

Movement generalization of variable initial task state based on Euclidean transformation dynamical movement primitives

International Journal of Advanced Robotic Systems
November-December 2021: 1–17
© The Author(s) 2021
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/17298814211065577
journals.sagepub.com/home/arx



Boyang Ti , Yongsheng Gao, Ming Shi, Le Fu and Jie Zhao

Abstract

Robots need the ability to tackle problems of movement generalization in variable task state and complex environment. Dynamical movement primitives can effectively endow robots with humanoid characteristics. However, when the initial state of tasks changes, the generalized trajectories by dynamical movement primitives cannot retain shape features of demonstration, resulting in the loss of imitation quality. In this article, a modified dynamical movement primitives based on Euclidean transformation is proposed to solve this problem. It transforms the initial task state to a virtual situation similar to the demonstration and then utilizes the dynamical movement primitive method to realize movement generalization. Finally, it reverses the movement back to the real situation. Besides, the information of obstacles is added to Euclidean transformation based dynamical movement primitives framework to endow robots with the ability of obstacle avoidance. The normalized root-mean-square error is proposed as the criterion to evaluate the imitation similarity. The feasibility of this method is verified through writing letters, wiping whiteboard in two-dimensional task, and stirring mixture in three-dimensional task. The results show that the similarity of movement imitation in the proposed method is higher than dynamical movement primitives when the initial state changes. Meanwhile, Euclidean transformation based dynamical movement primitives can still greatly retain shape feature of demonstration while avoiding obstacles in an unstructured environment.

Keywords

Dynamical movement primitives, Euclidean transformation, movement generalization, learning from demonstration

Date received: 08 January 2021; accepted: 23 November 2021

Topic: AI in Robotics; Human Robot/Machine Interaction

Topic Editor: Alessandro Di Nuovo

Associate Editor: Alessandro Di Nuovo

Introduction

In recent years, traditional robot programming algorithms with poor generalization performance, low programming efficiency, and weak human–robot interaction cannot meet the need of effectively intelligent robots learning.¹ However, intelligent control algorithms based on learning from demonstration improve the generalization ability of manipulation skills, with robustness against disturbance and relatively easier interaction process. These algorithms improve the efficiency of skill learning and endow robots with humanoid characteristics.

This article focuses on the problem of improving the performance of imitation and generalization from human demonstration in any relationship between robot state and

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, Heilongjiang, People's Republic of China.

Corresponding author:

Yongsheng Gao, State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin 150001, Heilongjiang, People's Republic of China.

Email: gao_ys@126.com



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without

further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

task state. The methods of learning from demonstration can be mainly classified into two categories.

One is to construct probability model of demonstration characteristics based on statistical methods. Calinon et al.²⁻⁶ proposed a Gaussian mixture model–Gaussian mixture regression (GMM-GMR) learning framework to extract the feature distribution of trajectories in task space as well as manipulator joint space and applied the learned model to new task scenarios. Paraschos et al.⁷⁻¹⁰ proposed the probabilistic movement primitives, which introduced linear weighted basis functions using Gaussian to construct the probabilistic model. However, these statistics-based methods usually require a large number of demonstration samples to train a good probability model, obviously increasing the burden of demonstrators and the generalized movement is greatly affected by the coverage of human demonstration, which limits the extrapolation ability of robot.

The second kind of method is based on dynamic models of demonstration. Ijspeert et al.¹¹⁻¹³ proposed the method of dynamical movement primitives (DMPs), which can realize the generalization to different goals through single demonstration. This method is derived from the attractor behavior of autonomous nonlinear dynamic system and it can learn nonlinear movement features with good generalization and versatility. In recent years, a variety of optimization studies on this method have been emerging. Deniša et al.^{14,15} used Gaussian process regression and¹⁶ used GMM to construct regression model of task parameters as well as weight parameters of coupling force, which makes the generalization ability of DMPs more flexible. However, these methods ignore the poor generalization ability of DMPs in complex movement when the initial task states change. There are also some strategies introducing neural network to increase the performance of imitation. In some application scenarios, Su et al.¹⁷ combine the GMM-GMR method with DMPs to generalize the surgical movements in the open surgery, which ensures the accuracy of the action, and model the uncertainty of multiple sets of demonstrating data. For increasing the regression accuracy of the imitation human motion, Su et al.¹⁸ proposed an incremental learning framework based on the deep convolutional neural network approach for fast online human-like motion learning. But it needs a lot of data sets to construct the neural network. Qi et al.¹⁹ proposed adaptive human activity recognition real-time monitoring system to classify and identify human motions in dynamic situations.

In general, DMPs method has the following problems:

1. While the initial state is close to or coincides with the goal state, the movement cannot be generated or it will generate extreme acceleration which will lead to exceed the physical limit of robots.

Researches have proposed a variety of methods to solve this problem. For instance, Hoffmann et al.²⁰ proposed Bio-

DMPs method by designing improved DMPs in acceleration domain. Kulvicius et al.²¹ introduced another exponential decay system to replace the goal term in the second-order dynamical system.

2. When the relative state between initial and goal states changes such as in different relative quadrants, the generalized movement cannot retain the characteristics of demonstration.

To this problem, Caccavale et al.²² considered the linear part of the DMP and neglected the forcing terms to prevent the problem of the excessive magnification of trajectories generated from different initial states. Gräve and Behnke²³ utilized the linear transformation method to retain the original movement characteristics to segment and keep the relative quadrants of each segment same as the demonstration. Zhou et al.²⁴ introduced the follower and leader idea using local coordinate transformation to generate the movement in the varying coordinate system and introduced the concept of follower and leader to solve the goal tracking problem. However, our work focuses on solving the problem raised by the global change of the initial state. Forte et al.²⁵ constructed DMPs model in the joint space of robot in the very initial state reaching task; however, this strategy needs plenty of demonstrations to construct the movement primitives and cannot utilize the advantage of DMPs. Although its generalization in joint space can avoid the coordinates' system problems, its planning is limited by the joint limit which will raise the constraint planning problem of DMPs. Calinon⁵ proposed multiple coordinate fusion systems, where they constructed the distribution of demonstrations using multiple coordinate systems and segmented the imitated movement. In this article, we introduce the single coordinate system to represent only one demonstration, which keeps the consistency and improves the efficiency of the imitation. Cohen and Berman²⁶ have some similar parts of our method but they did not consider the relation of different dimensions. Meanwhile, their method did not take obstacles perturbation into account. Khansari-Zadeh and Billard²⁷ ensured the global asymptotic stability of the generalized movement.

In this article, we propose a Euclidean transformation based DMPs (ET-DMPs) method to solve the second problem. We summarize the innovation in our work in the following three points:

1. We extract the relationship between current robot state and demonstration, which is used for the transformation between the actual and virtual situation.
2. The original DMPs are optimized by introducing the relationship obtained from demonstration, which is the proposed ET-DMPs method. It transforms and reverses the generalized movement using the relationship, which effectively guarantees the imitation and generalization performance of robots.

3. In simulation and experiment, the proposed method is compared with DMPs by using the similarity indicators nRSME to highlight the advantages of ET-DMPs. Besides, single and multiple obstacles information through Euclidean transformation are also added to the framework.

The organization of this article is divided into four sections: The second section describes the principle of DMPs. The third section introduces the ET-DMPs method and the extension for obstacle avoidance. The fourth section verifies the validity of ET-DMPs through simulation and experiment. The fifth section discusses the limitation of our method. The sixth section presents the conclusion of this article.

Dynamical movement primitives

DMPs have a good generalization performance in both discrete and periodic movement. Combining these two kinds of motion deliberately could generate more complex movement. It is originally a method applied to the representation of one-dimensional movement. When extended to multidimensional space, the coupling relation is established through the phase variables in canonical system and then we could get the multidimensional synchronous movement, although each dimension is independent.

DMPs are derived from the spring-damping dynamic attractor system combined with the nonlinear coupling force. Equations (1) and (2) give the expression of this system as follows

$$\tau \dot{\mathbf{v}} = \mathbf{K}(\mathbf{g} - \mathbf{x}) - \mathbf{D}\mathbf{v} + (\mathbf{g} - \mathbf{x}_0)\mathbf{f}(s) \quad (1)$$

$$\tau \dot{\mathbf{x}} = \mathbf{v} \quad (2)$$

where \mathbf{K} is the spring stiffness coefficient, \mathbf{D} is the damping constant, τ represents a constant scale of the movement duration, \mathbf{x} , \mathbf{v} , $\dot{\mathbf{v}}$ is the current position, velocity, and acceleration of the system, \mathbf{x}_0 represents the initial state of system, \mathbf{g} is the goal state and \mathbf{x} will converge toward \mathbf{g} monotonically with appropriate values of \mathbf{K} and \mathbf{D} , and s is a phase term which drives the system to going on. \mathbf{f} is a coupling forcing term that can be fitted by linear weighted basis function. The expression of forcing term is defined as follows

$$\mathbf{f}(s) = \frac{\sum_i \psi_i(s) \boldsymbol{\omega}_i}{\sum_i \psi_i(s)} \quad (3)$$

where term ψ_i is the basic function to fit the coupling forcing term typically selecting Gaussian kernel function to get a good performance as shown in the following equation

$$\psi_i(s) = \exp\left(-\frac{1}{2\sigma_i^2}(s - c_i)^2\right) \quad (4)$$

The function of phase s is to eliminate the dependence on time, and the definition of it refers to in the following equation

$$\tau \dot{s} = -\alpha s \quad (5)$$

where α is a constant and s can express the progress of a task with the initial value set to 1. It can be derived from Eq. (5) that as the movement going on, the phase s approaches to 0, and the influence of the coupling force term will gradually weaken to guarantee that the system is approximate to a linear spring-damping system, so it can converge to the goal point in the end. In addition, the duration of the movement can be adjusted by setting the constant τ .

The coupling force term \mathbf{f} is the key factor to extract the characteristics from demonstration. Through the weighted basis functions, \mathbf{f} corresponding to different demonstrations will be obtained. Each feature function corresponds to different groups of weight parameter ω_i , which could be learned through imitating from demonstration. Firstly, the trajectory information $\{x(t), v(t), \dot{v}(t)\}_{t=0 \dots T}^D$ is recorded from human demonstration. Equation (1) is then transformed into Eq. (6) and the coupling term of the target \mathbf{f}_{target} can be computed using the information from demonstration, while the corresponding phase variable at each timesteps is obtained by Eq. (5). The information of the essential feature is represented as $\{\mathbf{f}_{target}, s\}$, and the weight parameters ω_i corresponding to the \mathbf{f}_{target} are obtained by using locally weighted regression (LWR)²⁸ based on Eq. (3)

$$\mathbf{f}_{target} = \frac{\tau \dot{\mathbf{v}} - \mathbf{K}(\mathbf{g} - \mathbf{x}) + \mathbf{D}\mathbf{v}}{\mathbf{g} - \mathbf{x}_0} \quad (6)$$

According to the above introduction, DMPs have a good performance when goal changes. For example, when the relative state between initial and goal states changes, as shown in Figure 1, the generalized trajectory could still retain the characteristic of the demonstration. However, when the initial state changes, the beginning parts of the generalized trajectory lose the demonstration characteristic as the green-dotted line circle. Meanwhile in some tasks, we need to adapt the different feed directions. For example, we need to imitate the movement just like three-dimensional (3D) space part in Figure 1, which may appear when stirring mixture. But DMPs encoded the movement of different dimensions independently, which loses the movement relations of the interdependent dimensions, so it is hard to adapt the feed direction of movement.

There is another strategy for representing the movement, which employs the polar or spherical coordinate system. However, another problem will arise when angles are introduced into the position representation, namely that the angles change abruptly when the position crosses the boundary 2π . It is difficult to distinguish between 0 and 2π . In addition, it also has the same problem as Figure 1 that shows in one-dimensional (1D) space.

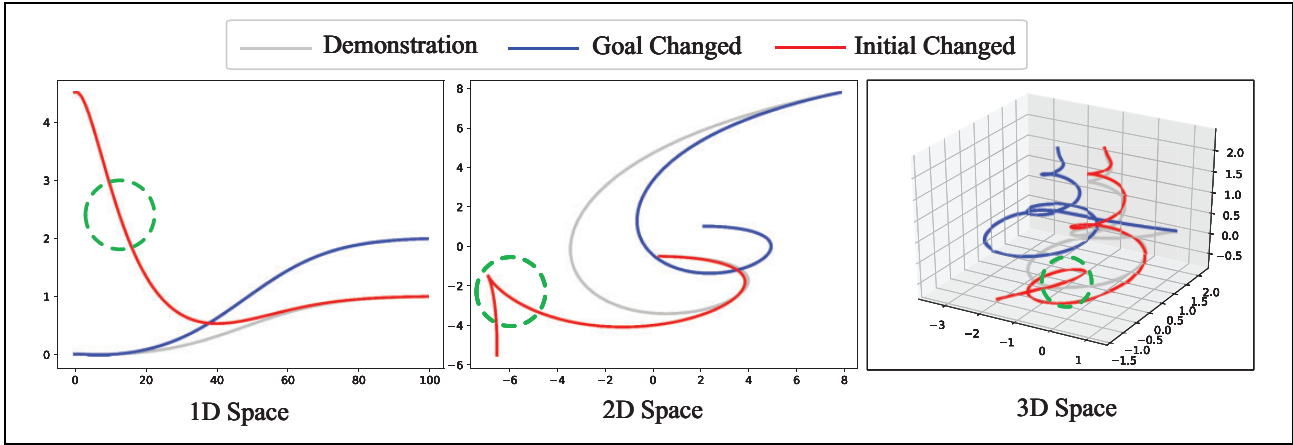


Figure 1. Generalized trajectories of DMPs in three spaces in the case of varying initial and goal states, respectively. DMP: dynamical movement primitives.

Euclidean transformation based DMPs

In this section, ET-DMPs are introduced and obstacle information is added to the ET-DMPs in an attempt to verify the robustness of the method under the interference of obstacles. The overall framework of ET-DMPs proposed in this article is shown in Figure 2 and the algorithm process of ET-DMPs is illustrated in Algorithm 1. The yellow area represents ET-DMPs, and the blue area represents Euclidean Transformation, which can be clearly seen that this framework can still inherit the advantage of DMPs. The region within the dash-dotted line can be replaced by any other optimized DMPs methods (which denoted by DMP+) such as Bio-DMPs²⁰ and task-specific DMPs,²⁹ without affecting the performance of the framework.

$t = 0$ and $t = T$. Firstly, the initial states are transformed from global coordinate to the coordinate of the goal, as shown in Figure 3, in 1D space; we transform the new initial states to be consistent with the initial state of demonstration.

For two-dimensional (2D) space, $I_1^C \sim I_4^C$ are four different initial states in the local coordinate system. To solve the problem of poor imitation ability due to the initial state in different quadrants, we rotate initial states to the same line as the initial demonstration represented as $I_1' \sim I_4'$. We call them virtual states in the following article.

For the generalization in 3D space, we propose two transformation modes. Mode 1 is similar to the method in 2D space but using Rodrigues rotation formula³⁰ to rotate

Algorithm 1. Process of our framework.

① Demonstration Sampling:

Get human demonstration data $\{t, \mathbf{x}(t), \mathbf{v}(t), \dot{\mathbf{v}}(t)\}_{t=0 \dots T}^D$

② Euclidean Transformation Dynamical Movement Primitives (ET-DMPs)

Input: Demonstration Data $\{t, \mathbf{x}(t), \mathbf{v}(t), \dot{\mathbf{v}}(t)\}_{t=0 \dots T}^D$ and Current Initial State \mathbf{x}_0^C

Output: Generalized Trajectory $\{\mathbf{x}(s), \mathbf{v}(s), \dot{\mathbf{v}}(s)\}_{s=1 \dots 0}^G$

(*Relation between s and t reference to Eq.(5))

1: Encoded demonstration data by DMPs

2: Extract the relative state between initial and goal from the demonstration $\mathbf{I}^D = \mathbf{x}_0^D - \mathbf{g}^D$

3: Get current relative state between initial and goal $\mathbf{I}^C = \mathbf{x}_0^C - \mathbf{g}^C$

4: Euclidean Transform according to the relation between \mathbf{I}^C and \mathbf{I}^D

5: Generalize movement according to the Eq.(11) - Eq.(12)

6: Reverse the movement generalized in virtual situation according to the Eq.(13) - Eq.(14)

③ Control Robot to Follow the Generalized Movement

Input: Generalized Trajectory $\{\mathbf{x}(s), \mathbf{v}(s), \dot{\mathbf{v}}(s)\}_{s=1 \dots 0}^G$

Output: Robot Joint Velocity $\{\dot{q}_1 \dots \dot{q}_6\}$ (using 6-dofs robot as manipulator)

The principle of ET-DMPs

The feature information obtained from demonstration in the global coordinate system includes , where T represents the total duration of the movement, the initial state and goal state, respectively, correspond to the trajectory when

the initial states to the same line as the demonstration, and in mode 2 the task initial states are rotated to a plane which is constructed by the vectors of initial demonstration and the axis of the independent dimension (in Figure 3 this dimension is z). This independent axis is not involved in

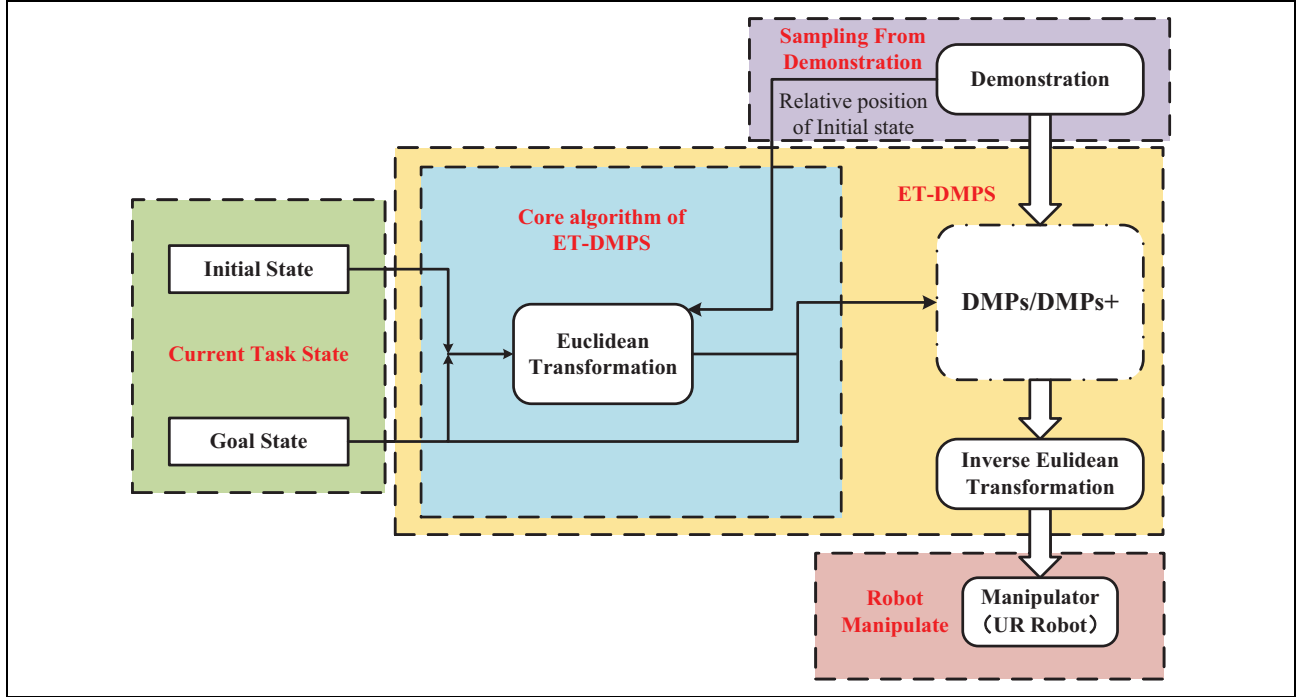


Figure 2. Flow diagram of ET-DMPs system. ET-DMPs: Euclidean transformation based dynamical movement primitives.

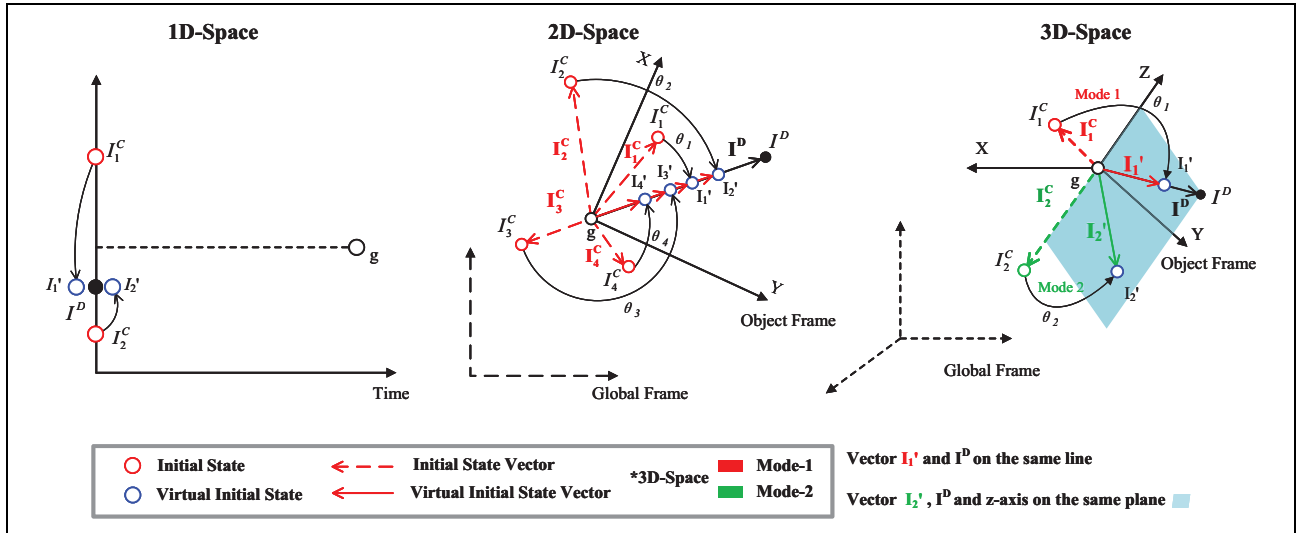


Figure 3. Euclidean transformation in 1D, 2D, and 3D spaces (with two modes). 1D: one-dimensional; 2D: two-dimensional; 3D: three-dimensional.

the rotation process but follows the generalization principle of 1D space. In this mode, the 3D space problem converts to a 1D and a 2D space generalization, respectively. Mode 1 is usually used when motions of three dimensions need Euclidean Transformation according to the task state. If task state has only impact on two dimensions, mode 2 will be selected.

Actually, points in any quadrants can be switched into the desired quadrant through Euclidean transformation. For 1D space, we change the initial states using scale factor n as

Eq. (7) shown, I is the initial state, I' is the virtual state scaled from I using n , g^C is the current goal state, and g^D is the demonstration goal state

$$I' = \frac{\text{sign}(I^C - g^C)}{\text{sign}(I^D - g^D)} \cdot n \cdot (I^C - g^D) + g^C \quad (7)$$

$$n = \left| \frac{I^C - g^C}{I^D - g^D} \right|$$

Meanwhile, the mean transformation of the 2D space and mode 2 of 3D space where we only use the planar rotation is as follows

$$\mathbf{I}' = \bar{\mathbf{R}}(\theta) \cdot \mathbf{I}$$

$$\bar{\mathbf{R}}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (8)$$

For mode 1 of 3D space, the Rodrigues rotation formula is as follows

$$\mathbf{I}' = \cos\theta \mathbf{I}^C + (1 - \cos\theta) (\mathbf{kI}^C) \mathbf{k} + \sin\theta \mathbf{k} \times \mathbf{I}^C \quad (9)$$

where \mathbf{I}^C is the vector of current state and \mathbf{I}^D is the initial state of demonstration. $\bar{\mathbf{R}}(\theta)$ is the planar rotation matrix and the rotation axis \mathbf{k} is the unit vector. The virtual state \mathbf{I}' is obtained through rotating \mathbf{I}^C by θ and the rotation angle θ is equal to the angle between the initial position of generalization and that of demonstration, corresponding to $\{\theta_1, \theta_2, \theta_3, \theta_4\}$ in Figure 3 They are computed through the following equation

$$\theta_i = \arccos\left(\frac{\mathbf{I}_i^C \cdot \mathbf{I}^D}{\|\mathbf{I}_i^C\| \|\mathbf{I}^D\|}\right) \mathbf{I}_i^C \in \{\mathbf{I}_1^C \sim \mathbf{I}_4^C\} \theta_i \in \{\theta_1 \sim \theta_4\} \quad (10)$$

After above transformation, the movement is generalized in the virtual situation. The formulation of ET-DMPs model is as follows

$$\tau \dot{\mathbf{v}}' = \mathbf{K}(\mathbf{g}^C - \mathbf{x}') - \mathbf{D}\mathbf{v}' + (\mathbf{g}^C - \mathbf{x}_0') \mathbf{f}(s) \quad (11)$$

$$\mathbf{x}_0' = \bar{\mathbf{R}}(\theta) \mathbf{I}^C + \mathbf{g}^C \quad (12)$$

$$\mathbf{I}^C = \mathbf{x}_0^C - \mathbf{g}^C$$

where $\{\mathbf{x}', \mathbf{v}'\}$ denotes the position and velocity of generalized movement in the virtual situation, respectively, and \mathbf{x}_0' is the virtual initial position. Then, by using Eqs (13) and (14), the trajectory obtained from virtual situation is transformed back to real situation, where $\{\mathbf{x}, \mathbf{v}\}$ represents the position and velocity in real situation. Finally, the executable generalized trajectory is obtained and passed to the manipulators

$$\mathbf{x} = \bar{\mathbf{R}}(-\theta)(\mathbf{x}' - \mathbf{g}^C) + \mathbf{g}^C \quad (13)$$

$$\mathbf{v} = \dot{\mathbf{x}} \quad (14)$$

For quantifying the similarity of the imitation, we introduce the similarity evaluation method, which computes the root-mean-square error between the generalized and demonstration trajectories after normalizing and rotating them into the same similarity metric. Firstly, we rotate the generalized trajectory by the angle computed from Eq. (10) making sure that \mathbf{I} and \mathbf{I}_{demo} are on the same line, and we define the rotated trajectories as D . Then we normalize the demonstration and generalization trajectory using Eq. (15), where D_x, D_y , and D_z represent the position of the trajectory along x, y , and z axes (if in 3D space we introduce the z axes trajectory). We then compute the error using Eq. (16) to evaluate the similarity of the trajectory

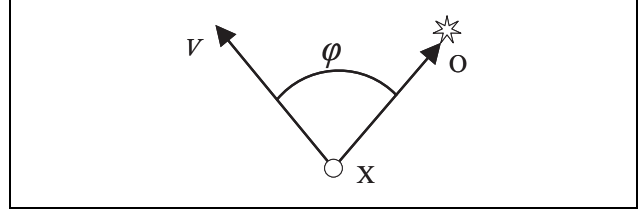


Figure 4. Illustration of obstacle avoidance.

$$D_{normal} = \frac{D}{\max(|D_x|, |D_y|, |D_z|)} \quad (15)$$

$$L_{error} = \sqrt{\sum_k^{x,y,z} \left(\frac{1}{T} \sum_i^T (D_{normal-i}^k - D_{demo-i}^k)^2 \right)^2} \quad (16)$$

Obstacle avoidance method based on ET-DMPs

Besides imitating and generalizing in simple environment, robots will also encounter complex environment in daily life, such as scene with obstacles. Therefore, we add the obstacle information to the ET-DMPs framework.

Obstacle avoidance in the DMPs. If obstacles exist in the task environment, robots will need robust environmental adaptability to avoid obstacles. The effect of obstacles is exactly contrary to that of the attractor in DMPs. As an external disturbance of the system, obstacles function as a repulsive role during the movement. Similar to the principle of artificial potential field,³¹ the coupling term $\lambda(\mathbf{x}, \mathbf{v})$ representing the repulsion force generated by obstacles is added to Eq.(17), and this coupling term is the function of the current speed as well as the relative position between the current state and the obstacle

$$\tau \dot{\mathbf{v}}' = \mathbf{K}(\mathbf{g} - \mathbf{x}) - \mathbf{D}\mathbf{v}' + (\mathbf{g} - \mathbf{x}_0) \mathbf{f}(s) + \lambda(\mathbf{x}, \mathbf{v}) \quad (17)$$

Inspired by human obstacle avoidance behavior, Fajen and Warren³² constructed a point obstacle avoidance model, in which the differential equation is as Eq.(18), where ϕ is the steering angle which is decided by current obstacle position and system state (position and velocity) and the constants γ and β affect the sensitivity to obstacle. The schematic diagram is shown in Figure 4. To combine this obstacle avoidance function with DMPs, the function between the steering angle velocity and the acceleration of current state $\lambda(\mathbf{x}, \mathbf{v})$ is defined in Hoffmann et al.²⁰

$$\dot{\phi} = \gamma \phi \exp(-\beta |\phi|) \quad (18)$$

$$\phi = \cos^{-1} \left(\frac{(\mathbf{o} - \mathbf{x})^T \mathbf{v}}{(\|\mathbf{o} - \mathbf{x}\| \cdot \|\mathbf{v}\|)} \right) \quad (19)$$

$$\dot{\mathbf{v}} = \mathbf{R} \mathbf{v} \dot{\phi} \quad (20)$$

In Eq. (20), \mathbf{R} is the rotational matrix representing the modified direction rotated by $\pi/2$ while regarding

$\mathbf{r} = (\mathbf{o} - \mathbf{x}) \times \mathbf{v}$ as the axis, and \mathbf{o} is current obstacle position. Then the modified velocity is determined by steering angular velocity. It can be concluded that the coupling term generated from the repulsion force of obstacles as Eq. (21) shown affects the shape of the generalized trajectory

$$\lambda(\mathbf{x}, \mathbf{v}) = \gamma \mathbf{R} \mathbf{v} \phi \exp(-\beta |\phi|) \quad (21)$$

Besides, in environment with multiple obstacles, the coupling term of avoidance needs to contain the information of multiple repulsion force. Therefore, $\lambda(\mathbf{x}, \mathbf{v})$ turns into the following form as Eq. (22) shown

$$\lambda(\mathbf{x}, \mathbf{v}) = \gamma \sum_i \mathbf{R}_i \mathbf{v} \phi_i \exp(-\beta |\phi_i|) \quad (22)$$

where $\phi_i = \cos^{-1} \left(\frac{(\mathbf{o}_i - \mathbf{x})^T \mathbf{v}}{\|\mathbf{o}_i - \mathbf{x}\| \|\mathbf{v}\|} \right)$

Euclidean transformation of obstacles. Combined with the obstacle avoidance introduced in the section ‘‘Obstacle avoidance in the DMPs’’ and Euclidean transformation introduced in the section ‘‘The principle of Euclidean transformation based DMPs,’’, the position of obstacles can be

rotated by the same angle and direction as the initial state of robots. The expression of obstacles after Euclidean transformation is expressed as follows

$$\mathbf{o}_i' = \bar{\mathbf{R}}(\mathbf{o}_i - \mathbf{g}^C) + \mathbf{g}^C \quad (23)$$

In the case of single obstacle and multiple obstacles, the corresponding coupling terms after Euclidean transformation are modified as follows

$$\lambda'(\mathbf{x}', \mathbf{v}') = \gamma' \sum_i \mathbf{R}_i' \mathbf{v}' \phi_i' \exp(-\beta |\phi_i'|) \quad (24)$$

Then the generalized obstacle avoidance trajectory obtained in virtual state is restored to the real task through inverse Euclidean transformation, after which the movement is transferred to a robot.

However, the disturbance of obstacles can be ignored when the robot enters the safe region. In the safe region, the distance from current position to the goal is less than that to the obstacle, thus the attraction of the goal is the main driving force, as shown in the following equation

$$\lambda'(\mathbf{x}', \mathbf{v}') = \begin{cases} \gamma' \sum_i \mathbf{R}_i' \mathbf{v}' \phi_i' \exp(-\beta |\phi_i'|) & \text{if } \|\mathbf{o}_i - \mathbf{x}'\| < \|\mathbf{x}' - \mathbf{g}^C\| \\ 0 & \text{if } \|\mathbf{o}_i - \mathbf{x}'\| > \|\mathbf{x}' - \mathbf{g}^C\| \end{cases} \quad (25)$$

The function $\dot{\phi} = \phi \exp(-\beta |\phi|)$ is 0 for $\phi = 0$, which means the force is 0 when directly heading towards the obstacle. Referring to Rai et al.,³³ we add a new function

$$\lambda''(\mathbf{x}', \mathbf{v}') = \begin{cases} \gamma'' \sum_i \mathbf{R}_i' \mathbf{v}' \exp(-kd_i) & \text{if } \|\mathbf{o}_i - \mathbf{x}'\| < \|\mathbf{x}' - \mathbf{g}^C\| \\ 0 & \text{if } \|\mathbf{o}_i - \mathbf{x}'\| > \|\mathbf{x}' - \mathbf{g}^C\| \end{cases} \quad (26)$$

$$\lambda^* = \lambda' + \lambda'' \quad (27)$$

Finally, the Lyapunov equation is cited to prove that the movement still converges to the goal state even when information of obstacles is transferred into the virtual situation. The goal state is assumed to be a fixed point $(\mathbf{x}'; \mathbf{v}') = (\mathbf{g}^C; \mathbf{0})$, to which the initial state in any position will converge. The energy equation of spring-damping system with unit mass $\dot{\mathbf{v}}' = \mathbf{K}(\mathbf{g}^C - \mathbf{x}') - \mathbf{D}\mathbf{v}'$ is used and then the Lyapunov equation is as follows

$$V(\mathbf{x}', \mathbf{v}') = \frac{1}{2} (\mathbf{g}^C - \mathbf{x}')^T \mathbf{K} (\mathbf{g}^C - \mathbf{x}') + \frac{1}{2} \mathbf{v}'^T \mathbf{v}' \quad (28)$$

To prove system convergence, it should be guaranteed that $\dot{V}(\mathbf{x}'; \mathbf{v}') < 0$ as well as $\mathbf{v}' \neq 0$, and we get the following deduction

as Eq. (26) to the coupling term which forms a field of repulsion away from the obstacle, where d_i represent the current distance to the obstacles

$$\begin{aligned} \dot{V} &= \nabla_{\mathbf{x}'} V^T \dot{\mathbf{x}}' + \nabla_{\mathbf{v}'} V^T \dot{\mathbf{v}}' = -(\mathbf{g}^C - \mathbf{x}')^T \mathbf{K} \mathbf{v}' + \mathbf{v}'^T \dot{\mathbf{v}}' \\ &= -\mathbf{v}'^T \mathbf{K} (\mathbf{g}^C - \mathbf{x}') + \mathbf{v}'^T \mathbf{K} (\mathbf{g}^C - \mathbf{x}') - \mathbf{v}'^T \mathbf{D} \mathbf{v}' \\ &\quad + \gamma' \mathbf{v}'^T \mathbf{R}' \mathbf{v}' \phi' \exp(-\beta \phi') + \gamma'' \mathbf{v}'^T \mathbf{R}' \mathbf{v}' \exp(-kd) \\ &= -\mathbf{v}'^T \mathbf{D} \mathbf{v}' < 0 \end{aligned} \quad (29)$$

where $\mathbf{v}'^T \mathbf{R}' \mathbf{v}'$ equals to 0 due to the rotation angle of the rotation matrix \mathbf{R}' is 90° . The above proves that DMPs can converge to a fixed goal $(\mathbf{x}'; \mathbf{v}'; \dot{\mathbf{v}}') = (\mathbf{g}^C; \mathbf{0}; \mathbf{0})$ even after the initial state and obstacle state are transformed into the virtual state.

Simulation and experiment

To highlight the superiority of the method proposed in this article, the DMPs are used as comparative verification. In the following, parameters setting are presented as Table 1.

ET-DMPs: Euclidean transformation based DMPs; DMPs: dynamical movement primitives

Simulation

Verification of ET-DMPs. To verify ET-DMPs can imitate complex trajectories, the letters G, P, R, and W in the alphabet data set (<https://gitlab.idiap.ch/rli/pbdlb-matlab/>) are selected as demonstration. The simulation environment is built in Python. To reflect the adaptability of ET-DMPs to any initial states, two points are randomly selected from each quadrant of the goal state as the generalized initial state in simulation. The position of selected points is illustrated in Table 2, and these test points are going through all the quadrants.

The whole verification process can be divided into four steps. Firstly, according to the data set, an original DMPs model, which encodes the characteristic of demonstration, is trained. Secondly, the relations between initial and goal points are extracted from data set. Then the relation of current states is compared with the demonstration and use Euclidean Transformation according to different space principles mentioned in Figure 3 to convert current state to virtual situation. Finally, the generalized movement is

Table 1. The experimental parameters setting of ET-DMPs and DMPs.

ET-DMPs/DMPs parameters	K	D	τ	Num. of basis function
Value	156.25	25	1	100

Table 2. The initial and target positions in the global coordinate system.

Relative quadrant	Letter G	Letter P	Letter R	Letter W
I	[5.80, 4.25] [2.80, 1.25]	[3.40, 3.10] [5.00, -3.15]	[14.40, -2.00] [13.20, -3.00]	[16.20, 15.25] [13.80, 15.10]
II	[-3.40, 3.10] [-5.00, 3.15]	[-4.20, 4.25] [-7.80, 5.10]	[-5.80, -1.25] [0.50, 3.25]	[-5.80, 10.25] [-6.30, 9.25]
III	[-8.80, -7.25] [-5.00, -5.25]	[-5.80, -6.25] [-4.80, -3.25]	[-3.80, -12.25] [0.00, -14.25]	[-3.80, 2.25] [0.00, 0.25]
IV	[4.20, -2.25] [7.80, -5.10]	[3.80, -7.25] [5.00, -5.25]	[16.20, -10.25] [13.80, -11.10]	[19.40, 0.00] [18.50, -3.15]
Goal Position	[0.11, -0.51]	[-3.79, 0.16]	[7.18, -9.15]	[9.61, 6.83]

reversed back to real situation. Simulation results are shown in Figure 5, where the star and the ring in each subfigure individually represent the initial state and goal state. The first and second rows individually represent trajectories generated by ET-DMPs and DMPs. Original DMPs have a good performance while approaching to the goal, but at the beginning of the trajectory, they have a terrible tracking performance. In contrast, the trajectories generated by ET-DMPs have a perfect imitation of shape while retaining the intrinsic characteristic of the demonstration. We can conclude from the results that if the generalized initial state is in the same quadrant as that of demonstration, trajectories generated from ET-DMPs and DMPs could all maintain the shape of the letters. However, if the initial state is in different quadrants, the trajectories generalized by DMPs deviated terribly from the shape of demonstration while only ET-DMPs could greatly maintain the shape of the letters.

Then, we generalize the movement represented in the polar coordinate system, as Figure 6 shows. The first row shows the trajectories generalized by DMPs in the polar coordinate system. The last row plot shows two types and colors of lines, where the blue- and red-dashed lines are the demonstration trajectories of radius and angle, blue and red solid lines are the generalized trajectory of radius and angle in the polar coordinate. We can find that the movement generalized in the polar coordinate still cannot retain the features of the demonstration. Meanwhile, it has the same problem as the movement generalized in Cartesian coordinate, where we can conclude from the trajectories exhibited in the last row. When the initial state is far from the

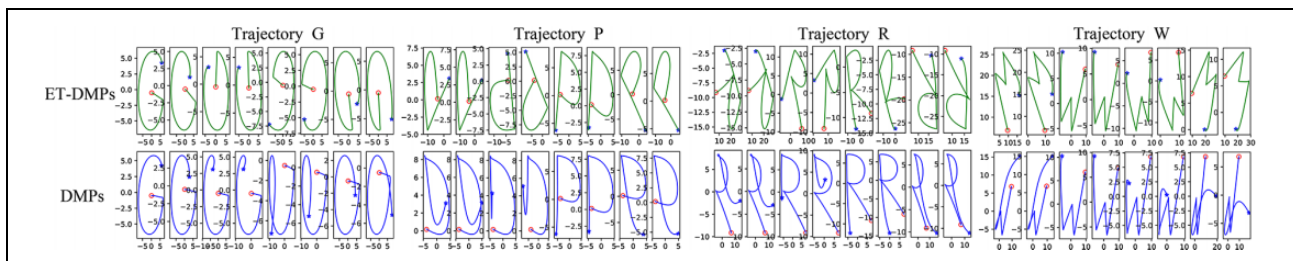


Figure 5. Comparison of the generalized trajectories of ET-DMPs and DMPs in 2D space. ET-DMPs: Euclidean transformation based DMPs; DMPs: dynamical movement primitives; 2D: two-dimensional.

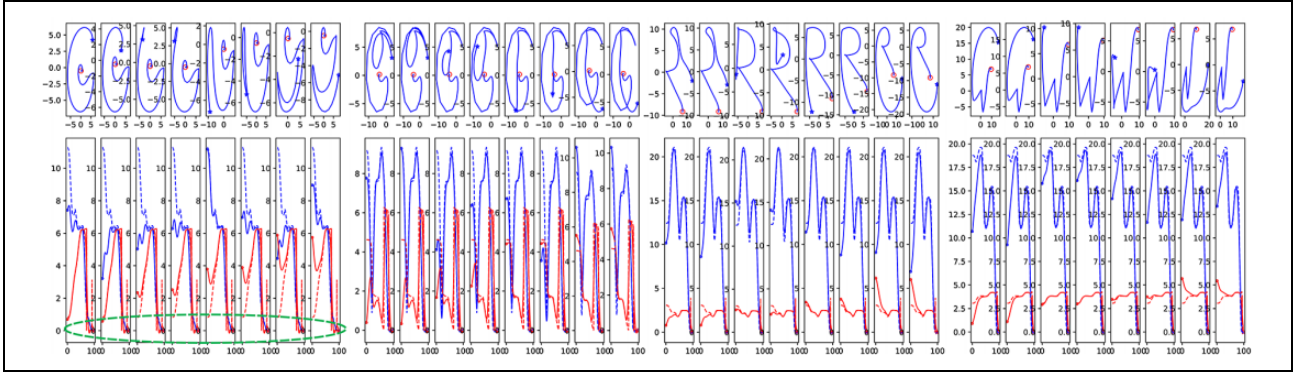


Figure 6. Generalized trajectories of DMPs in polar coordinate system. DMPs: dynamical movement primitives.

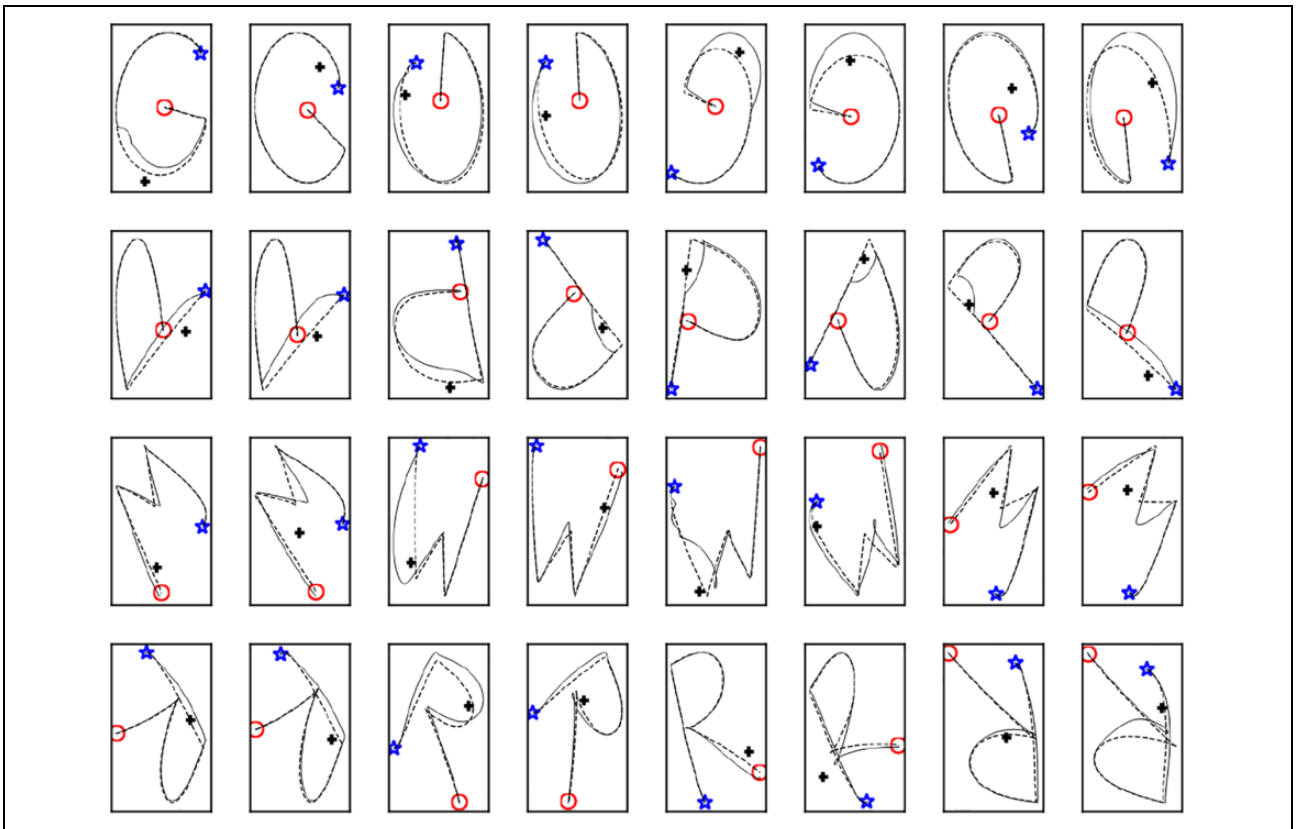


Figure 7. Trajectories generalized from ET-DMPs in single obstacle avoidance task. ET-DMPs: Euclidean transformation based dynamical movement primitives.

demonstration, it has to reach the same state as soon as possible. In addition, in the letter G writing task, we can find the problem of crossing between 0 and 2π . The angle is almost 0 at the end of the demonstration, but it changes from 0 to 2π as shown by the red-dashed line. This problem will have an effect on the generalized movement, as shown by the red solid line, where the end of the motion begins to fluctuate circled by the green-dashed ellipse. The same problem will appear in the spherical coordinate in 3D space. In general, the movement generalized by the DMPs in polar coordinate cannot give good results as we expect.

Verification of ET-DMPs with obstacle avoidance. We verify the obstacle avoidance based on ET-DMPs using the same demonstration data in the section “Verification of ET-DMPs,” namely, the trajectories of four kinds of letters. To test the obstacle avoidance of this method, single and multiple obstacles are added to disturb the current movement in the process of generalization. The position of obstacles is limited to the vicinity of expected trajectory to highlight their interference effect.

The solid line and the dotted line in Figure 7 are the generalized trajectories with and without obstacles,

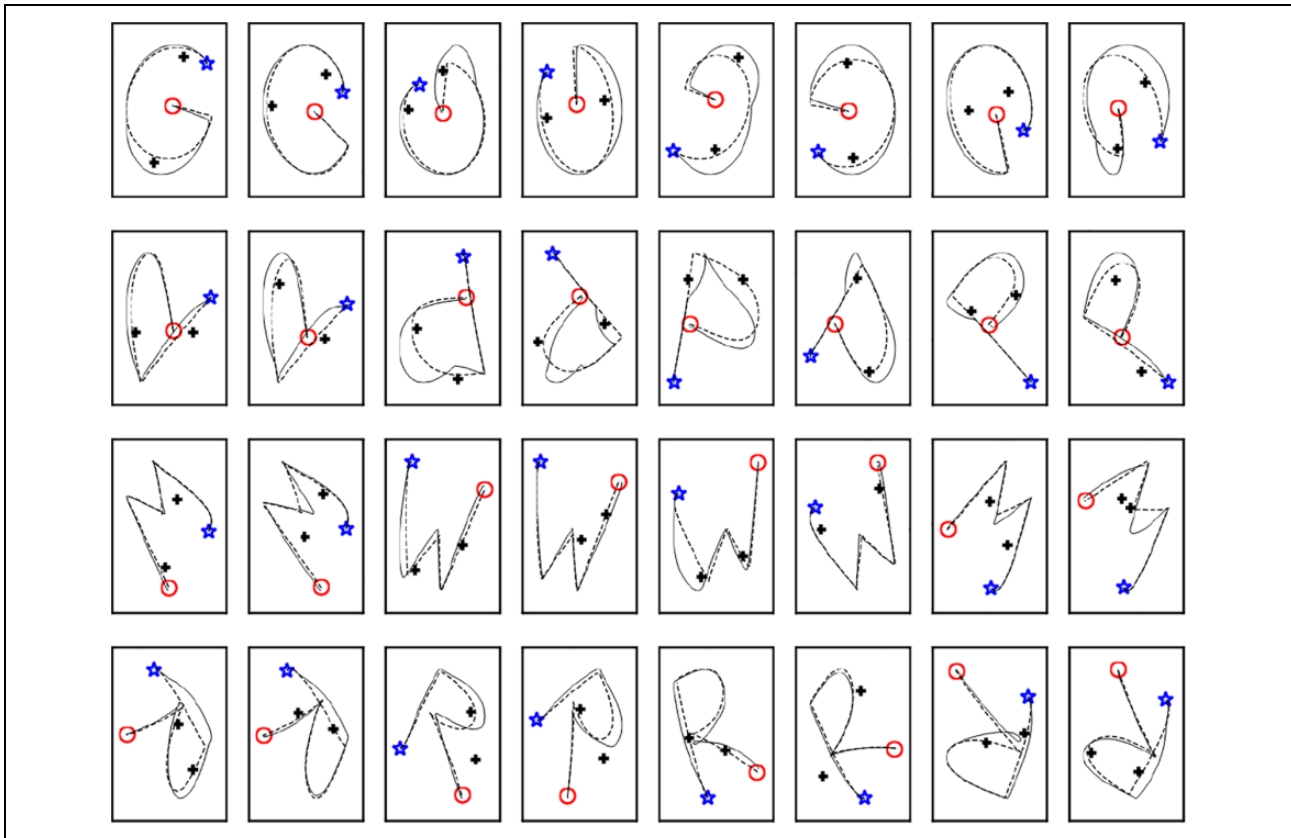


Figure 8. Trajectories generalized from ET-DMPs in multiple obstacle avoidance tasks. ET-DMPs: Euclidean transformation based dynamical movement primitives.

respectively. The whole process is almost the same as that in the section “Verification of ET-DMPs” except that obstacle transformation is added. The symbol “+” represents the obstacle. It can be proved that the trajectories obtained by ET-DMPs combined with obstacles information using Eq. (23), not only retains the shape of the demonstration but also avoids obstacles successfully. As shown in Figure 8, with multiple obstacles, the resultant force of two obstacles in some region makes the track deviate greatly, but the overall shape of movement can still meet the requirements of trajectory imitation.

Experiment

We evaluate the proposed method in several daily tasks. Firstly, in the writing task, we show that the validity of the ET-DMPs imitating the human writing trajectory with obstacle avoidance ability. Then, we show that the ET-DMPs could also imitate the periodic movement in the wiping whiteboard task. In 3D space, the robot learns to stir the mixture. The experiment environment is set as Figure 8. We select UR10 robot and the vision equipment RealSense D435 camera embedded in the robot system to detect goal and obstacles during the generalization process. The operation system is selected as Ubuntu 18.04 with ROS

Melodic. The setting of PC is CPU i7-10750 H 2.60 GHz and GPU RTX-2060.

Verification of ET-DMPs in writing task. For simplifying the demonstration process, we design a 2D simulation software to record the human writing behavior as shown in Figure 10 and then we scale the sampled demonstration data to proper size. We select two points in each quadrant as the robot initial state randomly as Figure 11(a) shown. We compare the writing letters G, P, R, and W trajectories generalized by the methods of ET-DMPs and DMPs. The whole process of our method has been mentioned in the section “Verification of ET-DMPs” and the data set is obtained from human demonstration. In Figure 12, we can see that the experiment results are not greatly different from simulation results. However, the letters are well imitated by ET-DMPs no matter the initial state is in which quadrants, so the proposed method has the imitation invariance of varying relative quadrants of the initial and goal states.

For quantifying the similarity of the imitation, we introduce the similarity evaluation method, which has been mentioned in the section “The principle of Euclidean transformation based DMPs” of trajectories. We show the mean and variance of the imitation error along the trajectories generalized from different quadrants in Figure 13; we can see that when relation between initial and goal states is the

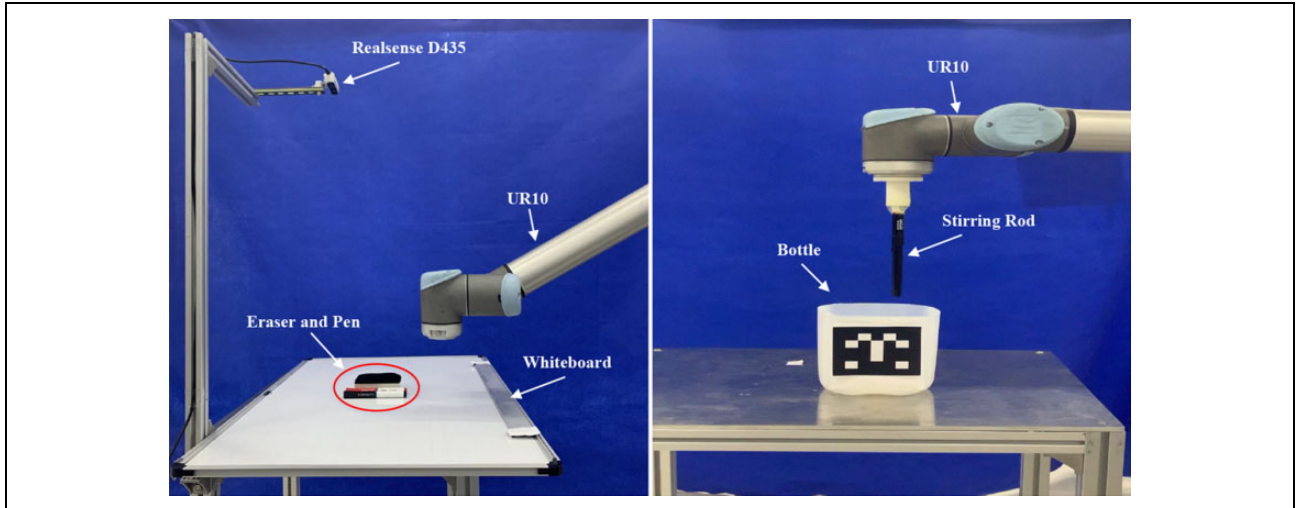


Figure 9. The setting of the 2D and 3D space task. 2D: two-dimensional; 3D: three-dimensional.

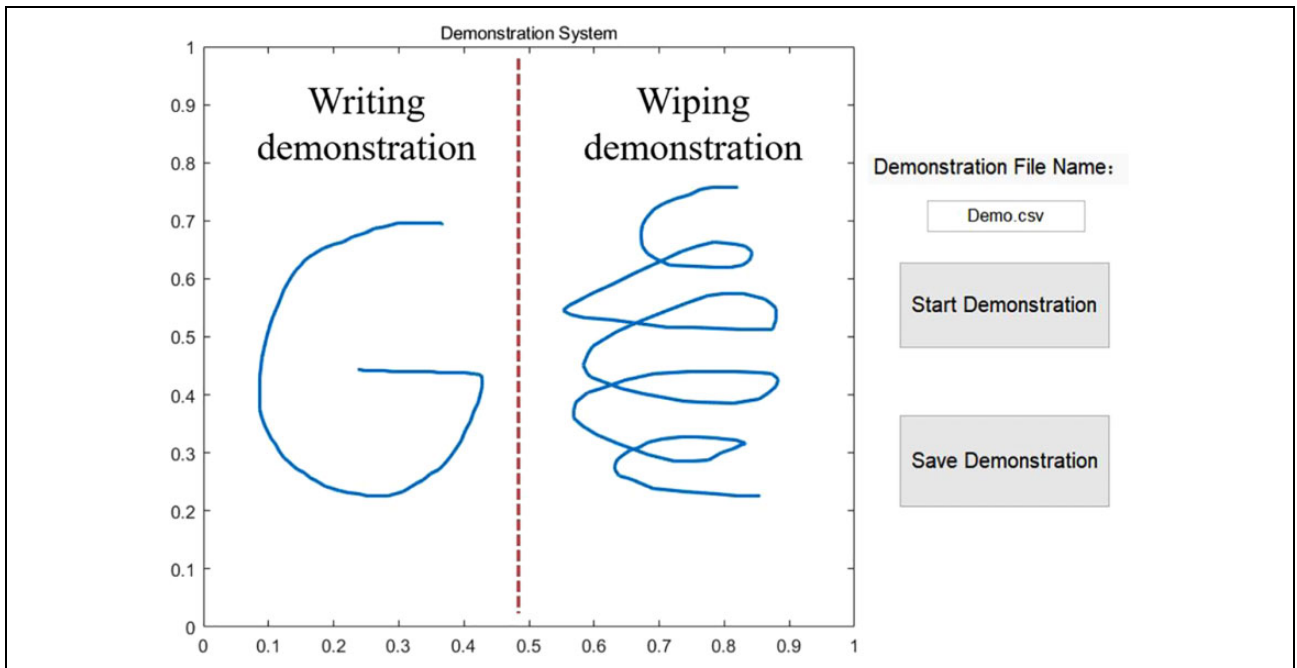


Figure 10. Demonstration software.

same as demonstration, the imitation error is the smallest among other cases. Besides, the evaluation indicators of ET-DMPs are smaller than DMPs in all conditions.

Verification of ET-DMPs in wiping whiteboard task. We design a task to verify that the proposed method could also imitate the periodic movement like wiping the whiteboard. Wiping whiteboard is a representative example because it involves the periodic motion. For the convenience of human demonstration, we use the same software as mentioned above to record the wiping trajectory as Figure 10 shown. As Figure 11(b) shows, firstly, we draw the handwriting to be wiped randomly in four quadrants and then select the

edge of the handwriting to be the initial wiping state while the goal state is set as the origin of the coordinate. We can conclude from Figures 14 and 15 that the trajectories generalized by DMPs (blue-dotted line) reached beyond the wiping area so that some parts of the handwriting cannot be wiped cleanly. However, when employing ET-DMPs, the wiping movement is limited strictly in the handwriting area as the green rectangle shown in Figure 11(b). Therefore, with the ET-DMPs, the robot can effectively complete the wiping task as well as get an excellent result.

Verification of ET-DMPs with obstacle avoidance in writing task. To test the performance of ET-DMPs in obstacle avoidance

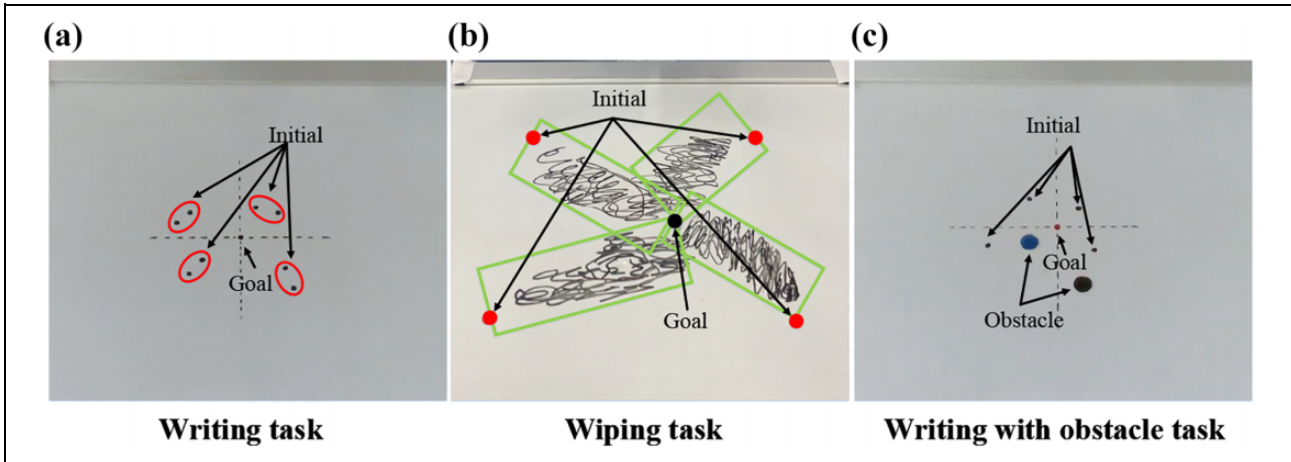


Figure 11. (a) to (c) Experiment initial states setting of 2D space task. 2D: two-dimensional.

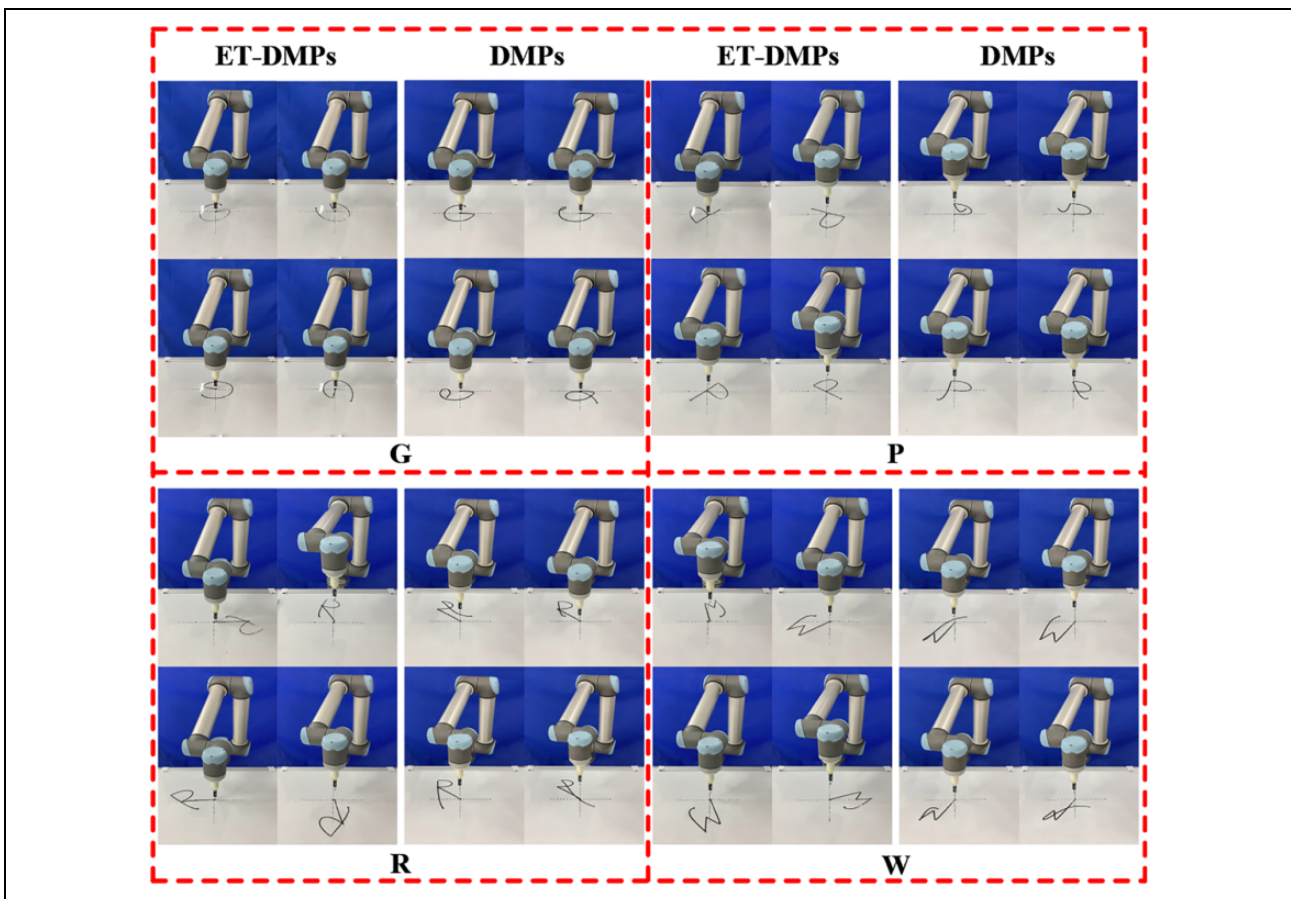


Figure 12. The writing trajectories of letter G, P, R, and W generalized by ET-DMPs and DMPs in four quadrants. ET-DMPs: Euclidean transformation based DMPs; DMPs: dynamical movement primitives.

task, we add two obstacles limited to the vicinity of the expected trajectory as Figure 11(c) shown. In Figure 16, we select two letters G and R, which are representative trajectories containing linear and curved motion. The first row shows the writing trajectories with obstacle disturbance, and on the contrary, the second row shows the trajectories

generalized without obstacles. As we can see, in the case of interference from two obstacles, the generalized motion has been blocked to track the expected motion, but the writing trajectories generalized by ET-DMPs can still retain the general shape feature of demonstration and avoid multiple obstacles successfully.

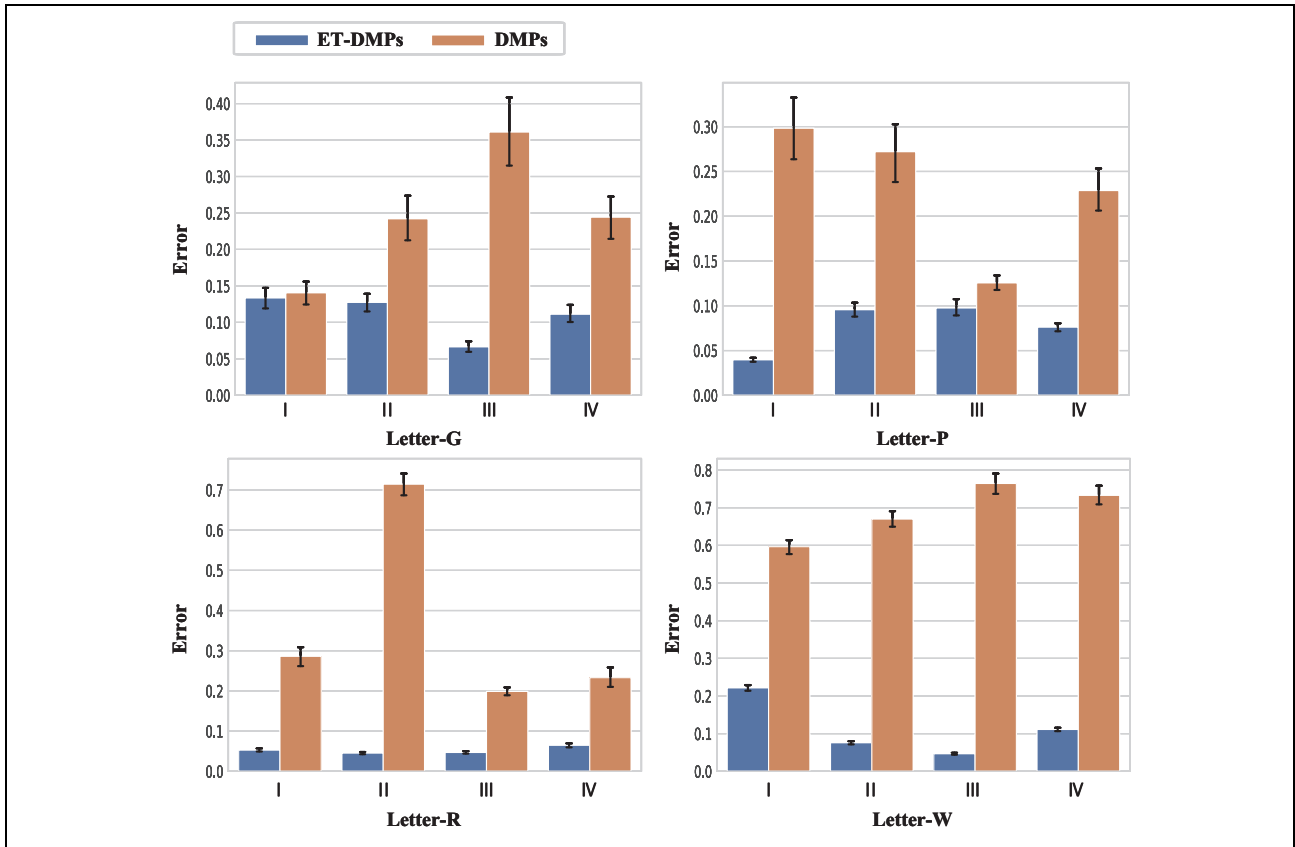


Figure 13. Imitation evaluation indicators of ET-DMPs and DMPs. ET-DMPs: Euclidean transformation based DMPs; DMPs: dynamical movement primitives.

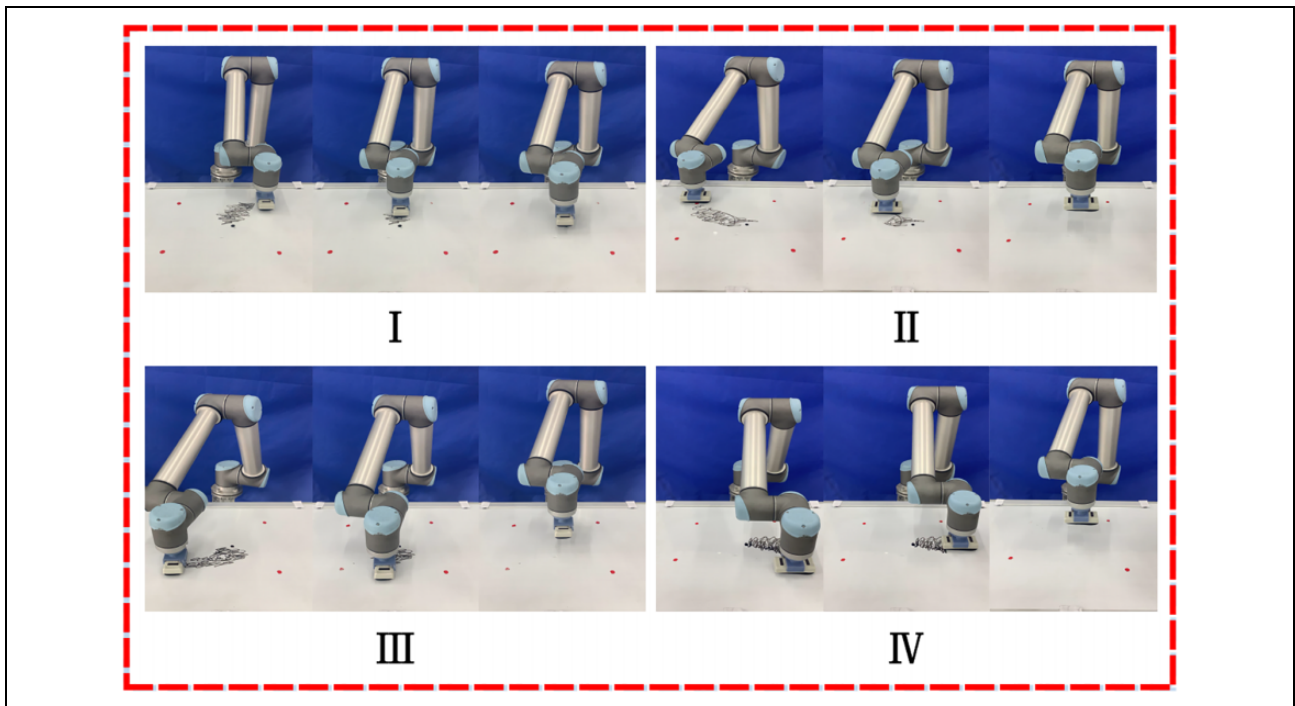


Figure 14. Snapshots of wiping whiteboard process using ET-DMPs. ET-DMPs: Euclidean transformation based dynamical movement primitives.

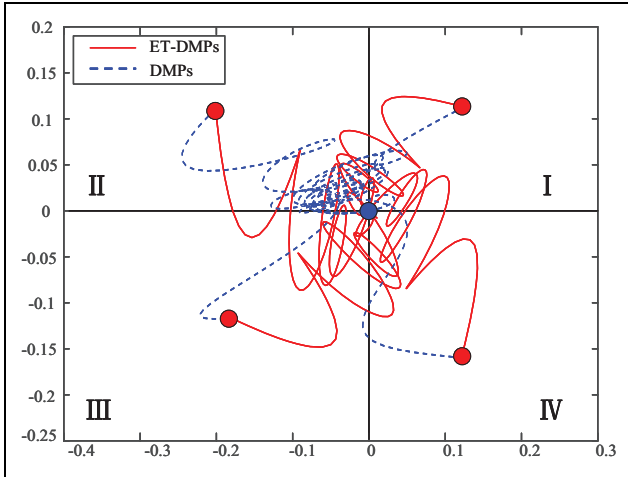


Figure 15. Wiping trajectories generalized by ET-DMPs and DMPs. ET-DMPs: Euclidean transformation based DMPs; DMPs: dynamical movement primitives.

Verification of ET-DMPs in stirring task. We employ ET-DMPs to accomplish tasks in 3D space, for example, transferring motions of stirring in bottle to robots to learn, we record the stirring trajectory through Kinesthetic teaching. The setting of the stirring task is shown in Figure 17. We select four positions close to the top of bottle in four quadrants as the initial states. Specifically, we still employ two modes mentioned above to manipulate in 3D space. In mode 1, the inclination of the central axis of bottle is considered into the rotation, which is determined by the direction from initial states to goal. In mode 2, movement along the z -axis is generalized using Eq. (7) independent of the other two dimensions while along the x -axis and y -axis movement is treated as the planar rotation using Eq. (8)

As shown in Figure 18, we compare the trajectories generalized by ET-DMPs (modes 1 and 2) and DMPs. We can conclude that when employing ET-DMPs, the stirring trajectories can always meet the requirements of the

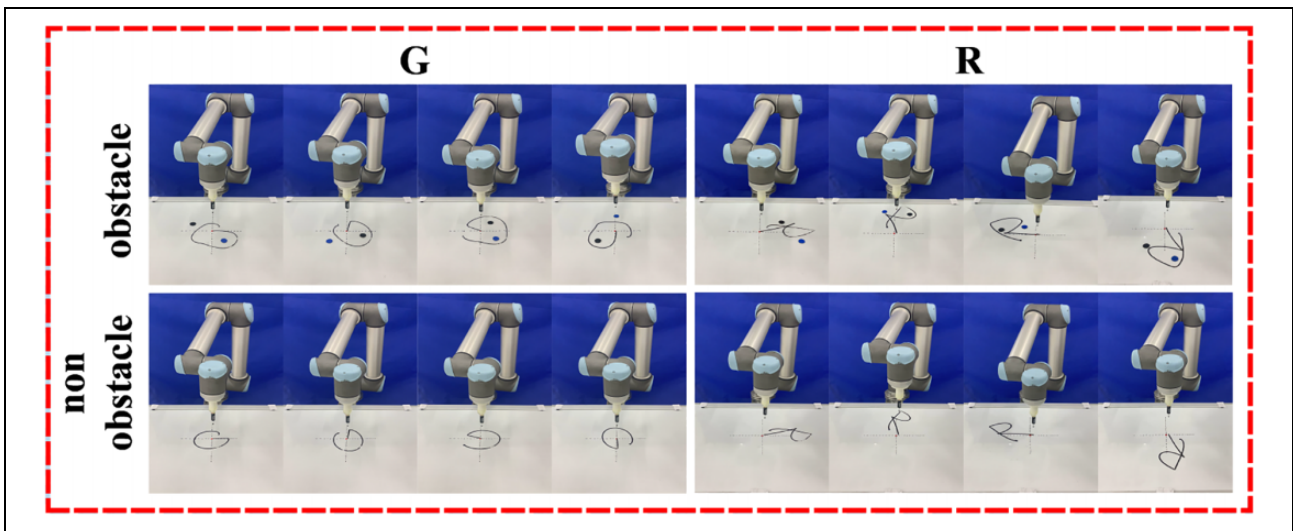


Figure 16. Writing letters G and R by ET-DMPs with obstacle avoidance. ET-DMPs: Euclidean transformation based dynamical movement primitives.

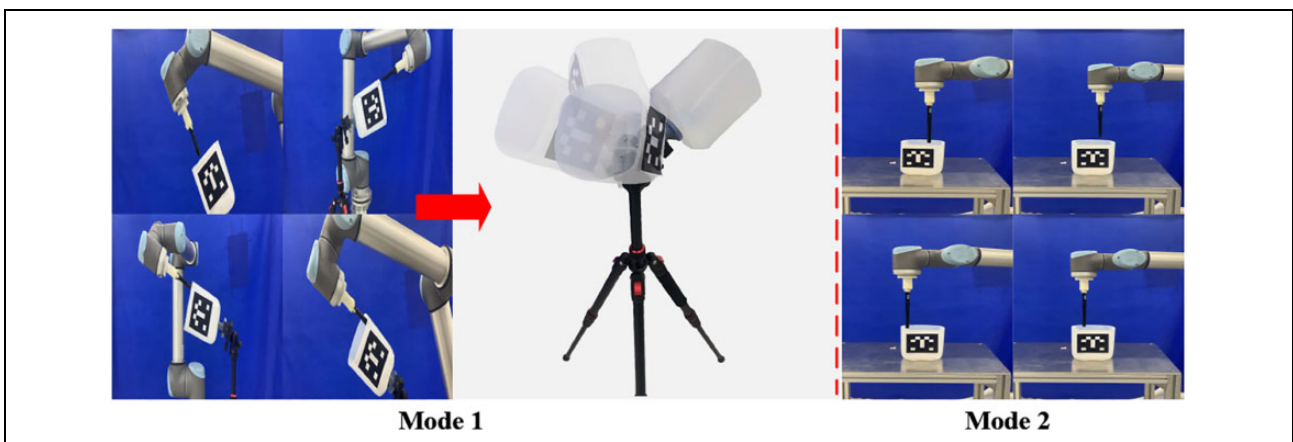


Figure 17. Initial states setting in stirring task.

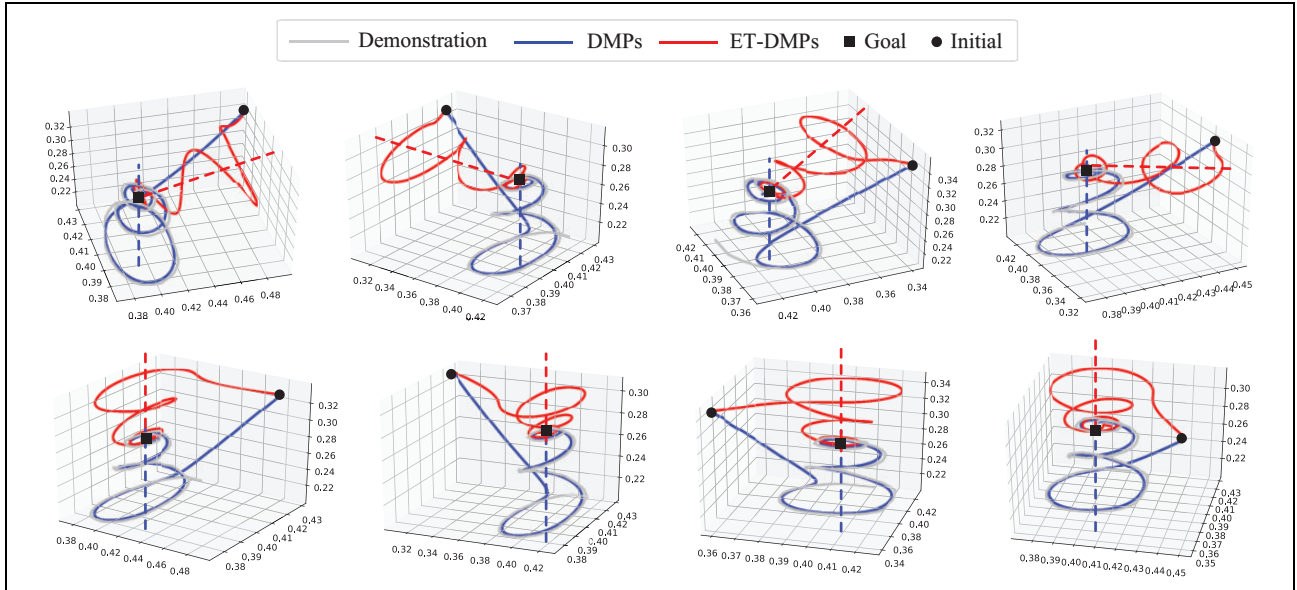


Figure 18. Stirling task trajectories generalized by ET-DMPs (mode 1/2) and DMPs. ET-DMPs: Euclidean transformation based DMPs; DMPs: dynamical movement primitives.

new task, either in mode 1 or in mode 2. However, in contrast to ET-DMPs, the shape feature nearly collapses when employing DMPs, for it only retains the independent feature of each dimension consistent with demonstration; meanwhile, the relation between the initial and goal states in different tasks are not taken into account in DMPs so it fails to generalize to other situations.

Discussion

This article introduces the Euclidean transformation matrix to modify the generalized part of DMPs. However, it is more than just rotation transformation. The generalization process of DMP is based on the proper relation state between the initial and goal. In different dimensions task, we discover many details about the transformation from this relation, such as 3D space, we divide it into two modes that meet the different task requirements. The ET-DMPs is a version of DMPs combined with such transformation, which solves the imitation problem caused by changes in the initial state.

Although there are important discoveries revealed by studies, it still has some limitations. The basic idea of ET-DMPs is based on offline movement generalization, so the relationship extracted from demonstration is fixed. If an unpredictable external disturbance during the generalized movement corrupts the initial relationship that we have extracted, the original ET-DMPs cannot immediately adjust the motion to compensate for the disturbance. In addition, our method does not consider the orientation generalization. The relationship of orientation maybe not as easy as position to obtained.

Our method can solve most situation problems as the experiment shown and it has been verified by adequate evaluation. However, maybe it is not the best way to solve this problem. In future research, we will try to employ this kind of transformation to the forcing term of DMPs, which may be the fundamental solution to the problem presented in this article. Then, we will try to optimize our method by introducing the real-time relationship between the current goal and robot state, which enables ET-DMPs to cope with application scenarios that change in real time. To imitate human behavior more completely, we will take the orientation transformation into account and plan to construct the relation model in the Riemannian manifold for the orientation part.

Conclusion

In this article, we proposed an ET-DMPs method to solve the problem that DMPs method cannot retain the shape feature of demonstration when the initial task state changes. It consists of Euclidean transformation module and generalization module. The first module is used to transform the robot state between the virtual situation and the real situation. The second module is used to imitate and generalize the movement. We tested our method in simulation and experiment. The complex letters such as G, P, R, and W are selected as database of demonstration in writing task and we demonstrated the wiping movement through a software designed for easily transferring the movement to the robot; finally, we also test our method in 3D space through stirring task. The results show that ET-DMPs retain the relative scope of the attractor field, which retains the shape feature of demonstration. Besides, the advantages of DMPs

are preserved in ET-DMPs. In addition, the Euclidean transformation of obstacles endows robots with the ability to avoid obstacles in more complex environment, and we employed this method in writing task to verify its validity. The results prove that ET-DMPs combined with obstacle transformation is capable of obstacle avoidance while retaining the characteristics of demonstration.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by the National Key Research and Development Program under Grant 2017YFB1303002 and the Major Research Plan of National Natural Science Foundation of China under Grant No. 91648201.

ORCID iD

Boyang Ti  <https://orcid.org/0000-0002-0303-8317>

References

- Zhu Z and Hu H. Robot learning from demonstration in robotic assembly: a survey. *Robotics* 2018; 7: 17.
- Calinon S and Billard A. Statistical learning by imitation of competing constraints in joint space and task space. *Adv Robot* 2009; 23: 2059–2076.
- Calinon S, Li Z, Alizadeh T, et al. Statistical dynamical systems for skills acquisition in humanoids. In: *2012 12th IEEE-RAS international conference on humanoid robots (Humanoids 2012)*, Osaka, Japan, 29 November, 1 December 2012, pp. 323–329. IEEE.
- Calinon S, Alizadeh T, and Caldwell DG. On improving the extrapolation capability of task-parameterized movement models. In: *2013 IEEE/RSJ international conference on intelligent robots and systems*, Tokyo, Japan, 3–7 November 2013, pp. 610–616. IEEE.
- Calinon S. A tutorial on task-parameterized movement learning and retrieval. *Intellig Serv Robot* 2016; 9: 1–29.
- Silvério J, Rozo L, Calinon S, et al. Learning bimanual end-effector poses from demonstrations using task-parameterized dynamical systems. In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Hamburg, Germany, 28 September, 3 October 2015, pp. 464–470. IEEE.
- Paraschos A, Daniel C, Peters J, et al. Probabilistic movement primitives. *Adv Neural Informat Process Syst* 2013; 2616–2624. <http://eprints.lincoln.ac.uk/id/eprint/25785/1/5177-probabilistic-movement-primitives.pdf>
- Paraschos A, Daniel C, Peters J, et al. Using probabilistic movement primitives in robotics. *Autonom Robot* 2018a; 42: 529–551.
- Paraschos A, Rueckert E, Peters J, et al. Probabilistic movement primitives under unknown system dynamics. *Adv Robot* 2018b; 32: 297–310.
- Paraschos A., Lioutikov R., , Peters J., , et al. Probabilistic prioritization of movement primitives. *IEEE Robot Automat Lett* 2017; 2: 2294–2301.
- Ijspeert AJ, Nakanishi J, and Schaal S. Movement imitation with nonlinear dynamical systems in humanoid robots. In: *Proceedings 2002 IEEE international conference on robotics and automation (Cat. No. 02CH37292)*, Washington, DC, USA, 11–15 May 2002, volume 2, pp. 1398–1403. IEEE.
- Schaal S. Dynamic movement primitives -A framework for motor control in humans and humanoid robotics. In: Kimura H, Tsuchiya K, Ishiguro A, and Witte H. (eds) *Adaptive motion of animals and machines*. Tokyo, Japan: Springer; 2006. https://doi.org/10.1007/4-431-31381-8_23
- Ijspeert AJ, Nakanishi J, Hoffmann H, et al. Dynamical movement primitives: learning attractor models for motor behaviors. *Neural Comput* 2013; 25: 328–373.
- Deniša M, Gams A, Ude A, et al. Learning compliant movement primitives through demonstration and statistical generalization. *IEEE/ASME Trans Mechatron* 2015; 21: 2581–2594.
- Ti B, Gao Y, Li Q, et al. Human intention understanding from multiple demonstrations and behavior generalization in dynamic movement primitives framework. *IEEE Access* 2019; 7: 36186–36194.
- Pervez A and Lee D. Learning task-parameterized dynamic movement primitives using mixture of GMMs. *Intellig Ser Robot* 2018; 11: 61–78.
- Su H, Mariani A, Ovrur SE, et al. Toward teaching by demonstration for robot-assisted minimally invasive surgery. *IEEE Trans Automat Sci Eng* 2021; 18: 484–494.
- Su H, Qi W, Hu Y, et al. An incremental learning framework for human-like redundancy optimization of anthropomorphic manipulators. *IEEE Trans Ind Informat* 2020.
- Qi W, Su H, and Aliverti A. A smartphone-based adaptive recognition and real-time monitoring system for human activities. *IEEE Trans Human-Mach Syst* 2020; 50: 414–423.
- Hoffmann H, Pastor P, Park D.-H, et al. Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance. In: *2009 IEEE international conference on robotics and automation*, Kobe, Japan, 12–17 May 2009, pp. 2587–2592. IEEE.
- Kulvicius T, Ning K, Tamosiunaite M, et al. Joining movement sequences: modified dynamic movement primitives for robotics applications exemplified on handwriting. *IEEE Trans Robot* 2011; 28: 145–157.
- Caccavale R, Saveriano M, Finzi A, et al. Kinesthetic teaching and attentional supervision of structured tasks in human–robot interaction. *Autonom Robot* 2019; 43: 1291–1307.
- Gräve K and Behnke S. Incremental action recognition and generalizing motion generation based on goal-directed features. In: *2012 IEEE/RSJ international conference on*

- intelligent robots and systems*, Vilamoura-Algarve, Portugal, 7–12 October 2012, pp. 751–757. IEEE.
24. Zhou Y, Do M, and Asfour T. Coordinate change dynamic movement primitives—a leader-follower approach. In: *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Daejeon, Korea (South), 9–14 October 2016, pp. 5481–5488. IEEE.
 25. Forte D, Gams A, Morimoto J, et al. On-line motion synthesis and adaptation using a trajectory database. *Robot Autonom Syst* 2012; 60: 1327–1339.
 26. Cohen Y and Berman S. Tight dynamic movement primitives for complex trajectory generation. In: *2013 IEEE international conference on systems, man, and cybernetics*, Washington, DC USA, 13–16 October 2013, pp. 2402–2407. IEEE.
 27. Khansari-Zadeh SM and Billard A. Learning stable nonlinear dynamical systems with Gaussian mixture models. *IEEE Trans Robot* 2011; 27: 943–957.
 28. Schaal S and Atkeson CG. Constructive incremental learning from only local information. *Neural Computat* 1998; 10: 2047–2084.
 29. Ude A, Gams A, Asfour T, et al. Task-specific generalization of discrete and periodic dynamic movement primitives. *IEEE Trans Robot* 2010; 26: 800–815.
 30. Piña E. Rotations with rodrigues’ vector. *Eur J Phys* 2011; 32: 1171.
 31. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. *Int J Rob Res* 1986; 5: 90–98.
 32. Fajen BR and Warren WH. Behavioral dynamics of steering, obstacle avoidance, and route selection. *J Exper Psychol: Human Percept Perform* 2003; 29: 343.
 33. Rai A, Meier F, Ijspeert A, et al. Learning coupling terms for obstacle avoidance. In: *2014 IEEE-RAS international conference on humanoid robots*, Madrid, Spain, 18–20 November 2014, pp. 512–518. IEEE.