


RESEARCH ARTICLE

A constrained framework based on IBLF for robot learning with human supervision

Donghao Shi¹ , Qinchuan Li¹, Chenguang Yang² and Zhenyu Lu²

¹School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou, China and ²Bristol Robotics Laboratory, University of the West of England, Bristol, UK

Corresponding author: Qinchuan Li; Email: lqchuan@zstu.edu.cn

Received: 26 August 2022; **Revised:** 17 December 2022; **Accepted:** 28 March 2023; **First published online:** 24 April 2023

Keywords: dynamic movement primitives; robotic skill learning; integral barrier Lyapunov function

Abstract

Dynamical movement primitives (DMPs) method is a useful tool for efficient robotic skills learning from human demonstrations. However, the DMPs method should know the specified constraints of tasks in advance. One flexible solution is to introduce the human superior experience as part of input. In this paper, we propose a framework for robot learning based on demonstration and supervision. Superior experience supplied by teleoperation is introduced to deal with unknown environment constrains and correct the demonstration for next execution. DMPs model with integral barrier Lyapunov function is used to deal with the constrains in robot learning. Additionally, a radial basis function neural network based controller is developed for teleoperation and the robot to track the generated motions. Then, we prove convergence of the generated path and controller. Finally, we deploy the novel framework with two touch robots to certify its effectiveness.

1. Introduction

Robots are now widely used in many fields, from industrial manufacturing to daily life. They are not limited to structured environments and single, repetitive tasks. To reduce the programming effort for different tasks, learning from demonstrations [1] is proposed. The dynamic movement primitives (DMPs) [2] method is a flexible and effective way to transfer manipulation skills from humans to robots easily. DMPs can generalize the actions learning from demonstration and guarantee convergence to a goal position. The method has been successfully applied in many robotic scenarios, such as assembly operations [3–5], robotic surgery [6–8], and collaborative bimanual tasks [9].

However, there are many factors cause task failures using DMPs [10] including: (1) the expert cannot demonstrate every correct action for all the possible states. (2) The environment of the tasks may be variable in actual (e.g., new obstacles and constrains of manipulator). To solve these questions, some scholars introduced human supervision as a part of the system. In ref. [11], Losey and O'Malley introduced the kinematic adjustments to successfully deduce parameters of an optimal policy. In ref. [12], Nemeč et al. proposed a learning from demonstration framework where DMPs based on kinematic corrections to the behavior of an impedance-controlled robot. In ref. [13], Hagenow et al. proposed corrective shared autonomy system, corrective shared autonomy is introduced to leverage user input to address uncertainty in robot tasks by targeting corrections to task-specific variables. Hagenow used the method of supervisory control [14], which is a kind of remote control methods to remain the responsibility of the human operator. Rather than correcting the robot directly, the remote-based approach is safer and space-free [15]. In this article, corrections of operator are also introduced by a remote control system.

The correction of expert improves the quality of task completion, but it also raises new questions: (1) correction imposed by supervisor may cause manipulator over its constraint space and (2) and the

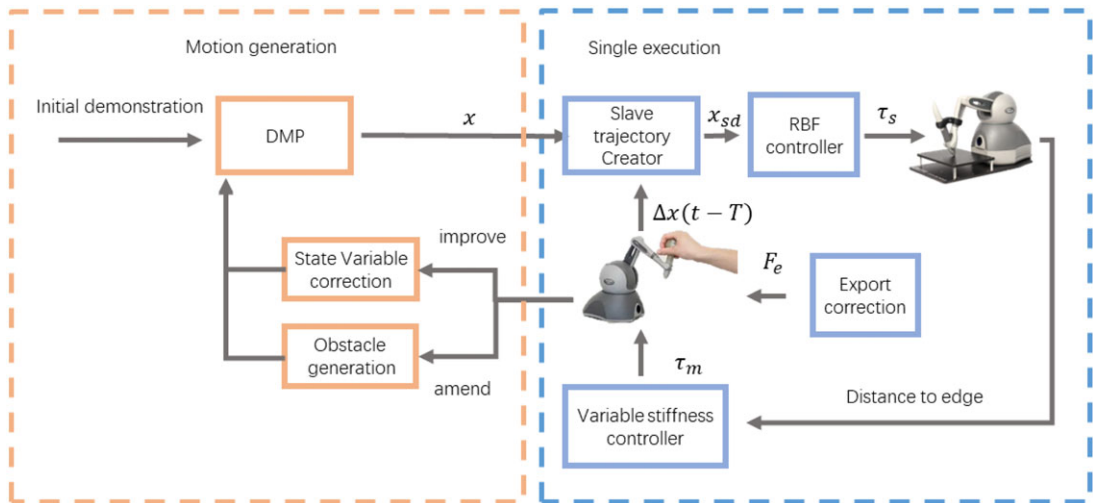


Figure 1. Block diagram illustrating the constrained system.

correction of trajectory cannot be unable to discern the intent. One effective way to deal with constraint problems is barrier Lyapunov function (BLF). In ref. [16], an asymmetric time-varying BLF is used to guarantee the time-varying output constraints. In contrast to the log [17] and tan [18] type BLF, integral BLF can limit state signals directly, rather than error signals [19]. Integral barrier Lyapunov function (IBLF) [20] is proposed to guarantee the end-effector of the robot in the constrained task space. As a result, IBLF is utilized in this article to ensure the manipulator's end-effector in the restricted task space following the repair of export.

Then correction is divided into two kinds by supervisor: improve the quality of demonstrations and avoid the collision in complex environment. The corrected motion can be used as a new quality demonstration. Collision information can be got from the correction. In ref. [21], constraints for DMPs have been successfully treated as point-like obstacles and volumetric obstacles. Based on DMPs model and BLFs, Lu et al. [22] propose a BLF-based DMPs framework with the classified constraints.

In this article, the proposed framework is illustrated in Fig. 1, which includes motion generation and single execution. Motion generation uses classic DMPs method with velocity limits inspired by ref. [23] to learn the corrected path. Single execution part designs a remote control system to ensure the end-effector of manipulator following the learned trajectory, where radial basis function neural network (RBFNN) is employed to approximate the unknown robot dynamics. The trajectory is modified by the correction information of export and the confined space.

The following is a list of the major contributions:

- A novel framework based on supervision is proposed that considers both the motion learning and task executing, including a modified DMPs method and a remote-control system for supervision.
- A new framework combining DMPs and IBLF is proposed to solve constrained trajectory planning problem based on the correction. Velocity limits are also met.
- A modified remote-control system is introduced, and constrains of system are limited via the IBLF. Furthermore, the stability of the system can be ensured using Lyapunov stability theorem.

The following is the list of paper that is organized as Section 2 introduces basic information of the DMPs and IBLF. The learning process of the corrected motion and remote corrective control system are introduced in Section 3. The experiments are presented in Section 4. Section 5 concludes this paper.

2. Preliminary work

2.1. Dynamics modeling of remote system

The dynamics of the teleoperation system for master and slave in the task space can be modeled as follows:

$$M_m(x_m) \ddot{x}_m + C_m(x_m, \dot{x}_m) \dot{x}_m + G_m(x_m) + D_m = f_m \tag{1}$$

$$M_s(x_s) \ddot{x}_s + C_s(x_s, \dot{x}_s) \dot{x}_s + G_s(x_s) + D_s = f_s \tag{2}$$

where $x_m, \dot{x}_m, \ddot{x}_m$ and $x_s, \dot{x}_s, \ddot{x}_s$ are the position, velocity, and acceleration signals of end-effort for master and slave manipulators, $M_m(x_m)$ and $M_s(x_s)$ are the inertia matrices, $C_m(x_m, \dot{x}_m)$ and $C_s(x_s, \dot{x}_s)$ account centrifugal/Coriolis terms, $G_m(x_m)$ and $G_s(x_s)$ are the gravitational matrices, D_m and D_s are the modeling errors and external disturbance, and f_m and f_s are the control input of master and slave devices.

Property 1: The matrices $\dot{M}_m - 2C_m$ and $\dot{M}_s - 2C_s$ are skew-symmetric.

2.2. RBFNN

RBFNN is used to approximation manipulator uncertainties to handle the uncertainty issue in the dynamic model. The following introduces the definition of the RBF neural network utilized in this article:

$$F(\vartheta) = W^T S(\vartheta) + \varepsilon \tag{3}$$

where ϑ denotes the input of the neural network, $W = [\omega_1, \omega_2, \dots, \omega_n]^T$ is the ideal weight parameter, n is the number of RBFNN nodes, and $\varepsilon(\vartheta)$ is the approximation errors. $S(\vartheta) = [s_1(\vartheta), s_2(\vartheta), \dots, s_n(\vartheta)]^T$ is the Gaussian basis function in the form as

$$s_i(\vartheta) = \exp\left(-\frac{(\vartheta - c_i)^2}{b_i^2}\right) \tag{4}$$

where c_i, b_i are the center and width of the neuron. The ideal weight vector W^* is an artificial quantity for the purposed method, which aims to minimize the value of W .

$$W^* = \operatorname{argmin}\{\|\varepsilon\|\} \tag{5}$$

2.3. General DMPs model

DMPs is used to trajectory learning. DMPs consists of two main components: (1) spring-damper-type equation which draws our system to the target and (2) a forcing term that gets the desire behaviors.

The DMPs model is first introduced as

$$\begin{cases} \tau \dot{v} = \alpha_z(\beta_z(g - x) - v) + f(s) \\ \tau \dot{x} = v \\ \tau \dot{s} = -\alpha_s s, s_0 = 1 \end{cases} \tag{6}$$

where α_z, β_z are the positive parameters, x is the position variable, g is the goal point, v is the velocity, \dot{v} is the acceleration, $\tau > 0$ is the time constant, s is a phase variable that avoid explicit time dependence, the initial value of s is set as 1, and α_s is the factor to modify the converging time

The forcing term $f(s)$ is defined as

$$f(s) = \frac{\sum_{i=1}^N \psi_i w_i}{\sum_{i=1}^N \psi_i} s(g - y_0) \tag{7}$$

The forcing term has the components that consist of N Gaussian basis functions, enable the encoding of demonstrated trajectory, where y_0 is the starting position state, w_i is the column of the weight vector,

ψ_i is Gaussian radial basis function, where $\psi_i(s) = \exp(-h_i(s - c_i)^2)$, c_i is the center of Gaussian kernels and h_i is the variance. The vector w_i can be trained with supervised learning algorithms such as locally weighted regression.

The calculating process is proposed to minimize the error function as following:

$$\min\{f^r(s) - f(s)\} \tag{8}$$

where $f(s)$ is an item calculated by the trajectory in demonstration, and $f^r(s)$ represents the target value as following:

$$f^r(s) = \tau \dot{v} - \alpha_z(\beta_z(g - x) - v) \tag{9}$$

2.4. General IBLF

Consider the strict feedback nonlinear system described as

$$\begin{cases} \dot{x}_1 = f_1 + g_1x_2 \\ \dot{x}_2 = f_2 + g_2u \\ y = x_1 \end{cases} \tag{10}$$

where f_1, f_2, g_1, g_2 are smooth functions, x_1, x_2 are the states, and u and y are the input and output.

Introduce the IBLF candidate in the following:

$$V_1 = \int_0^{z_1} \frac{\rho k_c^2}{k_c^2 - (\rho + x_r)^2} d\rho \tag{11}$$

where k_c is the constant, x_r is the variable, ρ is a member of integrating, and the error variable $z_1 = x_1 - x_r, z_2 = x_2 - \alpha, \alpha$ is a continuously differentiable function.

The time derivative of V_1 is given by

$$\dot{V}_1 = \frac{z_1 k_c^2}{k_c^2 - x_1^2} \dot{z}_1 + \frac{\partial V_1}{\partial x_r} \dot{x}_r \tag{12}$$

where

$$\frac{\partial V_1}{\partial x_r} = z_1 \left(\frac{k_c^2}{k_c^2 - x_1^2} - \lambda \right) \tag{13}$$

$$\lambda = \frac{k_c}{2z_1} \ln \frac{(k_c + z_1 + x_r)(k_c - x_r)}{(k_c - z_1 - x_r)(k_c + x_r)} \tag{14}$$

The virtual control variable α can be designed as follows:

$$\alpha = \frac{1}{g_1} \left(-k_1 z_1 - f_1 + \frac{g_1(k_c^2 - x_1^2) \dot{x}_r \lambda}{k_c^2} \right) \tag{15}$$

where k_1 is positive constant. Substituting (15) into (12), we can get

$$\dot{V}_1 = -\frac{k_1 z_1^2 k_c^2}{k_c^2 - x_1^2} + \frac{g_1 z_1 z_2 k_c^2}{k_c^2 - x_1^2} \tag{16}$$

Then, a Lyapunov function candidate is chosen as follows:

$$V_2 = V_1 + \frac{1}{2} z_2^2 \tag{17}$$

The time derivative of V_2 is given by

$$\dot{V}_2 = -\frac{k_1 z_1^2 k_c^2}{k_c^2 - x_1^2} + \frac{g_1 z_1 z_2 k_c^2}{k_c^2 - x_1^2} + z_2(f_2 + g_2u - \dot{\alpha}) \tag{18}$$

The control law is designed as

$$u = \frac{1}{g_2} \left(-f_2 + \dot{\alpha} - k_2 z_2 - \frac{g_1 z_1 k_c^2}{k_c^2 - x_1^2} \right) \tag{19}$$

3. Proposed framework

A novel approach that combines DMPs and IBLF is introduced to address limited trajectory planning based on the correction. The remote-control system is made to guarantee that the manipulator end-effector is within the restricted space and to offer export corrective information. The controller of master and slave is designed and analyzed.

3.1. DMPs based on IBLF

The expression of DMPs as (6) can be revised as nonlinear system as follows:

$$\begin{cases} \dot{x}_1 = g_1 x_2 \\ \dot{x}_2 = f_2 + u + \Delta u \\ \tau \dot{s} = -\alpha_s s, s(0) = 1 \\ \dot{\tau} = -\gamma_0(\tau - \tau_0) + \gamma_1 \tau v^T \sigma(v), \tau_0 = 1 \\ y = x_1 \end{cases} \tag{20}$$

where x_1, x_2, g_1 represent the $x, v, 1/\tau$ in Eq. (5), $f_2 = (\alpha_z(\beta_z(g - x) - v) + \gamma\sigma(v))/\tau$, u is the forcing function $f(s)$, Δu is a term added by IBLF, $\gamma, \gamma_0, \gamma_1$ are the positive constants, to allow the velocity being close to the limit while still not exceeding it, the σ_i is designed as

$$\sigma_i(v_i) = \frac{v_i}{(A_i - v_i)(B_i + v_i)} \tag{21}$$

where A_i, B_i are the positive constants.

Similar to the general form of IBLF, we define $z_1 = x_1 - x_r, z_2 = x_2 - \dot{\alpha}$, x_r is the desired state in control system, while there is no such state in DMPs, so here we introduce the motion generated by DMPs method without any limits as the desired state:

$$\begin{cases} \dot{x}_r = g_1 v_c \\ \dot{v}_c = \frac{\alpha_z(\beta_z(g - x_r) - v_c) + f(s)}{\tau} \end{cases} \tag{22}$$

Theorem 1: The output constraint is never violated, and all closed loop signals are confined if the following conditions are met for the DMPs function represented by (20).

$$\begin{cases} \alpha = \frac{1}{g_1} \left(-k_1 z_1 + \frac{g_1(k_c^2 - x^2) \dot{x}_r \lambda}{k_c^2} \right) \\ u + \Delta u = -f_2 + \dot{\alpha} - k_2 z_2 - \frac{g_1 z_1 k_c^2}{k_c^2 - x_1^2} \end{cases} \tag{23}$$

where λ is introduced in (14)

To get the form of Δu , we calculate u without added term is

$$u = \dot{v} - f_2 \tag{24}$$

So

$$\Delta u = -k_2 z_2 - \frac{g_1 z_1 k_c^2}{k_c^2 - x_1^2} + \dot{\alpha} - \dot{v} \tag{25}$$

Proof:

We synthesize a Lyapunov function as

$$V^c = V_1 + \frac{1}{2}z_2^2 \tag{26}$$

where V_1 is the IBLF candidate in (11)

Then

$$\begin{aligned} \dot{V}^c &= \frac{z_1 k_c^2}{k_c^2 - x^2} \dot{z}_1 + \frac{\partial V_1}{\partial x_r} \dot{x}_r + z_2^T \dot{z}_2 \\ &= \frac{z_1 k_c^2}{k_c^2 - x^2} (\dot{x} - \dot{x}_r) + \frac{\partial V_1}{\partial x_r} \dot{x}_r + z_2^T (\dot{v} - \dot{\alpha}) \\ &= \frac{z_1 k_c^2}{k_c^2 - x^2} (g_1(z_2 + \alpha) - \dot{x}_r) + \frac{\partial V_1}{\partial x_r} \dot{x}_r \\ &\quad + z_2^T (f_2 + u + \Delta u - \dot{\alpha}) \end{aligned} \tag{27}$$

Taking the expressions of (16) into (20), we have

$$\begin{aligned} \dot{V}^c &= \frac{g_1 z_2 z_1 k_c^2}{k_c^2 - x^2} - \frac{k_1 z_1^2 k_c^2}{k_c^2 - x^2} + z_1 \lambda \dot{x}_r \\ &\quad - \frac{z_1 k_c^2}{k_c^2 - x^2} \dot{x}_r + z_1 \left(\frac{k_c^2}{k_c^2 - x_1^2} - \lambda \right) \dot{x}_r \\ &\quad + z_2^T \left(f_2 - f_2 + \dot{\alpha} - k_2 z_2 - \frac{g_1 z_1 k_c^2}{k_c^2 - x_1^2} - \dot{\alpha} \right) \\ &= \frac{g_1 z_2 z_1 k_c^2}{k_c^2 - x^2} - \frac{k_1 z_1^2 k_c^2}{k_c^2 - x^2} + z_2^T \left(-k_2 z_2 - \frac{g_1 z_1 k_c^2}{k_c^2 - x_1^2} \right) \\ &= -\frac{k_1 z_1^2 k_c^2}{k_c^2 - x^2} - k_2 z_2^2 \leq 0 \end{aligned} \tag{28}$$

where k_1, k_2 are positive numbers. according to Lyapunov stability theorem, we know that above expression guarantees global stability and the global tracking convergence in the system.

3.2. Master controller design

The control command on the master robot is designed as an impedance controller, such that the position of the robot end effector can be moved by export, let us consider the modeling of the operating torque. In this paper, a damping-stiffness model is considered.

$$F_e = k_m e_m + d_m \dot{e}_m \tag{29}$$

where k_m and d_m denote designed damping and stiffness matrices, where $e_m = x_{md} - x_m, x_{md}$ is the desired state, in this remote corrective system, it is fixed value. For avoiding the slave robot moving off the edge of the surface during control, a variable stiffness is proposed

$$k_m = k_{min} + e^{-d(x_m)} (k_{max} - k_{min}) \tag{30}$$

where α is the proximity of distance to boundary, closer get to the boundary, the closer get to 1, where $d(x_m)$ is the distance to the nearest edge, k_{max}, k_{min} are positive numbers.

Then, the control torque can be designed as

$$f_m = \hat{G}_m + u_m + F_e \tag{31}$$

where u_m is robust term, and \hat{G}_m is the estimation of G , which satisfies: $G - \hat{G}_m = \varepsilon_m$.

3.3. Slave controller design

Inspired by ref. [24], the desired position signal x_{sd} can be derived via the slave trajectory creator. For simplicity, a filter as

$$V_f(s) = 1 / (1 + \tau_f s)^2 \tag{32}$$

It is used with the input of $x_m(t - T(t))$ to create the correction state x_{sc} , then $x_{sd} = \Delta x_{sd} + x_{sdmp}$, where $\Delta x_{sd} = x_{sc} - x_{sd}$ is the forward kinematic function of the Touch x_{sdmp} is the path generate by DMPs.

Due to the disturb and errors in tracking, the trajectory generated by creator may not guarantee the x in the constraint space all the time, we can obtain x by a soft saturation function, then if the x cross the bound it will map the inside state. We derive the following via a saturation function to guarantee that the reference trajectory stays inside the limited region:

$$x_{sd} = \begin{cases} x_{sd}, & |x_{sd}| \leq \eta k_{c_i} \\ \eta k_{c_i}, & x_{sd} < -\eta k_{c_i} \\ -\eta k_{c_i}, & x_{sd} > \eta k_{c_i} \end{cases} \tag{33}$$

where η is a constant very close to 1.

Because of the uncertainty of robot dynamic and model, reference trajectory cannot ensure the end-effector stay in the constrained space. IBLF method is introduced to ensure the constrains of the predefined task space met.

We can obtain the dynamics system of slave

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = M_s^{-1}(f_s - D_s - G_s - C_s \dot{x}_s) \\ y = x_1 \end{cases} \tag{34}$$

Then, we define $z_1 = x_1 - x_{sd}, z_2 = x_2 - \alpha$, where z_1, z_2 are error variables, and α denotes the virtual control variable, we design it as

$$a = -k_{s1} z_1 + \frac{(k_c^2 - x_1^2) \dot{x}_r \lambda}{k_c^2} \tag{35}$$

where k_{s1} is the positive number, k_c is the limit in Cartesian space, and λ is given in (14)

The control law is designed as

$$f_s = -\frac{z_1 k_c^2}{k_c^2 - x_1^2} - k_{s2} z_2 + \hat{W}^T S(Z) + u_s \tag{36}$$

In the actual system, the dynamics parameters and are typically unknown. RBFNN is used to approximation manipulator uncertainties to handle the uncertainty issue in the dynamic model. The neural network's input on the slave side can be chosen as $Z = [x_s, \dot{x}_s, \alpha, \dot{\alpha}]$. RBF neural network is defined as

$$W^T S(Z) = M_s \dot{a} + C_s a + G_s + e_s \tag{37}$$

3.4. Stability analysis

Define the Lyapunov function of master and slave as

$$V_m = \frac{1}{2} (\dot{x}_m^T M_m \dot{x}_m + e_m^T k_m e_m) \tag{38}$$

$$V_s = V_1 + \frac{1}{2} z_2^T M_s z_2 + \frac{1}{2} tr(\tilde{W}_s^T G_s^{-1} \tilde{W}_s) \tag{39}$$

where $\tilde{W} = \hat{W} - W$

The Lyapunov candidate function of whole system as

$$\begin{aligned}
 V &= V_m + V_s \\
 &= \frac{1}{2}(\dot{x}_m^T M_m \dot{x}_m + e_m^T k_m e_m) + V_1 + \frac{1}{2}z_2^T M_s z_2 + \frac{1}{2}\text{tr}(\tilde{W}_s^T \Gamma_s^{-1} \tilde{W}_s)
 \end{aligned} \tag{40}$$

Then, the derivative of V_m can be calculated as

$$\dot{V} = \ddot{x}_m^T M_m \dot{x}_m + \frac{1}{2}\dot{x}_m^T \dot{M}_m \dot{x}_m + \dot{e}_m^T k_m e_m + \dot{V}_1 + z_2^T M_s \dot{z}_2 + \frac{1}{2}z_2^T \dot{M}_s z_2 + \text{tr}(\tilde{W}_s^T \Gamma_s^{-1} \dot{\tilde{W}}_s) \tag{41}$$

Combine (1), (2), (16), (41)

$$\begin{aligned}
 \dot{V} &= \dot{x}_m^T \left(f_m - C_m \dot{x}_m - G_m - D_m + \frac{1}{2} \dot{M}_m \dot{x}_m \right) - \dot{e}_m^T k_m e_m \\
 &\quad - \frac{k_{s1} z_1^2 k_c^2}{k_c^2 - x_1^2} + \frac{z_1 z_2 k_c^2}{k_c^2 - x_1^2} + z_2^T (f_s - G_s - C_s \dot{x}_s - D_s - M_s \dot{\alpha}) \\
 &\quad + \frac{1}{2} z_2^T \dot{M}_s z_2 + \text{tr} \tilde{W}_s^T \Gamma_s^{-1} \dot{\tilde{W}}_s
 \end{aligned} \tag{42}$$

Substituting (31) (36) into (42), we have

$$\begin{aligned}
 \dot{V} &= \dot{x}_m^T \left(\hat{G}_m + u_m + F_e - C_m \dot{x}_m - G_m - D_m + \frac{1}{2} \dot{M}_m \dot{x}_m \right) - \dot{e}_m^T k_m e_m \\
 &\quad - \frac{k_{s1} z_1^2 k_c^2}{k_c^2 - x_1^2} + \frac{1}{2} z_2^T (\dot{M}_s - 2C_s) z_2 \\
 &\quad + z_2^T \left(\frac{z_1 k_c^2}{k_c^2 - x_1^2} + f_s - D_s - G_s - C_s \alpha - M_s(q) \dot{\alpha} \right) \\
 &\quad + \text{tr} \tilde{W}_s^T \Gamma_s^{-1} (\dot{\hat{W}}_s - \dot{W}_s)
 \end{aligned} \tag{43}$$

then

$$\begin{aligned}
 \dot{V} &= \dot{x}_m^T (\hat{G}_m + u_m + F_e - G_m - D_m) - \dot{e}_m^T k_m e_m - \frac{k_{s1} z_1^2 k_c^2}{k_c^2 - x_1^2} \\
 &\quad + z_2^T \left(\frac{z_1 k_c^2}{k_c^2 - x_1^2} + f_s - D_s - G_s - C_s a - M_s(q) \dot{\alpha} \right) + \text{tr} \tilde{W}_s^T G_s^{-1} \dot{\hat{W}}_s \\
 &= \dot{x}_m^T (\hat{G}_m + u_m - G_m - D_m) + \dot{e}_m^T (F_e - k_m e_m) - \frac{k_{s1} z_1^2 k_c^2}{k_c^2 - x_1^2} \\
 &\quad - k_{s2} z_2^2 + z_2^T (u_s + \hat{W}^T S(Z) - D_s - G_s - C_s a - M_s(q) \dot{\alpha}) + \text{tr} \tilde{W}_s^T G_s^{-1} \dot{\hat{W}}_s \\
 &= -d_m \dot{e}_m^2 - \frac{k_{s1} z_1^2 k_c^2}{k_c^2 - x_1^2} - k_{s2} z_2^2 + \dot{x}_m^T (u_m - e_m - D_m) \\
 &\quad + z_2^T (u_s - D_s - e_s) + \text{tr} \tilde{W}_s^T (G_s^{-1} \dot{\hat{W}}_s + z_2^T S(Z))
 \end{aligned} \tag{44}$$

Furthermore, the robust term in a controller, which is used to deal with estimating error, external disturbance, and modeling error, can be created as

$$u_m = -(d_m + \varepsilon_{mb}) \text{sgn}(\dot{x}_m) \tag{45}$$

where $\|D_m\| \leq d_m$, $\|\varepsilon_m\| \leq \varepsilon_{mb}$, d_m and ε_{mb} are positive constants.

$$u_s = -(d_s + \varepsilon_{sb}) \text{sgn}(z_2) \tag{46}$$

where $\|D_s\| \leq d_s$, $\|\varepsilon_s\| \leq \varepsilon_{sb}$ and d_s and ε_{sb} are positive constants.



Figure 2. Experimental setup for testing.

Thus, the adaptive law, which is used to estimate the RBF neural network parameters online and real time, can be designed as

$$\dot{\hat{W}}_s = -\Gamma_s S(Z) z_2^T \tag{47}$$

then

$$\begin{aligned} \dot{V} &\leq -\dot{e}_m^T D \dot{e}_m - |\dot{x}_m^T| (d_m + \varepsilon_{mb} + \varepsilon_m + D_m) \\ &\quad - \frac{k_1 z_1^2 k_c^2}{k_c^2 - x_1^2} - K_2 z_2^2 + |z_2^T| (d_s + \varepsilon_{sb} + \varepsilon_s + D_s) \\ &\leq 0 \end{aligned} \tag{48}$$

According to the Lyapunov stability theorem, the Lyapunov function is uniformly positive defined, its derivative is negative defined. The aforementioned formula ensures global stability and the convergence of global tracking in the system employing the suggested controller.

4. Experiment

The proposed method has been verified on two touch robots, which are haptic devices manufactured by 3D Systems. As shown in Fig. 2, the device has three degrees-of-freedom can be driven by a torque controller. In our experiments, two touch robots were served as master and slave. The dynamics of two touch robot can be modeled as (1) and (2), where the knowledge of the dynamic parameters is estimated by RBF neural network in the teleoperation process. The proposed method is tested by two groups of experiments:

The test of controllers: The slave robot is performed to move along the learned trajectory to accomplish the task, and trajectory is modified through operating the master robot in remote environment by human operator.

The test of trajectory learning: The IBLF-based DMPs is used to generate a learned trajectory. Constraints of task space and velocity are considered.

4.1. Constrained control effect

In this part, we apply the designed controllers to verify work of the proposed algorithm. The experiments are performed on the two touch robots. First, a demonstration trajectory is given by operator, then the

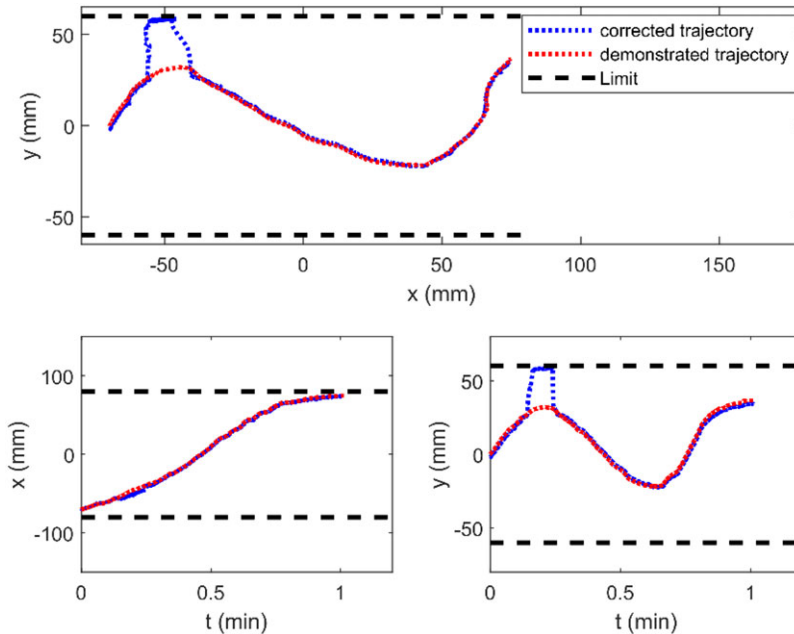


Figure 3. Results of remote corrective control.

slave robot moves along the trajectory and the master robot is operated to modify the trajectory of the slave robot.

For the master robot, the parameters are chosen as $k_{min} = 10$, $k_{max} = 100$, $d_m + \varepsilon_{mb} = 0.1$, $D = 1$, $\Gamma_m = 1$, $x_{md} = [50, 0, 60]$ is near the mid of its workspace. For the slave robot, control parameters are chosen as $k_{s1} = 10$, $k_c = 60$ and $k_{s2} = 15$, $d_s + \varepsilon_{sb} = 0.1$, $\Gamma_s = 1$. The slave robot receives the correction information from the master after filter and then applies the soft saturation function to generate the desired trajectory of the end effector in task space. Parameter of soft saturation function is designed as $\eta = 0.98$, the weight parameters of the RBF NN are initialized as 0, and the centers of the functions are distributed in the interval $[0, 1]$. The human operates the master device toward boundaries in the y-axes orderly.

The experiment results are shown in Fig. 3, our suggested controller guarantees that the end-effector tracks the reference trajectory in real time while operating inside the restricted area. Operator can feel the resistance when moving away from the hold position which prevent accidental contact. When the operator forced the slave to cross the bound of $y = 60$, the soft saturation function and IBLF controller work together to ensure the end-effector stay below the limit.

4.2. Trajectory learning

The second group of experiments aims to test the ILBF-based motion model. The ability of constrains of task space and the velocity are tested. A drawing task is designed for the test. The common parameters of DMPs are chosen as $\alpha_s = 5$, $\alpha_z = 10$, $\beta_z = 100$ and others are set separately in each experiment.

To validate the correction performance of the IBLF-based DMPs, the parameters of IBLF are $k_1 = 1$, $k_2 = 10$, $k_c = 8$. The speed constraint item is added to the IBLF-based method, and the selected parameter is $A = 100$, $B = -100$, $\gamma = 10$, $\gamma_0 = 5$, $\gamma_1 = 10$.

As shown in Fig. 4, during the operation, the track was corrected by teleoperation equipment, and an obstacle was added in the early stage of the experiment to make the movement within the edge. The corrected trajectory will be learned then, we hope that the learned trajectory will not cross the limit where we get the information from correction.

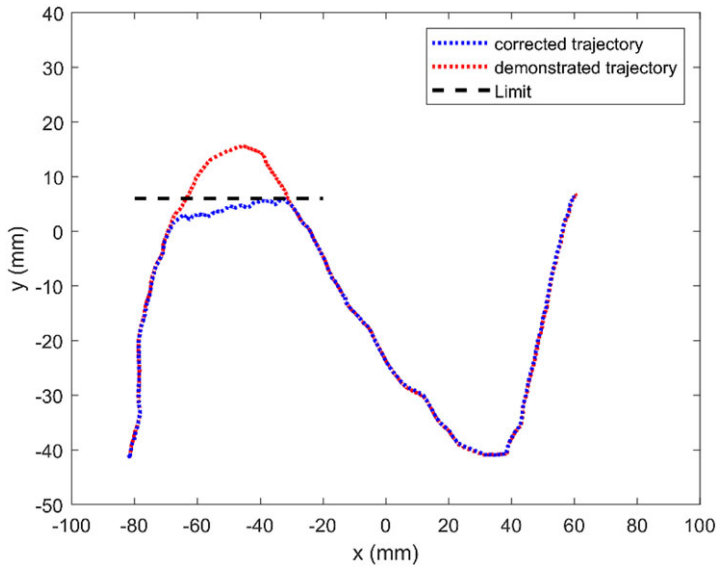


Figure 4. Motion generation of different methods.

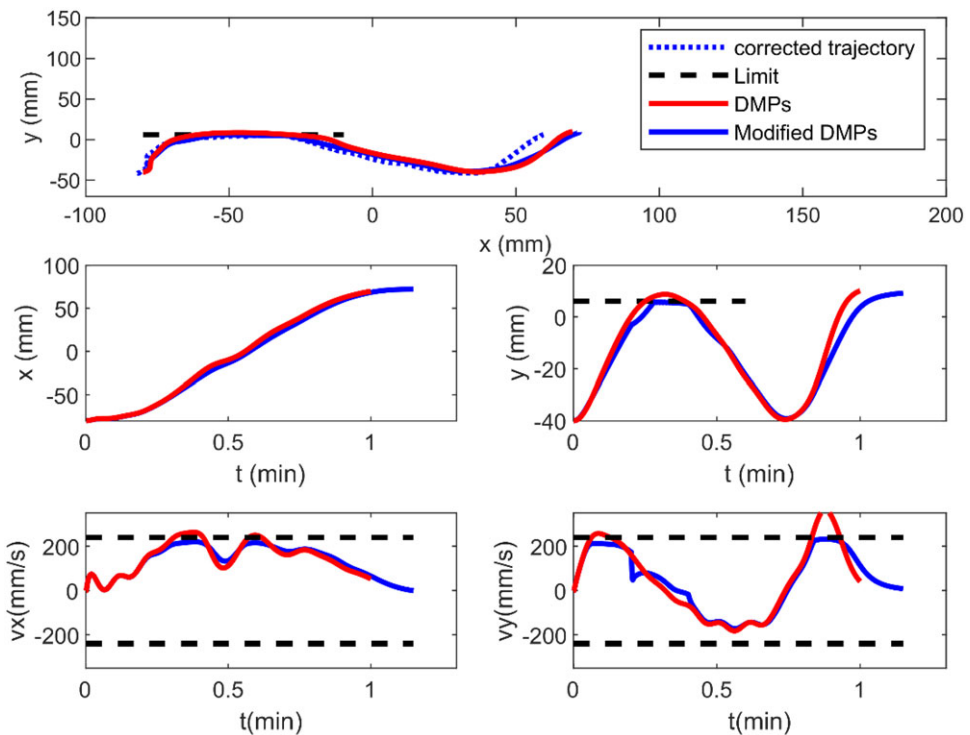


Figure 5. Motion generation of different methods.

Then the corrected trajectory is learned by a classic DMPs method and a modified DMPs method proposed in this article. As shown in Fig. 5, The classical DMPs method is compared with the IBLF-based method. In the top of the figure, the generalization process is shown, both can learn the characteristics of the trajectory, and finally approach the target point. However, it shows that the red line across the limit around $x = -50$. While the blue line always stays within the constraints. It can be clear expression

in the mid of the figure. It is the relationship between time and location. We can see that the red line over the border at $t = 0.4$. Compared with the classical method, IBLF-based method can constrain the motion within the set range. However, due to the few selected neural network nodes, the local features cannot be perfectly expressed. The bottom of the picture shows the speed constraint capability. The velocity oscillation at $t = 0.3$ is a confrontation to prevent crossing the obstacle and the desired trajectory. Comparing to the red line which is the velocity of the classic DMPs, the blue one can always stays within the constraints. It can be seen that velocity is able to stay within the limits comparing with the classic DMPs.

5. Conclusion

In this article, an IBLF constrained DMPs has been designed to generate the trajectory under limits. Our proposed controller guaranteed the state avoid the obstacle and velocity follow the bound. A control system involving IBLF-based slave controller and impedance master controller has been applied, and dynamic uncertainties are approximated by the RBFNN learning method. The proposed controller guaranteed the constrained performance in task space and robustness of the controller. The effectiveness of the system has been verified on the touch robots experiment platform. In our future work, we will do further research on the full state constrain problem of constrained DMPs method and focus on varying constrained methods.

Author contributions. Qinchuan Li and Chenguang Yang conceived and designed the study. Donghao Shi conducted data gathering and performed statistical analyses. Donghao Shi and Zhenyu Lu wrote the article.

Financial support. The authors would like to thank the Key Research and Development Project of Zhejiang Province (Grant 2021C04017).

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

- [1] C. G. Atkeson, C. G. Schaal and A. Systems, "Learning from demonstration," *Robot. Auton. Syst.* **47**(2-3), 65–67 (2004).
- [2] S. Schaal, P. Mohajerian and A. J. P. I. B. R. Ijspeert, "Dynamics systems vs. optimal control - a unifying view," *Brain Res.* **165**, 425–445 (2007).
- [3] T. Tang, H. C. Lin, Y. Zhao, Y. Fan and M. Tomizuka, "Teach Industrial Robots Peg-Hole-Insertion by Human Demonstration," *IEEE International Conference on Advanced Intelligent Mechatronics*, (2016).
- [4] D. Vogt, S. Stepputtis, S. Grehl, B. Jung and H. B. Amor, A System for Learning Continuous Human-Robot Interactions from Human-Human Demonstrations, *2017 IEEE International Conference on Robotics and Automation (ICRA)*, (2017).
- [5] R. Lioutikov, G. Neumann, G. Maeda, J. Peters and IEEE, Probabilistic Segmentation Applied to an Assembly Task, *2015 IEEE-RAS 15th International Conference on Humanoid Robots (IEEE-RAS International Conference on Humanoid Robots)*, (2015) pp. 533–540.
- [6] C. Moro, G. Nejat and A. Mihailidis, "Learning and personalizing socially assistive robot behaviors to aid with activities of daily living," *ACM Trans. Human-Robot Interact.* **7**(2), 1–25 (2018).
- [7] W. Xu, J. Chen, H. Y. K. Lau and H. Ren, "Automate surgical tasks for a flexible serpentine manipulator via learning actuation space trajectory from demonstration," *IEEE Int. Conf. Robot. Autom.*, 4406–4413 (2016).
- [8] T. Osa, K. Harada, N. Sugita, M. Mitsuishi and IEEE, Trajectory Planning under Different Initial Conditions for Surgical Task Automation by Learning from Demonstration, *2014 IEEE International Conference on Robotics and Automation (IEEE International Conference on Robotics and Automation ICRA)*, (2014) pp. 6507–6513.
- [9] A. Gams, B. Nemeč, A. J. Ijspeert and A. Ude, "Coupling movement primitives: Interaction with the environment and bimanual tasks," *IEEE Trans. Robot.* **30**(4), 816–830 (2014).
- [10] B. D. Argall, S. Chernova, M. Veloso and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.* **57**(5), 469–483 (2009).
- [11] D. P. Losey and M. K. O'Malley. *Learning the Correct Robot Trajectory in Real-Time From Physical Human Interactions*, vol. 27. ACM, New York, NY, USA, (2019).

- [12] B. Nemeč, L. Zlajpah, S. Slajpa, J. Piskur and A. Ude, An Efficient PBD Framework for Fast Deployment of Bi-Manual Assembly Tasks, *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, (2018).
- [13] M. Hagenow, E. Senft, R. Radwin, M. Gleicher and M. Zinn, “Corrective shared autonomy for addressing task variability,” *IEEE Robot. Autom. Lett.* **6**(2), 1–1 (2021).
- [14] T. B. J. M. P. Sheridan, Telerobotics, automation, and human supervisory control, (1992).
- [15] W. Y. Si, Y. Guan and N. Wang, “Adaptive compliant skill learning for contact-rich manipulation with human in the loop,” *IEEE Robot. Autom. Lett.* **7**(3), 5834–5841 (2022).
- [16] C. Yang, D. Huang, W. He, L. J. I. T. O. N. N. Cheng and L. Systems, “Neural control of robot manipulators with trajectory tracking constraints and input saturation,” *IEEE Trans. Neural Netw. Learn. Syst.* **99**, 1–12 (2020).
- [17] K. P. Tee, S. S. Ge and E. H. Tay, “Barrier Lyapunov functions for the control of output-constrained nonlinear systems,” *Automatica* **45**(4), 918–927 (2009).
- [18] X. Jin, “Fault tolerant finite-time leader follower formation control for autonomous surface vessels with LOS range and angle constraints,” *Automatica* **68**, 228–236 (2016).
- [19] H. Wei, Z. Shuang and S. S. J. I. T. O. I. E. Ge, “Adaptive control of a flexible crane system with the boundary output constraint,” *IEEE Trans. Ind. Electron.* **61**(8), 4126–4133 (2014).
- [20] W. He, C. Xue, X. Yu, Z. Li and C. J. I. T. O. A. S. Yang, “Admittance-based controller design for physical human-robot interaction in the constrained task space,” *IEEE Trans. Autom. Sci. Eng.* **99**, 1–13 (2020).
- [21] S. Calinon, F. D’Halluin, E. L. Sauser, D. G. Caldwell, A. G. J. R. Billard and A. M. IEEE, “Learning and reproduction of gestures by imitation,” *IEEE Robot. Autom. Mag.* **17**(2), 44–54 (2010).
- [22] Z. Lu, N. Wang and C. J. I. A. T. o. M. Yang, “A constrained DMPs framework for robot skills learning and generalization from human demonstrations,” *IEEE/ASME Trans. Mech.* **99**, 1 (2021).
- [23] W. Si, N. Wang and C. J. N. C. Yang, “Composite dynamic movement primitives based on neural networks for human-robot skill transfer,” *Neural Comput. Appl.* **5**, 1–11 (2021).
- [24] Z. Chen, F. Huang, W. Sun, J. Gu and B. J. I. A. T. o. M. Yao, “RBF neural network based adaptive robust control for nonlinear bilateral teleoperation manipulators with uncertainty and time delay,” *IEEE/ASME Trans. Mech.* **99**, 1 (2019).