SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Obstacle Avoidance in Human-Robot Cooperative Transportation with Force Constraint

Ying Zhang[1], Chenguang Yang[1*], Sheng Xu[2] & Yongsheng Ou[2]

[1]*College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China;*
[2]*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China*

Appendix A METHODOLOGY

Appendix A.1 Dynamic Movement Primitives (DMPs)

In this paper, two DMPs models are used to encode motion trajectory and force trajectory respectively, namely motion DMPs and force DMPs. According to the reference motion trajectory and force data obtained from the demonstration, DMPs for motion trajectories and force trajectories are learned respectively to obtain their respective model parameters. We demonstrate the effectiveness of the proposed method through an experiment of man-machine coordinated handling of boxes.

Appendix A.1.1 *DMPs for motion trajectory planning*

The essence of DMPs is a second order nonlinear dynamic system with spring and damper. The motion of a single degree of freedom can be expressed by the following formula [1]:

$$\tau\dot{\theta_2} = K(g-\theta_1) - D\theta_2 + h(s;\boldsymbol{\omega}) \tag{A1}$$

$$\tau\dot{\theta_1} = \theta_2 \tag{A2}$$

$$\tau\dot{s} = -\alpha_1 s \tag{A3}$$

where formula (A1) represents a transformation system with two parts: a second-order spring-damped system and a nonlinear term. K and D denote the spring constant and damping coefficient of the system respectively, and D is usually set as $K = D^2/4$ [2]. g is the target value of the motion trajectory. τ represents the time scaling constant. θ_1 and θ_2 respectively represent the position and velocity of motion, and the relationship between them is shown in formula (A2). $\boldsymbol{\omega}$ is the weight of the Gaussian model. s represents the phase variable of the system, which is determined by a canonical system, as shown in formula (A3), where α_1 is a positive constant. The nonlinear function is defined as

$$h(s;\boldsymbol{\omega}) = \frac{\sum_{i=1}^{N}\phi_i\omega_i}{\sum_{i=1}^{N}\phi_i}(g-\theta_0)s \tag{A4}$$

$$\phi_i = exp(-d_i(s-c_i)^2) \tag{A5}$$

where c_i and d_i are the center and width of the i-th Gaussian function respectively, θ_0 is the initial value of the motion trajectory, N is the number of Gaussian functions, w_i is the weight of the i-th Gaussian function. In general, the initial value of s was taken as 1, which decays over time and eventually goes to zero. Obviously, since the value of s tends to zero, the nonlinear function $h(s;\boldsymbol{\omega})$ is bounded, and the model becomes a stable second order spring-damped system.

In general, nonlinear regression algorithms such as local weighted regression algorithm (LWR) can be used to determine model parameters $\boldsymbol{\omega}$ [3]. Given the demonstration trajectory $\theta(t)$, where $t = [1, 2..., T]$, $g = \theta(T)$, the objective function can be determined according to formula (A1):

$$f_{target} = \tau\dot{\theta_2} - K(g-\theta_1) + D\theta_2 \tag{A6}$$

furthermore, $\boldsymbol{\omega}$ can be determined by the following formula:

$$\boldsymbol{\omega} = \underset{\boldsymbol{\omega}}{argmin}\sum(f_{target} - h(s;\boldsymbol{\omega}))^2 \tag{A7}$$

According to formula (A1), the original DMPs model are prone to generate large acceleration at the initial moment, which may cause damage to the robot. In order to avoid this problem, an exponential decay system in literature [1] was used to replace the moving target, that is, the current target point can be gradually approached from the initial point of the trajectory to the target point:

$$\tau\tilde{g} = -\alpha_2(\tilde{g} - g) \tag{A8}$$

where α_2 is a positive constant, $g = \theta(T)$ is a fixed constant, and $\tilde{g}$ starts from the initial value θ_0 and converges exponentially to g. According to formula (A8), the model can be transformed into:

$$\tau\dot{\theta_1} = \theta_2 \tag{A9}$$

$$\tau\dot{\theta_2} = K(\tilde{g}-\theta_1) - D\theta_2 + h(s;\boldsymbol{\omega}) \tag{A10}$$

* Corresponding author (email: cyang@ieee.org)

Appendix A.1.2 *DMPs for force trajectory planning*

Similar to the DMPs for motion trajectory planning, force trajectories of the demonstration are modeled by DMPs. Define the demonstration force as $f(t)$, where $t = [1, 2, ..., T]$, $g = f(T)$:

$$\tau \dot{f}_2 = M(\tilde{f}_g - f_1) - N f_2 + h(s; \boldsymbol{\beta}) \tag{A11}$$

$$\tau \dot{f}_1 = f_2 \tag{A12}$$

$$h(s; \boldsymbol{\beta}) = \frac{\sum_{i=1}^{N} \phi_i \beta_i}{\sum_{i=1}^{N} \phi_i} (f_g - f_0) s \tag{A13}$$

where M and N respectively represent the spring constant and damping coefficient of the system, and M is usually set as $M = N^2/4$ [1]; f_0 is the initial value of the force trajectory; f_1 and f_2 respectively represent the magnitude of the force and the speed of force change, and the relationship between them is shown in formula (A12); $f_g = f(T)$ is a fixed constant, and $\tilde{f}_g$ starts from the initial value f_0 and converges exponentially to f_g; β_i is the weight of the i-th Gaussian function. The two transformation systems are driven by the same phase variable s, such that they can be synchronized. The estimation process of the model parameters in the force DMPs are the same as that of the motion DMPs', as shown in formula (A6) and (A7).

Appendix A.2 Force Control

In the experiment of human-machine joint transportation of objects, the force exerted on the object needs to be constrained. In order to avoid object falling caused by small force or object distortion because of large force, the force received by the object during the reproduction process is expected to be maintained around the expected value. In this paper, a force controller is added on the basis of the position controller to regulate the manipulation force in the transportation process. The inputs of the force controller are the actual position and force, and the expected position and force, and the output is the desired acceleration. The expected movement of the robot in the interaction process is:

$$\boldsymbol{\Delta \ddot{X}} = \boldsymbol{K_p} \left(\boldsymbol{X_d} - \boldsymbol{X} \right) - \boldsymbol{K_v} \boldsymbol{\dot{X}} - \left(\boldsymbol{F_d} - \boldsymbol{F_r} \right) \tag{A14}$$

where,

$$\boldsymbol{K_p} = \begin{bmatrix} k_{px} & 0 & 0 \\ 0 & k_{py} & 0 \\ 0 & 0 & k_{pz} \end{bmatrix}, \boldsymbol{K_v} = \begin{bmatrix} k_{vx} & 0 & 0 \\ 0 & k_{vy} & 0 \\ 0 & 0 & k_{vz} \end{bmatrix} \tag{A15}$$

represent damping and stiffness respectively. By adjusting these two parameters, the force tracking performance of the robot end-effector can be changed. $\boldsymbol{X_d} = [x_d, y_d, z_d]^T \in R^{3\times 1}$ and $\boldsymbol{X} = [x, y, z]^T \in R^{3\times 1}$ are the expected trajectory and the actual trajectory respectively. $\boldsymbol{F_d} = [f_{d,x}, f_{d,y}, f_{d,z}]^T \in R^{3\times 1}$ and $\boldsymbol{F_r} = [f_x, f_y, f_z]^T \in R^{3\times 1}$ are the expected force and the actual force respectively. The output of the force controller is acceleration $\boldsymbol{\Delta \ddot{X}} = [\Delta\ddot{x}, \Delta\ddot{y}, \Delta\ddot{z}]^T \in R^{3\times 1}$, and the force can be adjusted by integrating $\boldsymbol{\Delta \ddot{X}}$ twice and adding to the desired position.

Appendix A.3 Obstacle Avoidance and Recovery

Humans and robots may encounter sudden obstacles in the process of jointly transporting objects. More specifically, obstacle avoidance needs to be considered, so that the position and orientation of the end-effector of the robot arm can track the expected trajectory when the robot meets an obstacle, and the redundancy of the robot can be used to avoid the obstacle. The solution of general inverse kinematics is [4]:

$$\boldsymbol{\dot{\theta}} = \boldsymbol{J^\dagger} \boldsymbol{\dot{x}} + \left(\boldsymbol{I} - \boldsymbol{J^\dagger J} \right) \boldsymbol{f} \tag{A16}$$

where $\boldsymbol{J^\dagger} = \boldsymbol{J^T} \left(\boldsymbol{J J^T} \right)^{-1}$, $\boldsymbol{f}$ is the vector for obstacle avoidance, and $\boldsymbol{I} \in R^{7\times 7}$ is the identity matrix, $\boldsymbol{\dot{\theta}} \in R^{7\times 1}$ is the angular velocity of each joint, $\boldsymbol{J} \in R^{7\times 7}$ is the Jacobian matrix, $\boldsymbol{x} \in R^{6\times 1}$ represents the position and Euler angle of the end-effector.

In order to ensure that the position and orientation of the manipulator can track the expected trajectory, a closed-loop control policy is added in Cartesian space:

$$\boldsymbol{\dot{x}_e} = \boldsymbol{\dot{\hat{x}}_d} + \boldsymbol{K_e e_x} = \boldsymbol{\dot{\hat{x}}_d} + \boldsymbol{K_e} \left(\boldsymbol{\hat{x}_d} - \boldsymbol{x_e} \right) \tag{A17}$$

where $\boldsymbol{\dot{x}_e}$ is the velocity of the end of the arm, $\boldsymbol{\hat{x}_d} = \boldsymbol{x_d} + \boldsymbol{\Delta x}$ is the desired position input to the position controller, $\boldsymbol{x_d} = [x, y, z, w_x, w_y, w_z]^T \in R^{6\times 1}$ is the expected trajectory generated from the DMPs, including the position and Euler angle of the end-effector, $\boldsymbol{\Delta x}$ is the output of the force controller, $\boldsymbol{x_e} \in R^{6\times 1}$ represents the actual position and Euler angle of the end-effector. In this paper, the orientation of the end-effector is expected to be always perpendicular to the surface of the object during the transportation of the object. Consequently, the orientation of the end-effector is constant during the experiment. $\boldsymbol{K_e}$ is a symmetric positive definite matrix. When an obstacle approaches close to the arm, the arm's joints can be moved away from the obstacle. The velocity of the end-effector must satisfy the motion constraint during obstacle avoidance:

$$\boldsymbol{\dot{x}_e} = \boldsymbol{J_e} \boldsymbol{\dot{\theta}} \tag{A18}$$

and

$$\boldsymbol{\dot{x}_o} = \boldsymbol{J_o} \boldsymbol{\dot{\theta}} \tag{A19}$$

where $\boldsymbol{J_o} \in R^{7\times 7}$ is the Jacobian matrix of the collision point, $\boldsymbol{J_e} \in R^{7\times 7}$ is the Jacobian matrix at the end-effector, and $\boldsymbol{\dot{x}_o}$ is obstacle avoidance speed.

In the process of obstacle avoidance, the distance between the obstacle and the robotic arm need to be calculated, and then whether obstacle avoidance is needed be determined. Here, we refer to the obstacle detection method in literature [5]. Firstly, the continuous K-means clustering method is applied to segment the point cloud obtained by Kinect into hyperpixels. Then, the segmented point clouds are used to generate a simplified 3D model of the robot. In the point cloud, points near the 3D model are recognized as obstacles. The distance between the obstacle and the robot arm can be obtained by calculating the minimum

Figure A1 The motion of each joint angle during obstacle avoidance.

distance between the simplified three-dimensional model and the object. Assuming that the coordinates of the obstacle are $\boldsymbol{p_{obj}}$, and the coordinates of the closest point to the obstacle on the mechanical arm are $\boldsymbol{p_r}$, as shown in Figure A1. The distance between the obstacle and the robotic arm is defined as $L = \|\boldsymbol{p_r} - \boldsymbol{p_{obj}}\|$. The end-effector moving speed of the robot arm is determined according to the distance between the obstacle and the robot arm. Besides, the maximum and minimum threshold of obstacle avoidance distance is set as L_{max} and L_{min}. When $L > L_{max}$, the obstacle is far away from the robot arm, and thus the obstacle avoidance is not needed, i.e. $\dot{\boldsymbol{x}}_{\boldsymbol{o}} = \boldsymbol{0}$. When $L_{min} < L < L_{max}$, the obstacle avoidance speed increases gradually with the decrease of distance, and we set $\dot{\boldsymbol{x}}_{\boldsymbol{o}} = h(d)\boldsymbol{v_{max}}$, where $h(d) = (L_{\max} - L)/(L_{\max} - L_{\min})$. To keep the arm moving away from obstacles, we set $\boldsymbol{v_{max}} = v_{\max}(\boldsymbol{p_{obj}} - \boldsymbol{p_r})/L$ as the maximum speed of obstacle avoidance. When $L < L_{min}$, the manipulator avoids the obstacle will be the maximum speed $\boldsymbol{v_{max}}$ [6].

However, when the collision point is too close to the base of the robot arm, the robot may not have enough degrees of freedom to achieve the obstacle avoidance speed $\dot{x_0}$, for example, when the rank of J_0 is 2, since $\dot{x_0}$ is three-dimensional, the formula (A19) has no solution. However, there is always another possible obstacle avoidance velocity $\dot{x_0}'$, such that the vector $\dot{x_0}' - \dot{x_0}$ falls on the normal plane N of $\dot{x_0}$, and $\dot{x_0}$ is the projection of $\dot{x_0}'$ on the plane N. Therefore, the obstacle avoidance speed $\dot{x_0}'$ will play a similar role to $\dot{x_0}$, making the robot arm away from the coming obstacle. And the relation between $\dot{x_0}$ and $\dot{x_0}'$ satisfies the formula:

$$\dot{\boldsymbol{x}}_o^T \left(\dot{\boldsymbol{x}}_o' - \dot{\boldsymbol{x}}_o\right) = 0 \tag{A20}$$

substitute $\dot{\boldsymbol{x}}_o' = \boldsymbol{J_o}\dot{\boldsymbol{\theta}}$ into formula (A20) we can get:

$$\dot{\boldsymbol{x}}_o^T \dot{\boldsymbol{x}}_o = \dot{\boldsymbol{x}}_o^T \boldsymbol{J_o}\dot{\boldsymbol{\theta}} \tag{A21}$$

since formula (A21) is a scalar equation, it is solvable even if the rank of J is reduced to 1.

When the obstacle is removed or avoided, the arm needs to return to the expected trajectory. Therefore, a parallel system is designed in the controller to restore the position of the manipulator when there is no obstacle. As shown in Figure A1, the posture of the robotic arm is mainly determined by the three joints S_1, E_1 and W_1. When there are no obstacle, the joints must meet the motion constraints:

$$\begin{bmatrix} \boldsymbol{J_e} \\ \boldsymbol{J_r} \end{bmatrix} \dot{\boldsymbol{\theta}} = \begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{e}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{r}} \end{bmatrix} \tag{A22}$$

where $\boldsymbol{J_r} = \left[\boldsymbol{J_{S_1}}, \boldsymbol{J_{E_1}}, \boldsymbol{J_{W_1}}\right]^T$ are Jacobian matrices of joint S_1, E_1, W_1 respectively. $\dot{\boldsymbol{x}}_{\boldsymbol{r}} = \left[\dot{\boldsymbol{x}}_{\boldsymbol{S_1}}, \dot{\boldsymbol{x}}_{\boldsymbol{E_1}}, \dot{\boldsymbol{x}}_{\boldsymbol{W_1}}\right]^T$ represents the speed at the joint S_1, E_1, W_1 returning to the original state.

The arm returns to its original position at a speed of:

$$\dot{\boldsymbol{x}}_{\boldsymbol{r}} = \boldsymbol{K_r}\Delta\boldsymbol{\theta_r} \tag{A23}$$

where $\boldsymbol{K_r}$ is a symmetric positive definite matrix; $\boldsymbol{\Delta\theta_r} = [\Delta\theta_{S1}, \Delta\theta_{E1}, \Delta\theta_{W1}]^T \in R^{3\times 1}$ is the error between the actual and expected joint angles.

By substituting (A19) and (A23) into the general solution of inverse kinematics (A16), we obtain

$$\boldsymbol{J_o}\boldsymbol{J_e^\dagger}\dot{\boldsymbol{x}}_{\boldsymbol{e}} + \boldsymbol{J_o}\left(\boldsymbol{I} - \boldsymbol{J_e^\dagger}\boldsymbol{J_e}\right)\boldsymbol{f_o} = \dot{\boldsymbol{x}}_{\boldsymbol{o}} \tag{A24}$$

$$\boldsymbol{J_r}\boldsymbol{J_e^\dagger}\dot{\boldsymbol{x}}_{\boldsymbol{e}} + \boldsymbol{J_r}\left(\boldsymbol{I} - \boldsymbol{J_e^\dagger}\boldsymbol{J_e}\right)\boldsymbol{f_r} = \dot{\boldsymbol{x}}_{\boldsymbol{r}}, \tag{A25}$$

where $\boldsymbol{J_e^\dagger} = \boldsymbol{J}^T\left(\boldsymbol{J}\boldsymbol{J}^T\right)^{-1}$. based on formulas (A24) and (A25) we can find solutions for f_o and f_r:

$$\boldsymbol{f_o} = [\boldsymbol{J_o}\left(\boldsymbol{I} - \boldsymbol{J_e^\dagger}\boldsymbol{J_e}\right)]^\dagger(\dot{\boldsymbol{x}}_{\boldsymbol{o}} - \boldsymbol{J_o}\boldsymbol{J_e^\dagger}\dot{\boldsymbol{x}}_{\boldsymbol{e}}) \tag{A26}$$

$$\boldsymbol{f_r} = [\boldsymbol{J_r}\left(\boldsymbol{I} - \boldsymbol{J_e^\dagger}\boldsymbol{J_e}\right)]^\dagger(\dot{\boldsymbol{x}}_{\boldsymbol{r}} - \boldsymbol{J_r}\boldsymbol{J_e^\dagger}\dot{\boldsymbol{x}}_{\boldsymbol{e}}). \tag{A27}$$

In order to ensure the smooth transition between obstacle avoidance and recovery, the weighted sum of f_o and f_r is used to replace the terms in the general solution (A16) of inverse kinematics. Then, we acquire

$$\dot{\boldsymbol{\theta}}_{\boldsymbol{d}} = \boldsymbol{J_e^\dagger}\dot{\boldsymbol{x}}_{\boldsymbol{e}} + \left(\boldsymbol{I} - \boldsymbol{J_e^\dagger}\boldsymbol{J_e}\right)[(1-\beta)\boldsymbol{f_o} + \beta\boldsymbol{f_r}] \tag{A28}$$

where β is a piecewise function:

$$\beta = \begin{cases} 1 & L \geqslant L_{\max} \\ \frac{L_{\min} - L}{L_{\max} - L_{\min}} & L_{\min} < L < L_{\max} \\ 0 & L \leqslant L_{\min} \end{cases}. \tag{A29}$$

Therefore, in obstacle avoidance mode, the desired joint angle of the manipulator can be expressed as:

$$\boldsymbol{\theta_d} = \boldsymbol{\theta_e} + \dot{\boldsymbol{\theta}}_{\boldsymbol{d}}. \tag{A30}$$

Appendix A.4 Stability Analysis

In the paper, a simple PID controller was used for the position controller, which is stable. A force controller was added to the original control ring, the addition of the force controller does not affect the stability of the original position controller. Next, we prove the stability of the obstacle avoidance algorithm. Consider the required velocity $\dot{\boldsymbol{\theta}}_{\boldsymbol{d}}$ as defined in (A28). If the joint velocity $\dot{\boldsymbol{\theta}}$ completely follows $\dot{\boldsymbol{\theta}}_{\boldsymbol{d}}$, then the end-effector position error $\boldsymbol{e_x}$ will converge asymptotically to zero. According to [7], we proven that when joint angular velocity $\dot{\boldsymbol{\theta}}$ completely tracks the expected joint angular velocity $\dot{\boldsymbol{\theta}}_{\boldsymbol{d}}$, and the end-pose error $\boldsymbol{e_x} = \hat{\boldsymbol{x}}_{\boldsymbol{d}} - \boldsymbol{x_e}$ will tend to 0. Let the Lyapunov function be:

$$V = \frac{1}{2}\boldsymbol{e_x^T e_x} \tag{A31}$$

then:

$$\begin{aligned} \dot{V} &= \boldsymbol{e_x^T}\dot{\boldsymbol{e}}_{\boldsymbol{x}} \\ &= \boldsymbol{e_x^T}(\dot{\hat{\boldsymbol{x}}}_{\boldsymbol{d}} - \dot{\boldsymbol{x}}_{\boldsymbol{e}}) + (\boldsymbol{I} - \boldsymbol{J_e^\dagger J_e})[(1-\beta)\boldsymbol{f_o} + \beta\boldsymbol{f_r}]] \\ &= \boldsymbol{e_x^T}\dot{\hat{\boldsymbol{x}}}_{\boldsymbol{d}} - \boldsymbol{e_x^T J_e}[\boldsymbol{J_e^\dagger}(\dot{\boldsymbol{x}}_{\boldsymbol{d}} + \boldsymbol{e_x}) + (\boldsymbol{I} - \boldsymbol{J_e^\dagger J_e})[(1-\beta)\boldsymbol{f_0} + \beta\boldsymbol{f_r}]] \\ &= -\boldsymbol{e_x^T J_e J_e^\dagger e_x} - (\boldsymbol{e_x^T J_e} - \boldsymbol{e_x^T J_e J_e^\dagger J_e})[(1-\beta)\boldsymbol{f_o} + \beta\boldsymbol{f_r}] \\ &= -\boldsymbol{e_x^T J_e J_e^\dagger e_x} \end{aligned} \tag{A32}$$

Since $\boldsymbol{J_e J_e^\dagger} = \boldsymbol{J_e J_e^T}\left(\boldsymbol{J_e J_e^T}\right)^{-1} = \boldsymbol{I}$, we get $\dot{V} = -\boldsymbol{e_x^T J_e J_e^\dagger e_x} \leqslant 0$. According to Lyapunov theorem [7], the position and orientation errors of the manipulator end-effector will converge to 0 as time increases.

Appendix B Experimental Verification

The developed control scheme is verified by the Baxter robot. A force sensor is added at the end-effector of the arm to measure the force on the object. During the experiment, the sampling interval of the force sensor is 0.02s. The robot manipulator works together with a Kinect sensor to detect obstacles in the surrounding environment. The experimental process consists of three basic stages. We demonstrate the effectiveness of the proposed method through an experiment of man-machine coordinated handling of boxes. The whole experiment is introduced in three parts. The first part is the teaching part, in which the teacher drags the robot arm to perform the task and records the motion trajectory data and force data at the same time. Then the teaching trajectories are modeled by DMP model. When the parameters of the task change, DMPs models are used to generalize the trajectory to get the trajectories of the new task, and the robot can perform the new task through the hybrid control of force and position. The following is a detailed introduction to each part.

In the built-in kinematics, the teaching mode is provided by the robot manufacturer, and the robot first learns the trajectory from the initial point to the object, which is defined as the "Move" stage. Then, one person holds an object (box) with the robot end-effector, and the other person, drags the arm of the Baxter robot. Next, we repeat the carrying box task four times, as shown in Fiure B1. This is defined as the "Carry" phase. During the demonstration, position data and force data of the end-effector are recorded.

In order to learn the trajectory characteristics from multiple demonstration data, GMR is used to fit the four motion trajectory data and four force data recorded in the demonstration, as shown in Figure B2. Then, DMPs models are used to model the motion trajectory and force trajectory obtained by GMR fitting, and the model parameters are estimated.

In the process of reproducing the above transport task, one person holds the box with the robot end-effector, as shown in Figure B1. The manipulator moves according to the characteristics of the learned trajectory. At the same time, the proposed force control method is used during the transportation to maintain the expected manipulation force.

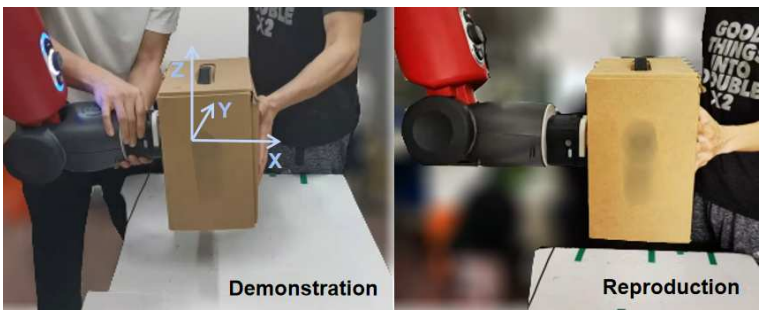


Figure B1 Demonstration and reproduction process.

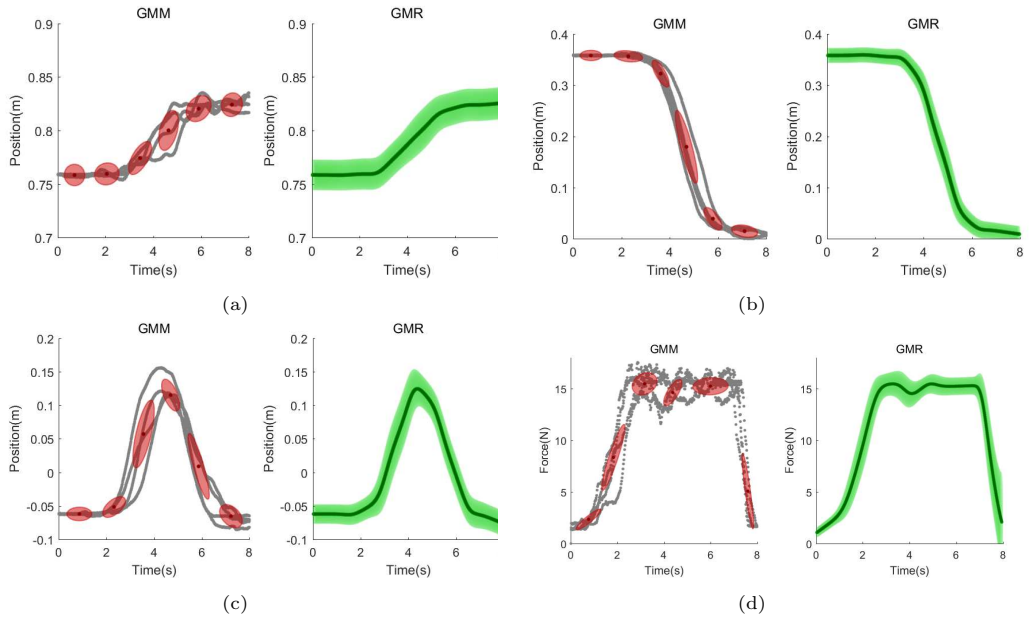


Figure B2 GMR is used to fit several training trajectories. (a) (b) (c) are respectively the motion trajectories of the end of the manipulator on the X, Y, Z axes, (d) is the force trajectory of the object. Trajectories (green lines) are generated based on the demonstration trajectories (gray dots).

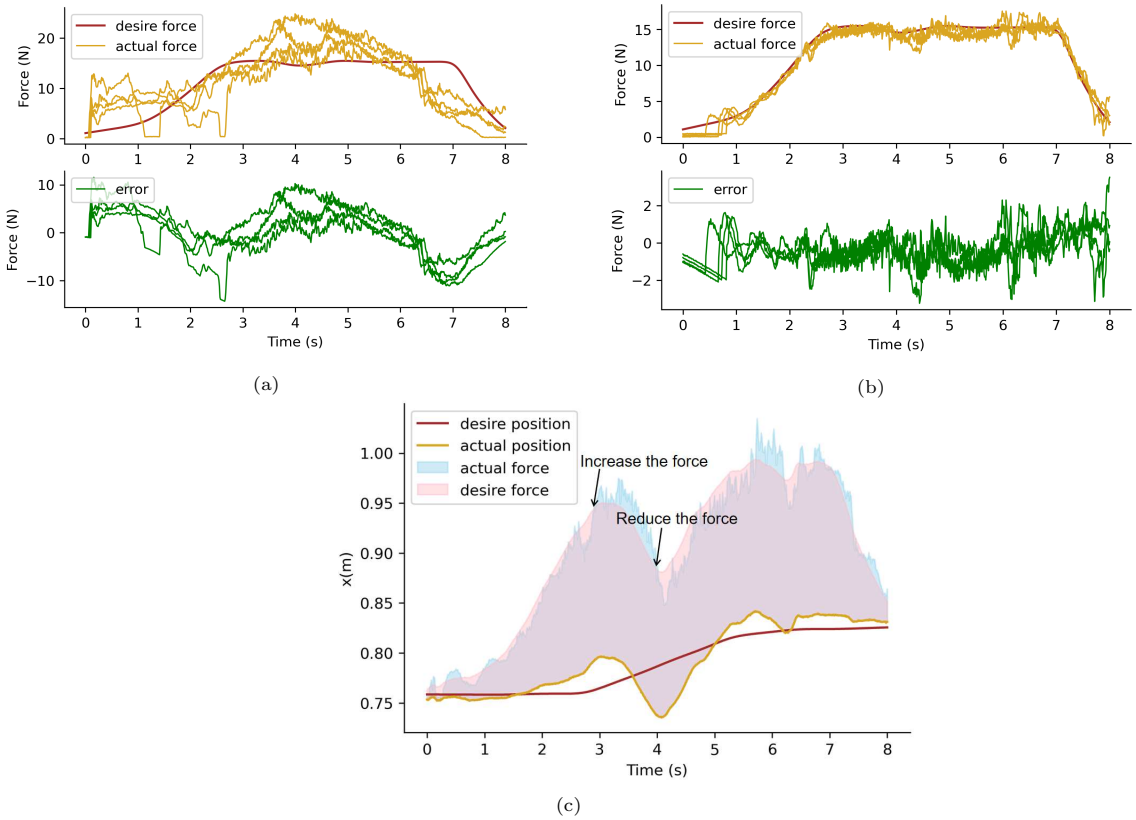


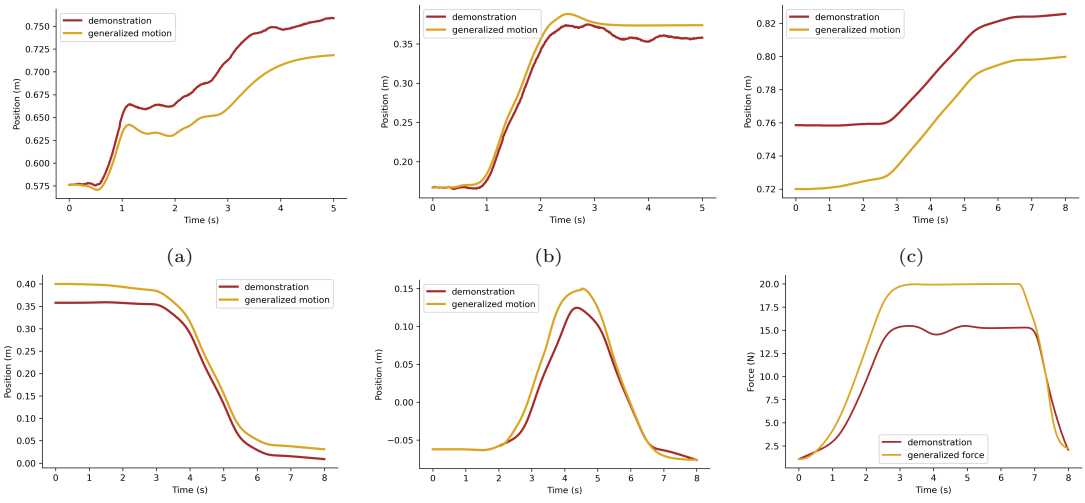
Figure C1 The results of the experimental comparison show that (a) the force on the object is deviated from the expected force when the robot only adopts position control. (b) By introducing the contact force information into the robot control, the actual force can be constrained within a certain range. (c) The manipulator can adapt to human interference with the force controller, the height of the shaded area is the value of the force.

Appendix C Results and Discussion

In the reproduction phase, firstly, a set of comparative experiments is designed to verify the effectiveness of the hybrid force/motion control method. In the first experiment, no force controller is added, only position control is carried out. Four people are randomly selected to complete the task of carrying objects with the robot. In the second experiment, a force controller is added, such that the robotic arm can adjust the force received by the object during the process of transporting the object. The experimental results are shown in Figure C1 (a) and Figure C1 (b). It can be seen that in the four experiments, when the force controller is not added, the error between the actual force and the expected force reaches 10N, which is very unfavorable for transporting the fragile objects. While after the application of the force controller, the error between the actual and the expected forces can be controlled within 3N, which ensures that robot end-effector do not cause damages to the object during transportation. Besides, with the force controller, the users can conduct online intervention on the movement of the robot arm during transportation, and fine-tune the movement trajectory of the robot by changing the magnitude of the force applied on the object. Figure C1 (c) shows that the robotic arm can adapt to human changes when human interference is exerted on objects during transportation. As shown in Figure C1 (c), when operators increase the force exerted on the object, the mechanical arm will move away from the object, and the force exerted on the object will reduce. On the contrary, when people decrease the force exerted on the object, the mechanical arm will move in the direction of the object, and the force on the object will increase close to the expected value.

(a) (b)

Figure C2 (a) New task situation: starting point, ending point and the weight of objects are changed. (b) Avoid obstacles in the process of man-machine cooperative object transportation.

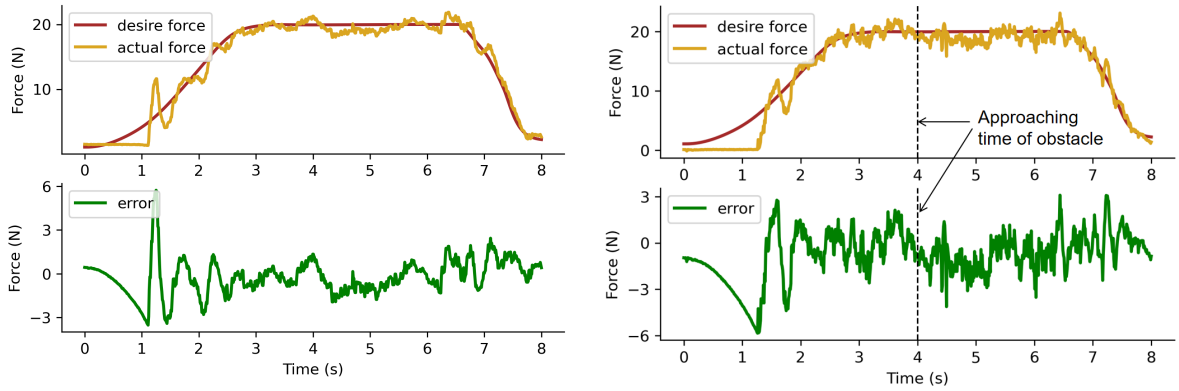


(a) (b) (c)

(d) (e) (f)

Figure C3 DMPs are used to generalize the demonstration trajectories. (a) and (b) are respectively the generalization of X and Y axes' trajectories in the "Move" stage, while the trajectories in the Z axis remain constant. (c), (d) and (e) are respectively the generalization of X, Y and Z axes' trajectories in the "Carry" stage. And (f) is the generalization of the force on the object at the "Carry" stage

In the generalization phase, a new example is designed to verify the generalization of proposed method. Specifically, the initial and target positions of the box are changed to the new locations, and the weight of the object is also different with the previous example, as shown in Figure C2 (a). In the "Move" phase, seven DMPs are used. While in the "Carry" stage, four DMPs are used to model the end position of the manipulator and the detected force. All the DMPs models parameters are set as: $D = N = 25$, $K = M = 25^2/4$, $\tau = 1$, $N = 40$. The experimental results are shown in Figure C3. It can be seen in Figure C4 (a) that after the mentioned elements are changed, the human and robot can still complete the task, and the error between actual and expected forces in the process of transporting the object can be still controlled within 3N.



(a) (b)

Figure C4 (a) The force exerted on an object during transport when the initial and target points of the transport trajectory, and the weight of the object are changed. (b) The force exerted on the object in the presence of an obstacle.

In addition, another experiment of a human-robot transportation task in the presence of obstacles is provided to verify the proposed strategy. When an obstacle occurs near the robot arm, the robot arm needs to change the joint angles to implement a avoidance behavior. Meanwhile, the end position and orientation should track the desired states to ensure the smooth completion of the task. In the process of cooperative object handling, obstacles were randomly placed near the robot arm to verify the obstacle avoidance function, as shown in Figure C2 (b). In this experiment, in order to ensure that the robot can still complete the task smoothly in the case of obstacles, robot redundancy is used to change the joint angle so that the robot arm can stay away from obstacles. When the obstacle is close to the manipulator randomly, the manipulator can avoid the obstacle and complete the task smoothly without deviating from the desired trajectory. Experimental results demonstrate the effectiveness of the proposed framework. At the same time, force constraints are also satisfied in the process of obstacle avoidance. The changes of actual and expected forces during obstacle avoidance are shown in Figure C4 (b). The obstacle starts to approach the manipulator arm at 4s and then leaves the manipulator arm. During the approach of the obstacle, the manipulator uses redundancy to change joint angles to avoid the obstacle. It can be seen from Figure C4 (b) that the end-effector makes slight changes during obstacle avoidance which leads to an error increase between the force exerted on the object and the expected value. We can find that the error between actual and expected forces can be controlled within 5N during obstacle avoidance, which is acceptable for the transported objects.

References

1 C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," IEEE transactions on neural networks and learning systems, vol. 30, no. 3, pp. 777-787, 2018.
2 P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in 2009 IEEE International Conference on Robotics and Automation, 2009, pp. 763-768.
3 C. E. Reiley, E. Plaku, and G. D. Hager, "Motion generation of robotic surgical tasks: Learning from expert demonstrations," in 2010 Annual international conference of the IEEE engineering in medicine and biology. IEEE, 2010, pp. 967-970.
4 A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," The international journal of robotics research, vol. 4, no. 3, pp. 109-117, 1985.
5 X. Wang, C. Yang, Z. Ju, H. Ma, and M. Fu, "Robot manipulator self-identification for surrounding obstacle detection," Multimedia Tools and Applications, vol. 76, no. 5, pp. 6495-6520, Mar. 2017.
6 C. Yang, X. Wang, L. Cheng, and H. Ma, "Neural-learning-based telerobot control with guaranteed performance," IEEE transactions on cybernetics, vol. 47, no. 10, pp. 3148-3159, 2016.
7 H. Qiao, M. Wang, J. Su, S. Jia, and R. Li, "The concept of attractive region in environment and its application in high-precision tasks with low-precision systems," IEEE/ASME Transactions on Mechatronics, vol. 20, no. 5, pp. 2311-2327, 2015.