\mathrm{I}}, x^{\mathrm{I}*}\right) \tag{17}$$

In practice it is often assumed that $\begin{bmatrix} \mu\left(x^{\mathrm{I}}\right) \\ \mu\left(x^{\mathrm{I}*}\right) \end{bmatrix} = 0$. Gaussian processes can thus be completely defined by their second order statistics. Different mathematical tools can be used to choose the appropriate values of the GPR parameters $\left(\sigma_n, \sigma_f, L\right)$, such as maximizing the marginal likelihood and leave-one-out cross-validation [32].

3. Robot skill learning method with multi-space fusion

3.1. sEMG-based stiffness estimation

It is effective to modulate the stiffness properties of the robot in the workspace by estimating the endpoint stiffness of the human tutor's arm. In this research, the sEMG signals are collected from the MYO armband and the envelope is extracted from the raw sEMG signals by the following moving average technique.

$$l\left(A_t\right) = \frac{1}{W}\sum_{i=0}^{W-1} \mathrm{EMG}\left(A_{t-i}\right) \tag{18}$$

where $l\left(A_t\right)$ is the amplitude of the envelope, W is the length of the window function, $\mathrm{EMG}\left(A_t\right)$ is the amplitude of raw sEMG. **Fig. 3** gives an example of extracting envelope.

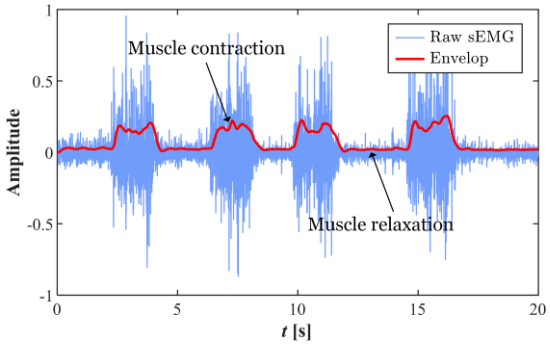


Fig. 3. The envelope is extracted from the raw sEMG signal.

Further, it is defined that

$$\kappa = \frac{1}{D}\sum_{i=1}^{D} \left| l\left(A_t\right) \right| \tag{19}$$

$$c_\kappa = \frac{1 - e^{-\eta\bar{\kappa}}}{1 + e^{-\eta\bar{\kappa}}} \quad \text{with} \quad 0 \le \bar{\kappa} = \frac{\kappa}{\kappa_{\max}} \le 1 \tag{20}$$

where $D = 8$ is the total channels number of the MYO, and

κ is the mean of the amplitude. c_κ is the stiffness index, and η is the designed parameter. Thus, the mapping between the c_κ and the estimated endpoint stiffness k is given by:

$$k = c_\kappa \left(k_{\max} - k_{\min}\right) + k_{\min} \tag{21}$$

$$K = S_k k \tag{22}$$

where the range of stiffness is determined by $k_{\max}$ and $k_{\min}$, S_k is the diagonal matrix that is used to select the axes in which the stiffness should be modulated (units of the first three elements are N/m, while the last three are Nm/rad).

3.2. Multi-space dynamic movement primitive

Given multi-demonstrations $Y = \left\{\ddot{y}_{t,n}, \dot{y}_{t,n}, y_{t,n}\right\}$ with $t = 1, 2, \cdots, T;\ n = 1, 2, \cdots, N_d$ for specific tasks. GPR is used to approximate the nonlinear forcing term $f_p(x)$ of the DMP. Simultaneously, alignment of motion data from multi-demonstrations is usually required. The generalized time warping (GTW) [33] algorithm is used to deal with this problem in this article. GTW minimizes the following objective function:

$$J_{GTW} = \sum_{i=1}^{N_d}\sum_{j=1}^{N_d}\frac{1}{2}\left\| F_i^{\text{嘶}} Y_i W_i - F_j\ Y_j W_j \right\|^2 + \sum_{i=1}^{N_d}\left(\phi(F_i) + \varphi(W_i)\right) \tag{23}$$

where F and W are a nonlinear time-transformed embedding matrix and a low-dimensional spatial embedding matrix, respectively. $\phi(\cdot)$ and $\varphi(\cdot)$ are regularization functions. Eq. (23) shows that GTW will ensure that $F_i^{\bullet}\ Y_i W$ is aligned with the other vectors in a least-squares way.

Further, the traditional DMP formulation is reconstructed based on the S^3 Riemannian manifold for encoding robot orientation. By defining $\{t_i, z_i\}_{i=1}$ as a set of robot's orientation evolution demonstrations. Its 1st- time derivative is calculated by

$$\dot{z}_i = \Gamma_{z_i \to z_0}\left(\frac{\mathrm{Log}_{z_{i-1}}(z_i)}{dt}\right) \tag{24}$$

The 2nd- time derivative is calculated directly using standard Euclidean tools. Thus, a set of orientation learning database is constructed as $Z_i = \{t_i, z_i, \dot{z}_i, \ddot{z}_i\}_{i=1}$.

The S^3 Riemannian metric-based DMP (RM-DMP) can be rewritten as

$$\tau^2 \ddot{z} = \alpha_o\left(\beta_o \Gamma_{z \to z_0}\left(\mathrm{Log}_z(z_g)\right) - \tau \dot{z}\right) + \mathrm{Log}_{z_0}(z_g)\, f_o(x) \tag{25}$$

where $f_o(x)$ is also modeled by GPR. z_0 and z_g can be modified to generalize based on task requirements.

In the reproduction phase, the following equation is used to obtain the imitated orientation.

$$z(t + \delta t) = \mathrm{Exp}_{z(t)}\left(\frac{\Gamma_{z_0 \to z(t)}\left(\dot{z}(t)\right)}{\tau}\delta t\right) \tag{26}$$

Therefore, the traditional DMP, described in Eq. (1), is used to encode the robot's position and stiffness. The RM-DMP, described in Eq. (25), is used to encode the robot's orientation. And the synchronization is guaranteed by the same canonical system of Eq. (2).

3.3. Admittance control method

Robots must adapt to a variety of unknown physical environments to be a reliable partner for human, i.e., have superior interaction capabilities. Admittance control has been proven by many studies [34] to be an effective method to ensure stability and safety when robots interact with unknown environments. In this research, the following variable admittance controller is designed.

$$\begin{bmatrix} F_e \\ \tau_e \end{bmatrix} = \begin{bmatrix} K_{pd}(y_r - y_d) + B_{pd}\dot{y}_r \\ K_{od}\mathrm{Log}_{z_d}(z_r) + B_{od}\omega_r \end{bmatrix} \text{ with } \begin{bmatrix} B_{pd} \\ B_{od} \end{bmatrix} = \zeta\sqrt{\begin{bmatrix} K_{pd} \\ K_{od} \end{bmatrix}} \tag{27}$$

where F_e/τ_e is the interaction force/torque between the robot and the external environment, which is obtained by the 6-Axis sensor. K_{pd} and B_{pd} represent the translational stiffness matrix and damping matrix of the robot in the workspace, respectively. K_{od} and B_{od} represent the rotational stiffness matrix and damping matrix, respectively. $K_{(\cdot)d} \in \square^{3\times3}$ is diagonal matrix whose components are learned/generalized by the DMP. ζ is designed positive constant coefficient. $y_d \in \square^3$ and $z_d \in S^3$ are the desired position and orientation, learned/generalized by DMP and RM-DMP, respectively. $y_r \in \square^3$ and $z_r \in S^3$ are the command position and

orientation, respectively. $\omega_c \in \mathbb{R}^3$ is the angular velocity, and the mapping between ω_r and 1st- time derivative of the z_r is given by

$$\dot{z}_r = \frac{1}{2}\left[0, \omega_r^{\cdot}\right]^{\cdot} * z_r \tag{28}$$

In summary, $\begin{bmatrix} y_r \\ z_r \end{bmatrix}$ is derived by substituting $\begin{bmatrix} y_d \\ z_d \end{bmatrix}$, $\begin{bmatrix} K_{pd} \\ K_{od} \end{bmatrix}$ and $\begin{bmatrix} F_e \\ \tau_e \end{bmatrix}$ into the admittance model, which is used as the Cartesian input for the subsequent adaptive neural network (NN)-based robot trajectory tracking controller.

4. Adaptive neural network-based finite-time trajectory tracking control

In this section, a novel adaptive neural network (NN)-based finite-time control method of robot is proposed to track the trajectory learned by the multi-space DMP by considering the error convergence rate and the steady-state tracking accuracy.

4.1. Problem description

The dynamics model of the robot can be expressed as:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) + \vartheta_e(t) = u \tag{29}$$

where $\vartheta_e = J^{\cdot}\begin{bmatrix} F_e \\ \tau_e \end{bmatrix}$, which is the force/torque applied to the robot in the workspace. $M \in \mathbb{R}^{n\times n}$, $C \in \mathbb{R}^{n\times n}$ and $G \in \mathbb{R}^{n\times 1}$ are the inertia matrix, the Centrifugal force and Coriolis force matrix and gravity matrix, respectively. $q \in \mathbb{R}^{n\times 1}$, $\dot{q} \in \mathbb{R}^{n\times 1}$ and $\ddot{q} \in \mathbb{R}^{n\times 1}$ are the joint angle, joint velocity and joint acceleration, respectively. $J \in \mathbb{R}^{m\times n}$ is the Jacobian matrix, m is the dimension of the workspace and n is the DOF of the robot. $u \in \mathbb{R}^{n\times 1}$ is the control input.

Generally, M, C and G have uncertain nonlinear terms, i.e. $M = M_0 + \Delta M$, $C = C_0 + \Delta C$ and $G = G_0 + \Delta G$. M_0, C_0 and G_0 represent the nominal parts, ΔM, ΔC and ΔG represent the other unknown parts. Thus, Eq. (29) can be rewritten as:

$$M_0\ddot{q} + C_0\dot{q} + G_0 + \vartheta_e + \Delta = u \tag{30}$$

where $\Delta = \Delta M\ddot{q} + \Delta C\dot{q} + \Delta G$. It is defined that $x_1 = q = \left[x_{11}, \cdots, x_{1n}\right]^{\cdot}$, $x_2 = \dot{q} = \left[x_{21}, \cdots, x_{2n}\right]^{\cdot}$, Eq. (30) can be converted as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = M_0^{-1}\left(u - C_0x_2 - G_0 - \vartheta_e - \Delta\right) \end{cases} \tag{31}$$

The error variable is defined as:

$$\begin{cases} e_1 = x_1 - q_r \\ e_2 = x_2 - \alpha \end{cases} \tag{32}$$

where q_r is command joint position, which is obtained by performing the inverse kinematic solution for $x_r = \begin{bmatrix} y_r \\ z_r \end{bmatrix}$. α is the virtual control law.

The Lyapunov function is defined as:

$$V_1 = \frac{1}{2}\sum_{i=1}^{n} \ln \frac{b_i^2 k_{ai}^2(t)}{b_i^2 k_{ai}^2(t) - e_{1i}^2} \tag{33}$$

where $b_i = \frac{\chi(e_{1i})}{\mu_{i1}} + \frac{1-\chi(e_{1i})}{\mu_{i2}}$, with $\chi(e_{1i}) = \begin{cases} 1 & e_{1i} \geq 0 \\ 0 & e_{1i} < 0 \end{cases}$, $k_{ai} = \left(k_0 - k_\infty\right)e_{1i}^{-at} + k_\infty$ denotes the prescribed performance function that determines the error convergence bound.

Derivation of V_1 yields:

$$\begin{aligned} \dot{V}_1 &= \sum_{i=1}^{n} \frac{\left(b_i k_{ai}\right)^2 e_{1i}\dot{e}_{1i} - e_{1i}^2 b_i^2 k_{ai}\dot{k}_{ai}}{\left(b_i k_{ai}\right)^2\left(\left(b_i k_{ai}\right)^2 - e_{1i}^2\right)} \\ &= \sum_{i=1}^{n} \frac{e_{1i}}{\left(b_i k_{ai}\right)^2 - e_{1i}^2}\left(\dot{e}_{1i} - \frac{\dot{k}_{ai}}{k_{ai}}e_{1i}\right) \\ &= \sum_{i=1}^{n} \frac{e_{1i}}{\left(b_i k_{ai}\right)^2 - e_{1i}^2}\left(x_{2i} - \dot{q}_{ri} - \frac{\dot{k}_{ai}}{k_{ai}}e_{1i}\right) \end{aligned} \tag{34}$$

The virtual control law α is designed as

$$\alpha = -K_1e_1 - H_1K_{v1} + \dot{q}_r + \frac{\dot{K}_a}{K_a}e_1 \tag{35}$$

where $K_{v1} = \left[\frac{e_{11}^{2\gamma-1}}{\left(\left(b_i k_{ai}\right)^2 - e_{11}^2\right)^{\gamma-1}}, \cdots, \frac{e_{1n}^{2\gamma-1}}{\left(\left(b_n k_{an}\right)^2 - e_{1n}^2\right)^{\gamma-1}}\right]^{\cdot}$,

$K_a = \left[k_{a1}, \cdots, k_{an}\right]^{\cdot}$, $K_1 = diag\left[k_{1i}, \cdots, k_{1n}\right]$, $H_1 = diag\left[h_{11}, \cdots, h_{1n}\right]$.

Substituting Eq. (35) into Eq. (34).

$$\dot{V}_1 = \sum_{i=1}^{n} \frac{e_{1i}}{\left(b_i k_{ai}\right)^2 - e_{1i}^2}\left(e_{2i} - k_{1i}e_{1i} - h_{1i}\frac{e_{1i}^{2\gamma-1}}{(k_{ai}^2 - e_{1i}^2)^{\gamma-1}}\right) \tag{36}$$

Further, the sliding mode surface is defined as:

$$s = ce_1 + e_2 \tag{37}$$

It is defined that

$$V_2 = V_1 + \frac{1}{2} s^{\cdot} s \tag{38}$$

Derivation of V_2 yields:

$$\begin{aligned}\dot{V}_2 &= \sum_{i=1}^{n} \frac{e_{1i}}{\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}} \left(s_i - c e_{1i} - k_{1i} e_{1i} - h_{1i} \frac{e_{1i}^{\ 2\gamma-1}}{(k_{ai}^{\ 2} - e_{1i}^{\ 2})^{\gamma-1}} \right) \\ &\quad + s^{\cdot} \left(c\dot{e}_1 + \dot{x}_2 - \dot{\alpha} \right) \\ &= -\sum_{i=1}^{n} (c + k_{1i}) \frac{e_{1i}^{\ 2}}{\left(\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}\right)^2} - \sum_{i=1}^{n} h_{1i} \left(\frac{e_{1i}^{\ 2}}{\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}} \right)^{\gamma} \\ &\quad + s^{\cdot} \left(K_{v2} + c\dot{e}_1 + \dot{x}_2 - \dot{\alpha} \right) \\ &= -\sum_{i=1}^{n} (c + k_{1i}) \frac{e_{1i}^{\ 2}}{\left(\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}\right)^2} - \sum_{i=1}^{n} h_{1i} \left(\frac{e_{1i}^{\ 2}}{\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}} \right)^{\gamma} \\ &\quad + s^{\cdot} \left(K_{v2} + c\dot{e}_1 + M_0^{\ -1} \left(u - C_0 x_2 - G_0 - \tau_e \right) - f - \dot{\alpha} \right)\end{aligned} \tag{39}$$

where $K_{v2} = \left[\frac{e_{11}}{k_{a1}^{\ 2} - e_{11}^{\ 2}}, \cdots, \frac{e_{1n}}{k_{an}^{\ 2} - e_{1n}^{\ 2}} \right]^{\cdot}$, $f = M_0^{\ -1} \Delta$ is the unknown nonlinear term, which is approximated using RBFNN $f = W^{*\cdot} \theta + \varepsilon$, where W^* is the ideal neural network weight, θ is the radial basis function, and ε is the approximation error vector with $|\varepsilon_i| \le \bar{\varepsilon}_i$, $\bar{\varepsilon}_i > 0$ is the upper bound of the error..

The control input u is designed as:

$$\begin{aligned}u &= M_0 \left[\dot{\alpha} - K_{v2} - c\dot{e}_1 - H_2 K_s - K_2 s - \beta \operatorname{sign}(s) \right] \\ &\quad + C_0 x_2 + G_0 + \vartheta_e + \hat{W}^{\cdot} \theta\end{aligned} \tag{40}$$

where $H_2 = diag\left[h_{21}, \cdots, h_{2n}\right]$, $K_s = \left[s_1^{\ 2\gamma-1}, \cdots, s_n^{\ 2\gamma-1} \right]^{\cdot}$, $K_2 = diag[k_{2i}, \cdots, k_{2n}]$, $\beta \ge \bar{\varepsilon}$.

Substituting Eq. (40) into Eq. (39).

$$\begin{aligned}\dot{V}_2 &= -\sum_{i=1}^{n} (c + k_{1i}) \frac{e_{1i}^{\ 2}}{\left(\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}\right)^2} - \sum_{i=1}^{n} h_{1i} \left(\frac{e_{1i}^{\ 2}}{\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}} \right)^{\gamma} \\ &\quad + s^{\mathrm{T}} \left[-K_2 s - \beta \operatorname{sign}(s) - H_2 K_s + \tilde{W}^{\mathrm{T}} \theta - \varepsilon \right] \\ &= -\sum_{i=1}^{n} (c + k_{1i}) \frac{e_{1i}^{\ 2}}{\left(\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}\right)^2} - \sum_{i=1}^{n} h_{1i} \left(\frac{e_{1i}^{\ 2}}{\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}} \right)^{\gamma} \\ &\quad - \sum_{i=1}^{n} k_{2i} s_i^{\ 2} - \sum_{i=1}^{n} h_{2i} (s^2)^{\gamma} + \sum_{i=1}^{n} \tilde{W}_i^{\cdot} \theta_i s_i - \sum_{i=1}^{n} s_i \left(\varepsilon_i + \beta_i \operatorname{sign}(s_i) \right)\end{aligned} \tag{41}$$

4.2. Stability analysis

Lemma 1. ***(Young's Inequality).*** For $\forall (x, y) \in \mathbb{R}^2$, such that:

$$xy \le \frac{\lambda^p}{p} |x|^p + \frac{1}{q \lambda^q} |y|^q \tag{42}$$

where $\lambda > 0$, $p > 1$, $q > 1$ and $(p-1)(q-1) = 1$.

Lemma 2. For any variables Υ and ν, and given constants d_1, d_2 and d_3, the following inequality holds:

$$|\Upsilon|^{d_2} |\nu|^{d_1} \le \frac{d_2}{d_1 + d_2} d_3^{-\frac{d_1}{d_2}} |\Upsilon|^{d_1 + d_2} + \frac{d_1}{d_1 + d_2} d_3 |\nu|^{d_1 + d_2} \tag{43}$$

Lemma 3. For any positive constant $k \in R^n$, the following inequality holds for any $e \in R^n$ in the interval $|e| < |k|$.

$$\log \frac{k^{\mathrm{T}} k}{k^{\mathrm{T}} k - e^{\mathrm{T}} e} \le \frac{e^{\mathrm{T}} e}{k^{\mathrm{T}} k - e^{\mathrm{T}} e} \tag{44}$$

The Lyapunov function is selected as follow:

$$V = V_2 + \frac{1}{2} \sum_{i=1}^{n} \tilde{W}_i^{\cdot} r_i^{-1} \tilde{W} \tag{45}$$

where $r = diag\left[r_1, \cdots, r_n\right]$ is a predetermined positive definite diagonal matrix.

Derivation of V yields:

$$\begin{aligned}\dot{V} &= -\sum_{i=1}^{n} (c + k_{1i}) \frac{e_{1i}^{\ 2}}{\left(\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}\right)^2} - \sum_{i=1}^{n} h_{1i} \left(\frac{e_{1i}^{\ 2}}{\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}} \right)^{\gamma} \\ &\quad - \sum_{i=1}^{n} k_{2i} s_i^{\ 2} - \sum_{i=1}^{n} h_{2i} (s^2)^{\gamma} + \sum_{i=1}^{n} \frac{\tilde{W}_i^{\cdot}}{r_i} (r_i \theta_i s_i + \dot{\tilde{W}}_i) \\ &\quad - \sum_{i=1}^{n} s_i \left(\varepsilon_i + \beta_i \operatorname{sign}(s_i) \right)\end{aligned} \tag{46}$$

The adaptive law of $\hat{W}$ is defined as:

$$\dot{\hat{W}}_i = -r_i (\theta_i s_i + \sigma \hat{W}_i) \tag{47}$$

where σ is positive constant. Thus, Eq. (46) can be rewritten:

$$\begin{aligned}\dot{V} &= -\sum_{i=1}^{n} (c + k_{1i}) \frac{e_{1i}^{\ 2}}{\left(\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}\right)^2} - \sum_{i=1}^{n} h_{1i} \left(\frac{e_{1i}^{\ 2}}{\left(b_i k_{ai}\right)^2 - e_{1i}^{\ 2}} \right)^{\gamma} \\ &\quad - \sum_{i=1}^{n} k_{2i} s_i^{\ 2} - \sum_{i=1}^{n} k_{2i} s_i \beta_i sats(s_i) - \sum_{i=1}^{n} h_{2i} (s^2)^{\gamma} \\ &\quad - \sum_{i=1}^{n} s_i \left(\varepsilon_i + \beta_i \operatorname{sign}(s_i) \right) - \sum_{i=1}^{n} \sigma_i \tilde{W}_i^{\cdot} \hat{W}_i\end{aligned} \tag{48}$$

According to **Lemma 1.**

$$-\sum_{i=1}^{n} \sigma_i \tilde{W}_i^{\mathrm{T}} \hat{W}_i \le -\sum_{i=1}^{n} \frac{\sigma_i}{2} \tilde{W}_i^{\mathrm{T}} \tilde{W}_i + \sum_{i=1}^{n} \frac{\sigma_i}{2} \left\| W_i^* \right\|^2 \tag{49}$$

According to **Lemma 2**, it is defined $v=1$, $d_1=1-\gamma$, $d_2=\gamma$, $d_3=\gamma^{\gamma/(1-\gamma)}$, $\Upsilon=\sum_{i=1}^{n}\frac{\sigma_i}{4}\tilde{W}_i^{\mathrm{T}}\tilde{W}$.

$$-\sum_{i=1}^{n}\frac{\sigma_i}{4}\tilde{W}_i^{\mathrm{T}}\tilde{W}_i\le(1-\gamma)\gamma^{\gamma/(1-\gamma)}-\left(\sum_{i=1}^{n}\frac{\sigma_i}{4}\tilde{W}_i^{\mathrm{T}}\tilde{W}_i\right)^{\gamma} \tag{50}$$

Additionally, it should be noted that when $s_i<0$, $-\sum_{i=1}^{n}s_i\left(\varepsilon_i+\beta_i\mathrm{sign}(s_i)\right)=-\sum_{i=1}^{n}s_i\left(\varepsilon_i-\beta_i\right)<0$, and when $s_i>0$, $-\sum_{i=1}^{n}s_i\left(\varepsilon_i+\beta_i\mathrm{sign}(s_i)\right)=-\sum_{i=1}^{n}s_i\left(\varepsilon_i+\beta_i\right)<0$.

Combined with **Lemma 3,** Eq. (48) can be rewritten as

$$\begin{aligned}\dot{V}\le&-\sum_{i=1}^{n}(c+k_{1i})\log\frac{\left(b_ik_{ai}\right)^2}{\left(\left(b_ik_{ai}\right)^2-e_{1i}^{\ 2}\right)^2}\\&-\sum_{i=1}^{n}h_{1i}\left(\log\frac{\left(b_ik_{ai}\right)^2}{\left(b_ik_{ai}\right)^2-e_{1i}^{\ 2}}\right)^{\gamma}-\sum_{i=1}^{n}k_{2i}s_i^{\ 2}\\&-\sum_{i=1}^{n}h_{2i}(s_i^{\ 2})^{\gamma}-\sum_{i=1}^{n}\frac{\sigma_i}{2}\tilde{W}_i^{\mathrm{T}}\tilde{W}_i+\sum_{i=1}^{n}\frac{\sigma_i}{2}\left\|W_i^*\right\|^2\end{aligned} \tag{51}$$

Further

$$\begin{aligned}\dot{V}\le&-\sum_{i=1}^{n}(c+k_{1i})\log\frac{\left(b_ik_{ai}\right)^2}{\left(\left(b_ik_{ai}\right)^2-e_{1i}^{\ 2}\right)^2}-\sum_{i=1}^{n}k_{2i}s_i^{\ 2}\\&-\sum_{i=1}^{n}\frac{\sigma_i}{4}\tilde{W}_i^{\mathrm{T}}\tilde{W}-\sum_{i=1}^{n}h_{1i}\left(\log\frac{\left(b_ik_{ai}\right)^2}{\left(b_ik_{ai}\right)^2-e_{1i}^{\ 2}}\right)^{\gamma}\\&-\sum_{i=1}^{n}h_{2i}(s_i^{\ 2})^{\gamma}-\left(\sum_{i=1}^{n}\frac{\sigma_i}{4}\tilde{W}_i^{\mathrm{T}}\tilde{W}_i\right)^{\gamma}\\&+(1-\gamma)\gamma^{\gamma(1-\gamma)}+\sum_{i=1}^{n}\frac{\sigma_i}{2}\left\|W_i^*\right\|^2\\\le&-\mu_1V-\mu_2V^{\gamma}+\rho\end{aligned} \tag{52}$$

where

$$\mu_1=\min\left\{\begin{array}{c}\min\limits_{i=1,\cdots,n}\left(\sigma_i/2\lambda_{\max}\left(r_i^{-1}\right)\right)\\\min\limits_{i=1,\cdots,n}\left[2\left(c+k_{1i}\right)\right],\min\limits_{i=1,\cdots,n}\left(2k_{2i}-1\right)\end{array}\right. \tag{53}$$

$$\mu_2=\min\left\{\min_{i=1,\cdots,n}2^{\gamma}k_{1i},\min_{i=1,\cdots,n}2^{\gamma}k_{2i},\min(\sigma_i/2\lambda_{\max}(r_i^{-1}))^{\gamma}\right\} \tag{54}$$

$$\rho=(1-\gamma)\gamma^{\gamma(1-\gamma)}+\sum_{i=1}^{n}\frac{\sigma_i}{2}\left\|W_i^*\right\|^2+\frac{1}{2}\sum_{i=1}^{n}\delta_i^{\ 2} \tag{55}$$

It should be ensured that $\min\limits_{i=1,\cdots,n}\left(\sigma_i/2\lambda_{\max}\left(r_i^{-1}\right)\right)>0$, $\min\limits_{i=1,\cdots,n}\left[2\left(c+k_{1i}\right)\right]>0$ and $\min\limits_{i=1,\cdots,n}\left(2k_{2i}-1\right)>0$ to make $\mu_1>0$ and $\mu_2>0$. According to [35], there always exists a finite time t_0 such that $V\ge\left(2\rho/\mu_2\right)^{(1/\gamma)}$ holds for all $t\in[0,t_0]$. Therefore, $V\le-\mu_1V-\left(\mu_2/2\right)V^{\gamma}$ holds and the closed-loop system can be rapidly stabilized in finite time $T_0\le\frac{1}{\mu_1(1-\gamma)}\mathrm{In}\left[\left(2\mu_1V^{1-\gamma}(0)+\mu_2\right)/\mu_2\right]$ for all $t\in[0,t_0]$. Furthermore, it can be shown that $t_0\le T_0$. $V\le\left(2\rho/\mu_2\right)^{1/\gamma}$ holds for $\forall t>T_0$. In finite time T_0, the internal signal e_{1i} will converge to the following compact set.

$$|e_{1i}|\le k_{ai}\left[1-e^{-2(2\rho/\mu_2)^{1/\gamma}}\right]^{\frac{1}{2}} \tag{56}$$

According to [35], if $|e_{1i}(0)|<k_{ai}$, $|e_{1i}|<k_{ai}$ for $\forall t>0$. Therefore, the steady-state trajectory tracking error e_1 of the robot system will converge to within the range of the prescribed performance function k_{ai} while ensuring the convergence speed.

5. Experiment

The effectiveness of the proposed method is verified on the collaborative robot ER3 Pro developed by ROKAE, and the experimental scenario is set to simulate the robot learning/generalizing water pouring skill. The hardware architecture of the control system is shown in **Fig. 4**. The host computer is constructed on the PREEMPT-Linux system to ensure real-time performance. Meanwhile, the human tutor wears a MYO armband on the arm to collect sEMG signals. The collected sEMG signals are sent to the host computer for processing via Bluetooth. The 6-Axis force sensor provides real-time force/torque feedback via the Ethernet protocol with the control cycle set to 200 (HZ). The electric gripper at the end of the robot is used to hold the water bottle. The built-in torque control mode allows us to drag the robot directly for demonstrations, and the estimated stiffness and robot's states are recorded for analysis of the experimental results.

In addition, the acquisition process of actual force/torque is shown in **Fig. 5**. Where $\{B\}$ is the base coordinate system, $\{S\}$ is the sensor coordinate system and $\{T\}$ is the tool coordinate system. Gravity/gravitational torque compensation

of the tool based on the raw data collected by the 6-axis force sensor is performed to obtain corrected data, which is multiplied with the rotation matrix ${}^{T}_{B}R\,{}^{B}_{S}R$ to align $\{T\}$. It is worth noting that all variables of the proposed control framework are considered under the coordinate system $\{T\}$.

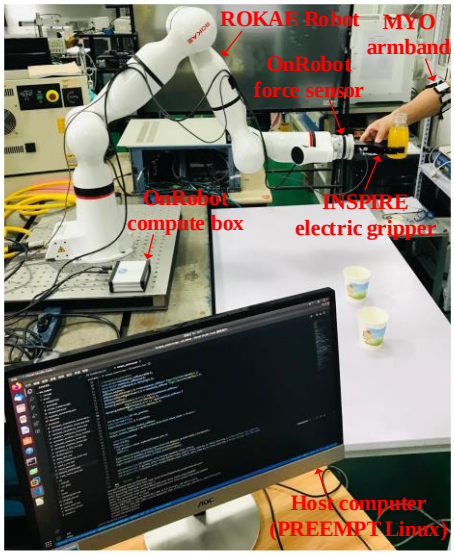


Fig. 4. The hardware architecture of the control system.

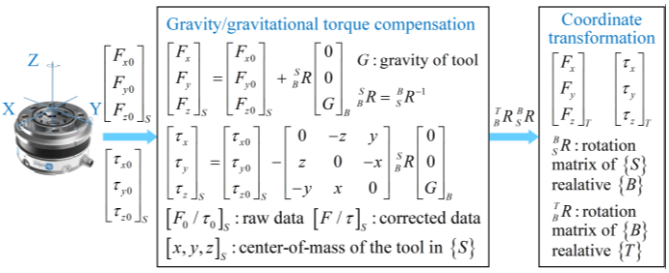


Fig. 5. The acquisition process of actual force/torque.

5.1. Test of the adaptive NN-based finite-time controller

In this experiment, the proposed adaptive NN-based finite-time controller was compared with the traditional PD controller, and the desired trajectory was defined as a circumferential motion with $p_x = 0.415 + 0.05 \times \sin(2\pi t/10)$, $p_y = 0.21 + 0.05 \times (\cos(2\pi t/10) - 1)$, $p_z = 0.3 + 0.05 \times \sin(2\pi t/10)$. The controller parameters were selected as follows: $k_0 = 0.2$, $k_\infty = 0.05$, $a = 1$, $\mu_1 = 1$, $\mu_2 = 1$, $k_{1i} = 20$, $k_{2i} = 10$, $h_{1i} = 0.01$, $h_{2i} = 0.01$, $\gamma = 0.6$, $c = 2$. The trajectory tracking performance of both was compared as shown in **Fig. 6**, and the Root Mean Square Error (RMSE) was used to analyze the trajectory tracking error to quantitatively compare the advantages of the designed controller as shown in **Fig. 7**. It was seen that the tracking accuracy of the proposed method is higher and the correctness of the aforementioned theoretical derivation was verified.

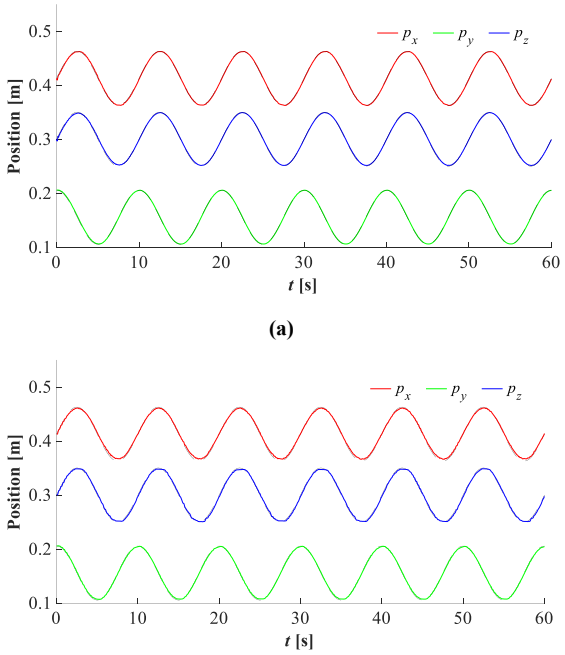


Fig. 6. Trajectory tracking performance comparison of the (a) proposed method with (b) traditional PD controller.

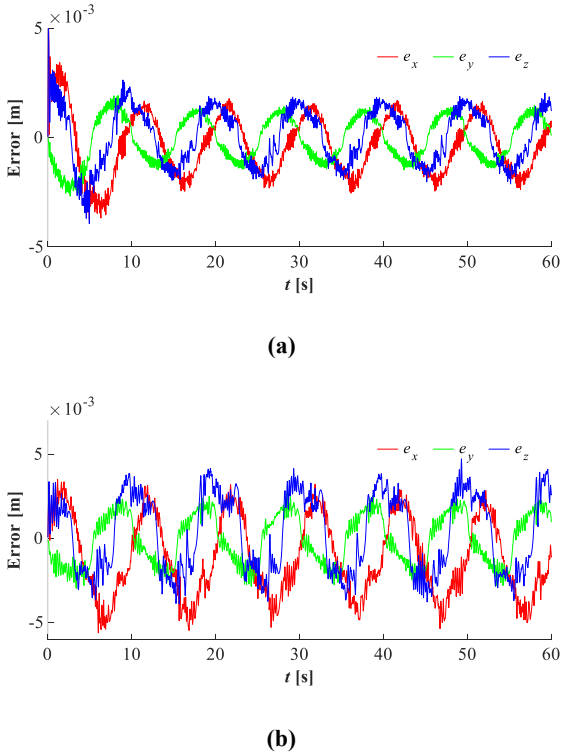
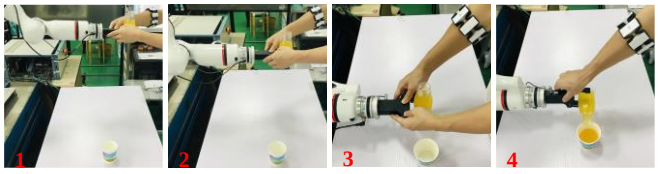


Fig. 7. Trajectory tracking errors comparison of the (a) proposed method with (b) traditional PD controller.

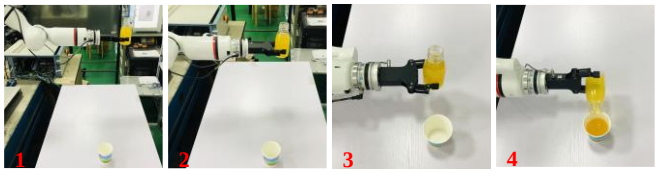
5.2. Test of the multi-space DMP

1): Demonstration: During this process, the human tutor transferred the skill of pouring water to the robot, including the adaptation of position, orientation and stiffness. The demonstration process decoupled the position and orientation of the water pouring skill to facilitate more accurate extraction of translational and rotational stiffness while conforming more to human operating habits. The tutor demonstrated four water pouring tasks. The position and quaternions of the end-effector and sEMG signals of the human arm were recorded for algorithmic analysis. The minimal and maximal translational stiffness values were set as $k_{\min} = 200$ (N/m) and $k_{\max} = 500$ (N/m), the values for the rotational stiffness were $k_{\min} = 10$ (Nm/rad), $k_{\max} = 20$ (Nm/rad) and $\eta = 15$.

2) Reproduction: The demonstration and reproduction process of the water pouring skill as shown in **Fig. 8**. It is noted that a plastic bottle filled with water was used in this case. **Fig. 9, Fig. 10** and **Fig. 12** show the reproduction results of the DMP for position, translational stiffness and rotational stiffness, respectively. **Fig. 11** shows the reproduction results of the RM-DMP for orientation.



(a)



(b)

Fig. 8. (a) Demonstration. (b) Reproduction.

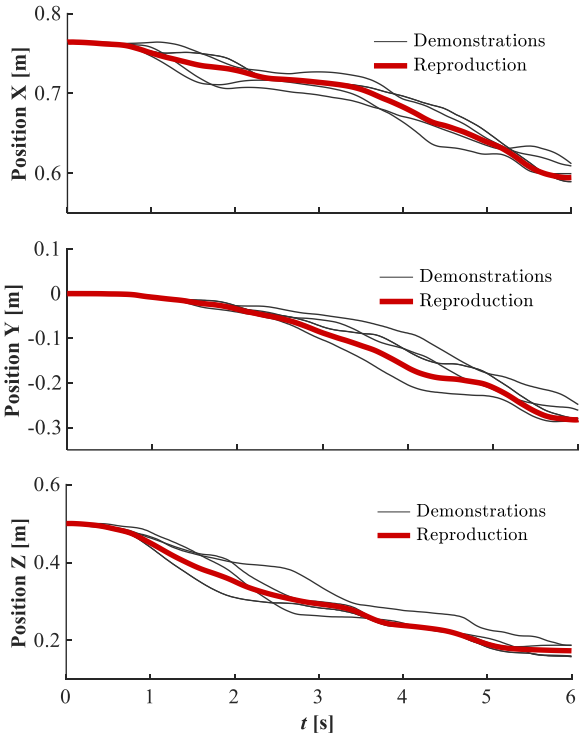


Fig. 9. Reproduction results of DMP for position.

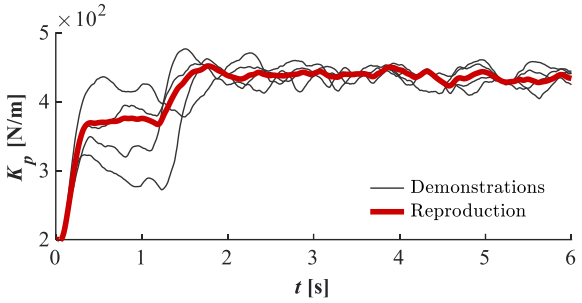
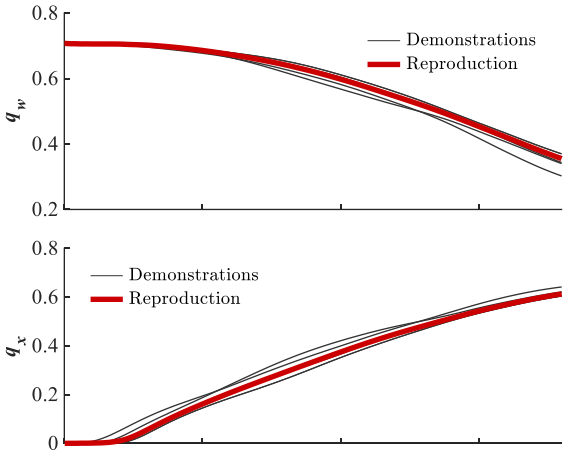


Fig. 10. Reproduction results of DMP for translational stiffness.



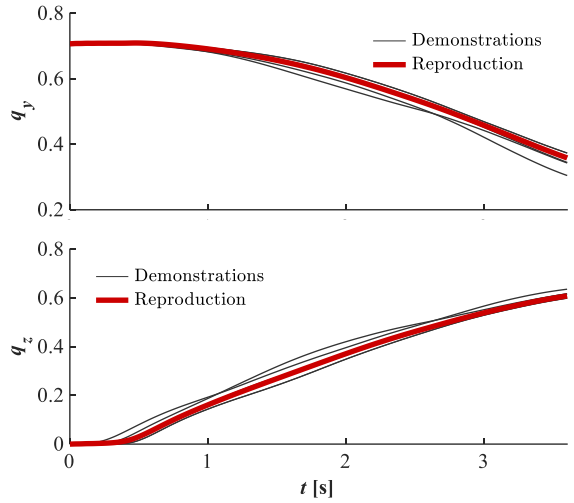


Fig. 11. Reproduction results of RM-DMP for Orientation.

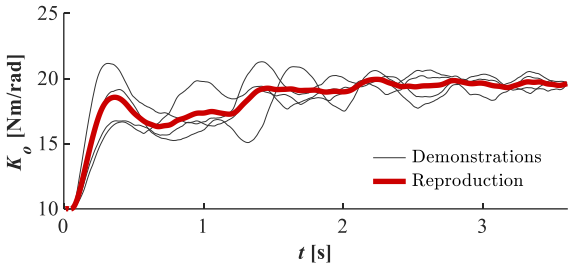
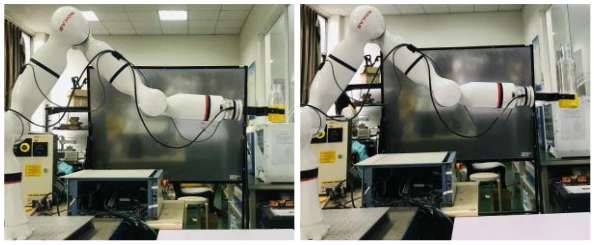


Fig. 12. Reproduction results of DMP for rotational stiffness.



(a) **(b)**

Fig. 13. Experimental scenario setup. (a) The plastic bottle with less water. (b) The glass bottle with less water.

3) Generalization: One of the advantages of the proposed framework was that it allows the robot to flexibly adapt the acquired skills, such as the re-adjustment of position, orientation and stiffness to a new given situation. In this subsection, five different operating conditions were set to verify the generalization of the learned skill.

Subtask 1: The plastic bottle filled with water was used and the robot performed the water pouring task with generalized position, reproduced orientation and stiffness.

Subtask 2: The plastic bottle with less water were used, as shown in **Fig. 13(a)**. The robot performed the task with generalized position, reproduced orientation and stiffness.

Subtask 3: The experimental setup of this case was same as subtask 2 and the robot performed the task with generalized position and orientation, as well as reproduced stiffness.

Subtask 4: The glass bottle with less water were used in this case, as shown in **Fig. 13(b)**. The robot performed the task with generalized position, orientation and reproduced stiffness.

Subtask 5: The experimental setup of this case was same as subtask 4 and the robot performed the task with generalized position, orientation and stiffness.

For Subtasks 1-5, **Fig. 14** shows the reproduced and measured positions, and **Fig. 15** shows the reproduced and measured orientation. Under Subtask 1 condition, the task was *successful* as shown in **Fig. 18(a)**, it indicated that the proposed method has good position generalization performance, i.e., adaptive for a given new position. However, Subtask 2 was *failed* as shown in **Fig. 18(b)**, which can be explained by the fact that it requires the joint generalization of position and orientation for such complex operational tasks. Further, Subtask 3 was *successful* as shown in **Fig. 18(c)**. As the operating conditions become more complex, i.e. the glass bottle is heavier and longer than the plastic one, which introduces a larger external force and torque. This leads to jitter when the robot executes translational motion and deviates from the desired orientation when it executes rotational motion. Therefore, Subtask 4 was *failed*. It can be explained by the fact that only position and orientation control cannot handle such tasks involving forces, which require adjusting the stiffness distribution to compensate for the mass variation of the robot end-effector. Therefore, Subtask 5 was *successful* as shown in **Fig. 18(d)** and the translational and rotational stiffness distributions are shown in **Fig. 16** and **Fig. 17**, respectively.

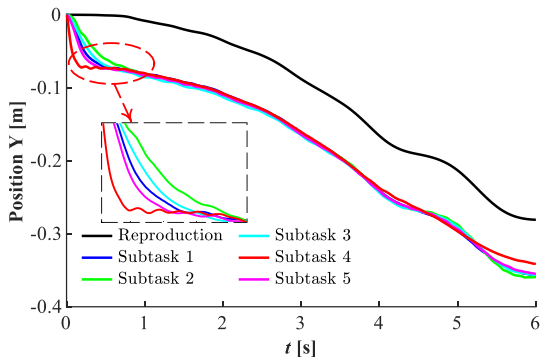


Fig. 14. Reproduced and measured position.

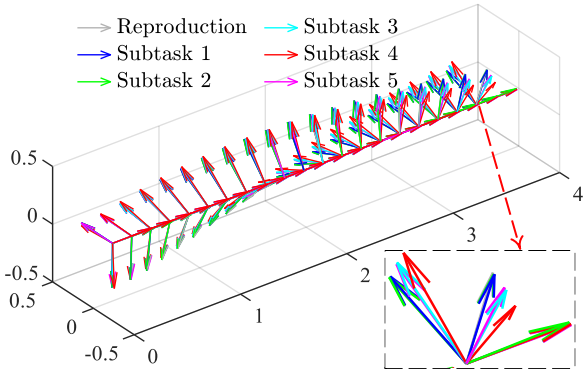


Fig. 15. Reproduced and measured orientation.

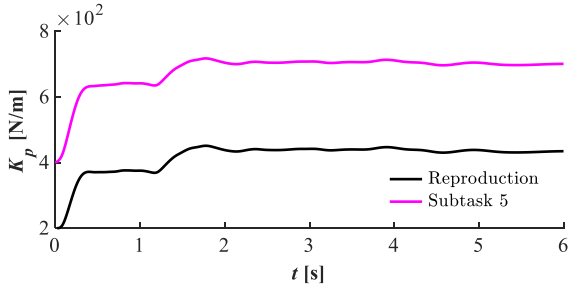


Fig. 16. Generalization of translational stiffness.

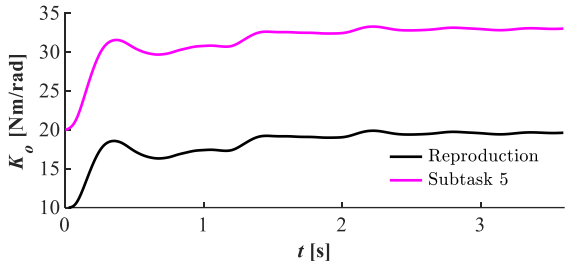
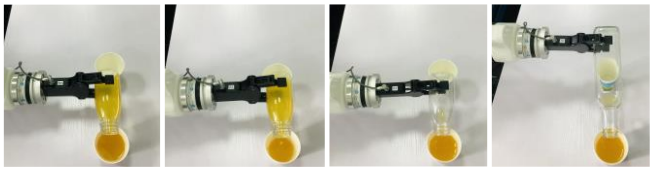


Fig. 17. Generalization of rotational stiffness.



(a) Subtask 1 **(b)** Subtask 2 **(c)** Subtask 3 **(d)** Subtask 5

Fig. 18. Comparison of the water poured by each subtask.

5.3. Discussion

For collaborative robots, it is difficult to achieve operation skills such as pouring water if only position profile is considered, since their motion also involves orientation evolution and tend to be perturbed. Our approach has the ability to solve this problem by enabling the robot to learn motor skills from human demonstrations, including position, orientation and stiffness information.

It should be mentioned that our approach does not include the learning of force information. However, it has been verified in human motor learning that stiffness and feedforward forces are learned separately [36, 37]. Our approach can be extended to encode force distribution. Some precision operational tasks may require highly accurate force/position control performance, in which case the dynamic evolution of the force needs to be well encoded and learned. One possible way to solve this problem is to add another component based on the proposed method to take force information into account.

A weakness of our approach is the accuracy of the estimated stiffness, since so far it is difficult to accurately calculate the human arm stiffness based on the sEMG signal. In this article we consider the stiffness as a diagonal matrix, and although this is sufficient for most tasks, the stiffness distribution may not be applicable in the case of more complex tasks. Therefore, another direction to improve our approach is to combine human upper limb dynamics and demonstration data to efficiently learn the complete stiffness.

6. Conclusions

In this article, we proposed a novel multi-space skill learning framework based on S^3 manifolds, DMP and adaptive finite-time admittance control, which can directly learn, reproduce and generalize kinematic/dynamic skills such as position, orientation and stiffness. The proposed approach can avoid any prior re-parameterization of non-Euclidean data such as quaternions and improves the ability of DMP to learn for multi-demonstrations based on GPR. The robot can learn

human-like motion skills from the demonstrations by human tutor with the aim of improving the robot's ability to perform contact tasks while ensuring safe interaction with the environment. The proposed framework is evaluated through an experimental study. Experimental results on the ROKAE ER3 Pro 7-DOF collaborative robot executing a water pouring task showed the effectiveness of the framework in performing contact tasks. It is worth noting that the framework is well generalized and easily transferable to other robotic platforms and has great potential in applications such as medical robots and other service robots that require integrated kinematic and dynamic skills. Our future work will focus on improving our approach, as described above.

Declaration of competing interest

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

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB1304800, in part by the National Natural Science Foundation of China under Grant 62203227 and in part by the Natural Science Foundation of Jiangsu Province under Grant BK20220442.

References

[1] G.Z. Yang, J. Bellingham, H. Choset, P. Dario, P. Fischer, T. Fukuda, N. Jacobstein, B. Nelson, M. Veloso, J. Berg, Science for robotics and robotics for science, Science Robotics, 1 (2016) eaal2099.

[2] Y. Zhang, M. Li, C. Yang, Robot learning system based on dynamic movement primitives and neural network, Neurocomputing, 451 (2021) 205-214.

[3] Z. Lu, N. Wang, Q. Li, C. Yang, A trajectory and force dual-incremental robot skill learning and generalization framework using improved dynamical movement primitives and adaptive neural network control, Neurocomputing, 521 (2023) 146-159.

[4] Z. Liao, G. Jiang, F. Zhao, Y. Wu, Y. Yue, X. Mei, Dynamic skill learning from human demonstration based on the human arm stiffness estimation model and Riemannian DMP, IEEE/ASME Transactions on Mechatronics, (2022) 1-12.

[5] L. Peternel, N. Tsagarakis, A. Ajoudani, A human-robot co-manipulation approach based on human sensorimotor information, IEEE Transactions on Neural Systems and Rehabilitation Engineering, 25 (2017) 811-822.

[6] N. Hogan, Impedance control: An approach to manipulation, American control conference, (1984) pp. 304-313.

[7] Z. Li, Y. Kang, Z. Xiao, W. Song, Human-robot coordination control of robotic exoskeletons by skill transfers, IEEE Transactions on Industrial Electronics, 64 (2017) 5171-5181.

[8] C. Zeng, X. Chen, N. Wang, C. Yang, Learning compliant robotic movements based on biomimetic motor adaptation, Robotics and Autonomous Systems, 135 (2021) 103668.

[9] X. Yu, P. Liu, W. He, Y. Liu, Q. Chen, L. Ding, Human-robot variable impedance skills transfer learning based on dynamic movement primitives, IEEE Robotics and Automation Letters, 7 (2022) 6463-6470.

[10] F.J. Abu-Dakka, L. Rozo, D.G. Caldwell, Force-based variable impedance learning for robotic manipulation, Robotics and Autonomous Systems, 109 (2018) 156-167.

[11] C. Yang, C. Zeng, C. Fang, W. He, Z. Li, A DMPs-based framework for robot learning and generalization of humanlike variable impedance skills, IEEE/ASME Transactions on Mechatronics, 23 (2018) 1193-1203.

[12] Y. Wu, F. Zhao, W. Kim, A. Ajoudani, An intuitive formulation of the human arm active endpoint stiffness, Sensors (Switzerland), 20 (2020) 1-15.

[13] S. Calinon, D. Bruno, D.G. Caldwell, A task-parameterized probabilistic model with minimal intervention control, in: IEEE International Conference on Robotics and Automation (ICRA), (2014) pp. 3339-3344.

[14] A.J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: Learning attractor models formotor behaviors, Neural Computation, 25 (2013) 328-373.

[15] Y. Huang, L. Rozo, J. Silvério, D.G. Caldwell, Kernelized movement primitives, International Journal of Robotics Research, 38 (2019) 833-852.

[16] Z. Jin, A. Liu, W.A. Zhang, L. Yu, C. Yang, Gaussian process movement primitive, Automatica, 155 (2023).

[17] M. Denisa, A. Gams, A. Ude, T. Petric, Learning compliant movement primitives through demonstration and statistical generalization, IEEE/ASME Transactions on Mechatronics, 21 (2016) 2581-2594.

[18] C. Zeng, Y. Li, J. Guo, Z. Huang, N. Wang, C. Yang, A unified parametric representation for robotic compliant skills with adaptation of impedance and force, IEEE/ASME Transactions on Mechatronics, 27 (2022) 623-633.

[19] Z. Lu, N. Wang, C. Yang, A constrained dmps framework for robot skills learning and generalization from human demonstrations, IEEE/ASME Transactions on Mechatronics, 26 (2021) 3265-3275.

[20] C. Yang, C. Zeng, Y. Cong, N. Wang, M. Wang, A learning framework of adaptive manipulative skills from human to robot, IEEE Transactions on Industrial Informatics, 15 (2019) 1153-1161.

[21] C. Yang, C. Chen, W. He, R. Cui, Z. Li, Robot learning system based on adaptive neural control and dynamic movement primitives, IEEE Transactions on Neural Networks and Learning Systems, 30 (2019) 777-787.

[22] S. Calinon, Gaussians on riemannian manifolds: Applications for robot learning and adaptive control, IEEE Robotics and Automation Magazine, 27 (2020) 33-45.

[23] F.J. Abu-Dakka, V. Kyrki, Geometry-aware dynamic movement primitives, in: IEEE International Conference on Robotics and Automation (ICRA), (2020) pp. 4421-4426.

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

[25] S. Schaal, J. Peters, J. Nakanishi, A. Ijspeert, Control, planning, learning, and imitation with dynamic movement primitives, workshop on bilateral paradigms on humans and humanoids: IEEE International Conference on Intelligent Robots and Systems (IROS), (2003) pp. 1-21.

[26] Z. Yang, J. Peng, Y. Liu, Adaptive neural network force tracking impedance control for uncertain robotic manipulator based on nonlinear velocity observer, Neurocomputing, 331 (2019) 263-280.

[27] A. Abooee, M. Moravej Khorasani, M. Haeri, Finite time control of robotic manipulators with position output feedback, International Journal of Robust and Nonlinear Control, 27 (2017) 2982-2999.

[28] L. Wang, H. Wang, P.X. Liu, S. Ling, S. Liu, Fuzzy finite-time command filtering output feedback control of nonlinear systems, IEEE Transactions on Fuzzy Systems, 30 (2022) 97-107.

[29] A.J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: learning attractor models for motor behaviors, Neural Computation, 25 (2013) 328-373.

[30] M.J.A. Zeestraten, I. Havoutis, J. Silverio, S. Calinon, D.G. Caldwell, An approach for imitation learning on riemannian manifolds, IEEE Robotics and Automation Letters, 2 (2017) 1240-1247.

[31] S. Calinon, D. Lee, Learning control, Humanoid Robotics: a Reference, (2016) 1-52.

[32] T. Alizadeh, M. Malekzadeh, S. Barzegari, Learning from demonstration with partially observable task parameters using dynamic movement primitives and Gaussian process regression, in: IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), (2016) pp. 889-894.

[33] F. Zhou, F. De La Torre, Generalized time warping for multi-modal alignment of human motion, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2012) pp. 1282-1289.

[34] Z. Li, X. Li, Q. Li, H. Su, Z. Kan, W. He, Human-In-the-Loop control of soft exosuits using impedance learning on different terrains, IEEE Transactions on Robotics, 38 (2022) 2979-2993.

[35] B. Cui, Y. Xia, K. Liu, G. Shen, Finite-time tracking control for a class of uncertain strict-feedback nonlinear systems with state constraints: A smooth control approach, IEEE Transactions on Neural Networks and Learning Systems, 31 (2020) 4920-4932.

[36] E. Burdet, G. Ganesh, C. Yang, A. Albu-Schäffer, Interaction force, impedance and trajectory adaptation: By humans, for robots, Springer Tracts in Advanced Robotics, 79 (2014) pp. 331-345.

[37] E. Burdet, R. Osu, D.W. Franklin, T.E. Milner, M. Kawato, The central nervous system stabilizes unstable dynamics by learning optimal impedance, Nature, 414 (2001) 446-449.

Chengguo Liu received the B.S. degree in Mechatronics from Wuhan Institute of Technology, Wuhan, China, in 2019. and the M.S. degree in Mechanical Engineering from Chongqing University, Chongqing, China, in 2022. He is currently pursuing a Ph.D. degree in the State Key Laboratory of Mechanical Transmission, Chongqing University, Chongqing, China. His research interests cover human-robot skill transfer, robot imitation learning and intelligent control.

Guangzhu Peng received the B.Eng. degree in automation from Yangtze University, Jingzhou, China, in 2014, the M.Eng. degree in pattern recognition and intelligent systems from the School of Automation Science and Engineering, South China University of Technology (SCUT), Guangzhou, China, in 2018, and the Ph.D. degree in computer science from the Faculty of Science and Technology, University of Macau, Macau, China, in 2021. His current research interests include robotics, human-robot interaction, intelligent control, etc.

Yu Xia received the B.Eng. degree in mechanical engineering from Chongqing University, Chongqing, China, in 2017. He is currently pursuing a Ph.D. degree in the State Key Laboratory of Mechanical Transmission at Chongqing University, Chongqing, China. His research interests include mechanical and electrical transmission control, stochastic systems, event-triggered, and adaptive control.

Junyang Li received the Ph.D. degrees in mechanical engineering from Chongqing University, Chongqing, China, in 2012. From 2013 to 2015, he engaged in the postdoctoral research at Chongqing University in the field of adaptive control of dynamic systems. He is currently an Associate Professor with the College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing, China. His research interests include precision transmission, and intelligent control.

Prof. Chenguang Yang received the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010, and postdoctoral training in human robotics from the Imperial College London, London, U.K. He was awarded UK EPSRC UKRI Innovation Fellowship and individual EU Marie Curie International Incoming Fellowship. As the lead author, he won the IEEE Transactions on Robotics Best Paper Award (2012) and IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award (2022). He is the Corresponding Co-Chair of IEEE Technical Committee on Collaborative Automation for Flexible Manufacturing, a Fellow of Institute of Engineering and Technology (IET), and a Fellow of British Computer Society (BCS). His research interest lies in human robot interaction and intelligent system design.