Dynamic Movement Primitives-based Human Action Prediction and Shared Control for Bilateral Robot Teleoperation

Zhenyu Lu, *Member*, *IEEE,* Weiyong Si, Ning Wang, *Member*, *IEEE,* and Chenguang Yang, *Fellow, IEEE*

***Abstract*—This paper presents a novel shared control teleoperation framework that integrates imitation learning and bilateral control to achieve system stability based on a new dynamic movement primitives (DMPs) observer. First, a DMPs-based observer is first created to capture human operational skills through offline human demonstrations. The learning results are then used to predict human action intention in teleoperation. Compared with other observers, the DMPs-based observer incorporates human operational features and can predict long-term actions with minor errors. A high-gain observer is established to monitor the robot's status in real-time on the leader side. Subsequently, two controllers on both the follower and leader sides are constructed based on the outputs of the observers. The follower controller shares control authorities to address accidents in real-time and correct prediction errors of the observation using delayed leader commands. The leader controller minimizes position-tracking errors through force feedback. The convergence of the predictions of the DMPs -based observer under the time delays and teleoperation system stability are proved by building two Lyapunov functions. Finally, two groups of comparative experiments are conducted to verify the advantages over other methods and the effectiveness of the proposed framework in motion prediction with time delays and obstacle avoidance.**

***Index Terms*—Dynamic Movement Primitives (DMPs) , Shared control , Stability proof, Time delay, Teleoperation**

I. INTRODUCTION

TELEOPERATION technology has been widely used for exploration in the deep sea and toxic environments, and nuclear decommissioning, which enhances human reachability and delivers human actions to guide the movements of robots. Shared control is a typical control mode in teleoperation [1], [2]. However, due to the time delays, the commands sent from the leader side may not be able to respond in real time to the accidents happened on the remote robot side. Therefore, shared control, allowing robot controllers to share the control authority between the autonomous reactions and the time-delayed action commands from leader side, can balance control requirements of both humans and robots to enable the effective interventions in emergencies [3], [4].

Generally, the two agents of teleoperation share information, such as velocity, position and force to realize semi-autonomous control. Some researchers extended the sharing of information to impedance [5], [6] and haptic information [7], [8] to improve system dexterity and manipulability through robot's autonomy. As reviewed in [9], shared control can be classified into three categories: Semi-Autonomous Control (SAC), State-Guidance Shared Control (SGSC), and State-Fusion Shared Control (SFSC), according to the sharing ways between humans and robots. Among the three classes, the SFSC has an innate and essential advantage in the seamless autonomy-level adaptation owing to the arbitration mechanism. For example, Ezeh *et al.* have proposed a probabilistic fusion mechanism to combine human's intended motions and autonomous planner's actions to control a wheelchair [10]. Selvaggio *et al.* proposed a shared - control teleoperation framework for robot manipulators, which transport an object on a tray, which considered the case that an object breaks contact with the robot end-effectors. The shared control method could regulate the remote robot's movement to prevent the object from sliding over the tray [11]. Gottardi *et al.* proposed a real-time shared control teleoperation framework that integrated an artificial potential field which is improved by the dynamic generation of escaping points around obstacles to overcome obstacles [13]. The methods addressed the problems in certain tasks such as preventing object sliding[10]. However, the robot's autonomous control was based on certain principles instead of human motion intentions.

Some researchers have addressed this problem by integrating imitation learning and teleoperation [12]-[17]. The objective is to enable robots to learn skills from human demonstrations first, and then these skills are generalized for the shared control with delayed human inputs. Typically, an arbitration mechanism is introduced to mediate between robots and humans. As outlined in [9], this arbitration mechanism takes the forms of weighted combination, probabilistic fusion, and phase switching. For example, Xi *et al.* proposed a shared control framework where manipulation skills are learned by a task-parameterized hidden semi-Markov model (TP-HMM) from human demonstrations. The estimation of robots based on learning results can correct the inputs of the operators and provide manipulation assistance

Manuscript received Month xx, 2xxx; revised Month xx, xxxx; accepted Month x, xxxx. *(Corresponding author: Chenguang Yang)*
Zhenyu Lu, Weiyong Si, Ning Wang, and Chenguang Yang are with the Bristol Robotics Laboratory, University of the West of England, BS16 1QY, United Kingdom, (E-mail: Zhenyu.lu@ uwe.ac.uk; weiyong.si@ uwe. ac.uk,katie.wang@brl.ac.uk; cyang@ieee.org).

[14]. El-Hussieny *et al.* extracted human hand positions and proposed a framework with two key components: intention prediction and command arbitration to reduce control time and labor burden [16]. Ly *et al.* proposed a shared control paradigm incorporating robot actions learned from human demonstrations and dynamically adjusting the level of robotic assistance based on how closely the detected intentions match these trajectories. Human motion intention was predicted by a Deep Q-Network (DQN) with consideration of current robot states and baseline trajectories learned using Probabilistic Movement Primitives to generate adaptive force guidance [17].

These frameworks can improve robot manipulation dexterity through learning from human demonstration. However, there are several key problems have not been solved. 1) The learned skills are not updated timely by human online intervention. The suitability of the learning results is questioned; 2) The key influence factor, time delay, is not considered; 3) Dynamics uncertainties and various errors are few considered and system stability is not strictly proved in theory [13]. For the questions, we developed a Dynamic Movement Primitives (DMPs) -based observer to make a timely prediction and correction of human intentions. Then a shared control teleoperation framework is developed with the following contributions:

1) We develop a DMPs-based observer capable of predicting human action intentions and correcting the predictions using delayed tracking errors. The observer can be applied independently and adjusted for integration into the teleoperation system to enhance robot autonomy.
2) We build a new shared control framework for teleoperation systems based on observers. The DMPs-based observer is used to estimate human intentions on the leader side, while the high-gain observer predicts the state of robots in real-time. Signal measuring errors and uncertain dynamics are taken into consideration in the controller design.
3) We prove the convergence of estimations from the DMPs-based observer and prove the stability of the shared control teleoperation system under varying time delays by creating two Lyapunov functions. The effectiveness of the proposed framework is validated through two experiments.

II. PRELIMINARY WORK

A. Model of teleoperation system

Using the symbols described in Table 1, the teleoperation system in a Lagrange form is expressed as:

$$\begin{cases} M_l(q_l)\ddot{q}_l + C_l(q_l,\dot{q}_l)\dot{q}_l + G_l = J_l^T(q_l)F_h - \tau_l \\ M_f(q_f)\ddot{q}_f + C_f(q_f,\dot{q}_f)\dot{q}_f + G_f = \tau_f - J_f^T(q_f)F_e \end{cases}, \quad (1)$$

where $M_i(q_i)$ and $C_i(q_i,\dot{q}_i), i=l,f$ are the inertia matrix and the centripetal and Coriolis matrix, which are expressed as M_i and C_i in simple, G_i is the gravitational torque, and $J_i(q_i)$ is the Jacobian matrix. F_h is the human operational force and $F_e = K_x(x_f - x_f^0) + D_x\dot{x}_f$ is the environment force, where K_x and D_x are stiffness and damping factors, x_f represent the position of the robot end. τ_l and τ_f represent the control torques.

TABLE 1. SYMBOLS AND MEANINGS

Symbols	Meanings
$q_i, i=l,f$	Joints of robots and manipulators, and i represent the agent in the leader and follower sides
d_t	Time delays
$q^{dt}, q(t-d_t)$	Delayed signals with the time delay d_t
$q_i, \hat{q}_i$	Real value and estimation of robot joints
τ_i	τ_i is control torque
η_i	Estimation errors $\eta_i := q_{\bar{i}} - \hat{q}_i, \bar{i}$ is the opposite role to i in the set (l,f)
e_i	Control errors $e_i := q_i - \hat{q}_i$
F_h	Human force exerting on the manipulator
F_e	Environmental force against the robots

Several assumptions and a lemma are presented as follows:

Assumption 1: [19] The communication delays are bounded: $\underline{d}_t \le d_t \le \bar{d}_t$ and the time derivative of d_t satisfies $0 < |\dot{d}_t| \le \mu_t < 1$, where μ_t is a constant factor.

Assumption 2: [20] [23] Due to the measuring noise and time delays in measurement, there exists the following relationship: $|\bar{F}_e - F_e| \le \beta_e, |K_e\bar{F}_h^e - F_h| \le \beta_h$, where $\bar{F}_e$ is the environmental force measurements and $\bar{F}_h^e$ represents the rendering force in the leader side , and β_e, β_h and K_e are positive constants.

Assumption 3: [21] The symmetric positive definite matrices M_l and M_f, and inverse matrices M_l^{-1} and M_f^{-1} are bounded:

$$\begin{aligned} &\lambda_{min}(M_i)I \le M_i \le \lambda_{max}(M_i)I, \\ &\lambda_{min}(M_i^{-1})I \le M_i^{-1} \le \lambda_{\max}(M_i^{-1})I \end{aligned}, \quad (2)$$

with the minimum and maximum eigenvalues of $\lambda_{min}(M_i)$, $\lambda_{\max}(M_i)$, $\lambda_{min}(M_i^{-1})$ and $\lambda_{min}(M_i^{-1}), i = l, f$.

Assumption 4: [22] Matrix $\dot{M}_i - 2C_i, i=l,f$ is symmetric and C_i is bounded by a quadratic term of the joint velocities $\dot{q}_i$

$$\|C_i(q_i,\dot{q}_i)\dot{q}_i\| \le \|C_i^b(q_i)\|\|\dot{q}_i\|^2, \quad (3)$$

where $C_i^b(q_i)$ is a scalar function, For a robot with all revolute joints, $C_i^b(q_i)$ is constant.

Lemma 1 (Jensen's Integral Inequality) [21] For any constant matrix $\mathrm{M} \in R^{n\times n}, \mathrm{M} = \mathrm{M}^T < 0$, a scalar $\vartheta < 0$, a vector function $w(s): [0, \vartheta] \to n$ such that the integrations concerned are well-defined, then

$$\left[\int_0^\vartheta w(s)ds\right]^T \mathrm{M}\left[\int_0^\vartheta w(s)ds\right] \le \vartheta\int_0^\vartheta w^T(s)\mathrm{M}w(s)ds. \quad (4)$$

B. Dynamic Movement Primitives (DMPs)

The DMPs model proposed by Ijspeert *et al.* [24] is

$$\begin{cases} \tau\dot{z} = K(g-y) - Dz + (g-y_0)f(s) \\ \tau\dot{y} = z \end{cases}, \quad (5)$$

where $K, D > 0$ are stiffness and damping factors and $\tau > 0$ is a timing parameter for adjusting the duration of the trajectory y. y_0 and g are the start and the end position of the trajectory y

and $\dot{y}$ represent the velocity. Generally, in order to enable y converge to g, K and D satisfy $K = 4D^2$ [24]. $f(s) = \theta^T \Psi(s)$ is a combination of normalized Gaussian functions ψ_j, where $\theta = [w_1, w_2, ..., w_n]^T$, $\Psi(s) = [\psi_1, \psi_2, ..., \psi_n]^T$, and w_j is a weight term and the expression of state variable ψ_j is

$$\psi_j = \frac{\varphi_j(s)s}{\sum_{i=1}^{n} \varphi_i(s)}, \varphi_j(s) = \exp(-h_j(s-c_j)^2), \tag{6}$$

where c_j and $h_j > 0$ are the centers and widths of the radial basis function $\varphi_j(s)$. The number of n, and the center c_j and bandwidth h_j can be set automatically by using nonparametric regression technique from locally weighted learning (RFWR) [25], [26]. The transformation function (or named as forcing function) $f(s)$ has a phase variable s, which is calculated by a canonical system

$$\tau \dot{s} = -\gamma s, \qquad \gamma > 0. \tag{7}$$

The converging time is modified by factor γ to make sure $s \to 0$ at the end of trajectory for erasing the influence of $f(s)$. The θ is estimated by minimizing $\left\| f^{Tar}(s) - f(s) \right\|$, where $f^{Tar}(s)$ is calculated by y and z in the demonstration:

$$f^{Tar}(s) = \left(\tau \dot{z} - K(g-y) - Dz\right) / (g - y_0). \tag{8}$$

III. Control Diagram

As illustrated in Fig. 1, the diagram is based on the bilateral control framework of teleoperation, similar to [15]. This control system comprises one observer and one controller on both the leader and follower sides, respectively. The observations serve for robotic autonomy on the follower side and feedback force rendering on the leader side, and are then shared and controlled with delayed feedback from the remote side. The following sub -sections will introduce these modules in sequence.

A. DMPs-based observer

Set $Y = [y \quad z]^T$, then (5) can be rewritten as

$$\dot{Y} = K_1 Y + K_2 + \frac{g - y_0}{\tau} F(s) \tag{9}$$

where $K_1 = \frac{1}{\tau}\begin{bmatrix} 0 & 1 \\ -K & -D \end{bmatrix}$, $K_2 = \frac{1}{\tau}\begin{bmatrix} 0 \\ Kg \end{bmatrix}$ and $F(s) = \begin{bmatrix} 0 \\ f(s) \end{bmatrix}$.

Set $\hat{Y}$ as the estimation of Y, then an observer based on (9) is created as

$$\dot{\hat{Y}} = K_1 \hat{Y} + K_2 + K_n \left(Y(t-d_t) - \hat{Y}(t-d_t) \right) + \frac{g - y_0}{\tau} F^U(s), \tag{10}$$

where the estimation error $\Delta Y = Y - \hat{Y}$ will be compensated by the known errors $Y(t-d_t) - \hat{Y}(t-d_t)$ after receiving $Y(t-d_t)$ after the time delay d_t. K_n is a positive factor and $F^U(s)$ is initialized by (5) and updated in the following calculation.

Since new operational actions may be different from those in

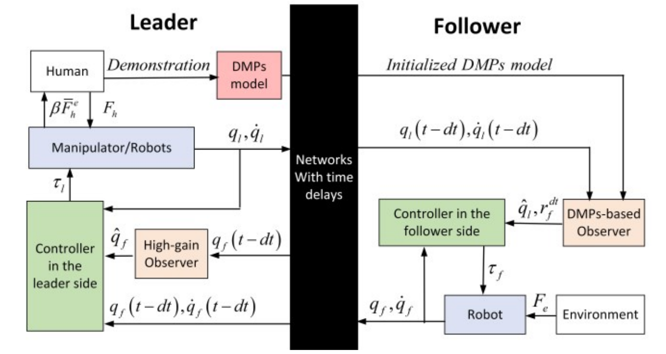

Fig. 1. Illustration of sketch map of system control diagram

demonstration, we express the real-time and the delayed human actions as Y and $Y(t-d_t)$, which can be also expressed by the DMPs with a different $F^N(s)$ as

$$\dot{Y} = K_1 Y + K_2 + \frac{g - y_0}{\tau} F^N(s). \tag{11}$$

Then, according to (10) and (11), we can get

$$\begin{aligned} \dot{Y} - \dot{\hat{Y}} = {} & K_1 \left(Y - \hat{Y} \right) - K_n \left(Y(t-d_t) - \hat{Y}(t-d_t) \right) + \\ & \frac{g - y_0}{\tau} \left(F^N(s) - F^U(s) \right) \end{aligned}, \tag{12}$$

where $F^N(s) - F^U(s)$ represents the difference of two forcing functions. Eq. (12) can be simplified as

$$\Delta \dot{Y} = K_1 \Delta Y - K_n \Delta Y(t-d_t) + \frac{g - y_0}{\tau} \left(F^N(s) - F^U(s) \right). \tag{13}$$

where $\Delta Y = \hat{Y} - Y$. We set $f^U(s)$ and $f^N(s)$ are calculated based on the same kernels $\Psi(s)$, that is

$$\begin{cases} f^N(s) = \left(\theta^N\right)^T \Psi(s) \\ f^U(s) = \left(\theta^U\right)^T \Psi(s) \end{cases}. \tag{14}$$

Then the errors of two forcing functions are expressed as

$$\begin{aligned} F^N(s) - F^U(s) &= \begin{bmatrix} 0 \\ f^N(s) - f^N(s) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \left(\theta^N - \theta^U\right)^T \Psi(s) \end{bmatrix} = \begin{bmatrix} 0 \\ \tilde{\theta}^T \Psi(s) \end{bmatrix} \end{aligned}, \tag{15}$$

and (13) will be updated by

$$\begin{aligned} \Delta \dot{Y} &= K_1 \Delta Y - K_n \Delta Y(t-d_t) + \frac{g - y_0}{\tau} \begin{bmatrix} 0 \\ \tilde{\theta}^T \Psi(s) \end{bmatrix}, \\ &= K_1 \Delta Y - K_n \Delta Y(t-d_t) + G \tilde{\Theta}^T \Psi(s) \end{aligned} \tag{16}$$

where $\tilde{\Theta} = \begin{bmatrix} 0 & \tilde{\theta} \end{bmatrix}^T$ and $G = \frac{g - y_0}{\tau}$. Since in (11), the θ^N is recognized as a desired value for θ^U, then set $\Theta = \begin{bmatrix} 0 & \theta^U \end{bmatrix}^T$ and use (17) to update θ^U to enable $\hat{Y}$ to approach Y:

$$\dot{\Theta} = (g - y_0) \Psi(s) \Gamma \Delta Y, \tag{17}$$

where Γ is a constant matrix as transformation of vector ΔY.

Since Θ desires to converge to $\Theta^N = \begin{bmatrix} 0 & \theta^N \end{bmatrix}^T$, then the parameter estimation error exists $\dot{\tilde{\Theta}} = \dot{\Theta}^N - \dot{\Theta}$ and has

$$\dot{\tilde{\Theta}} = -\left(g - y_0\right)\Psi(s)\Gamma\Delta Y . \tag{18}$$

Using Schur Complement, the sufficient stable condition for the stability of the estimation $\hat{Y}$ is shown in Theorem 1:

Theorem 1: For the observer (10) with a weight updating rate (17), if there exist positive matrices Ξ_1, Ξ_2, Ξ_3 and Q such that the following LMIs holds:

$$\Xi=\begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} \\ * & \Gamma_{22} & \Gamma_{23} \\ * & * & \Gamma_{33} \end{bmatrix} < 0 , \tag{19}$$

where $\Gamma_{11} = \Xi_1 + 2K_1\Xi_3 + \left(\bar{d}_t^2 K_1^2 - I\right)\Xi_2$, $\Gamma_{12} = \Xi_2 - \bar{d}_t^2 K_1 K_n \Xi_2 - K_n\Xi_3$, $\Gamma_{13} = G\left(\bar{d}_t^2 K_1 \Xi_2 + \Xi_3\right) + \Gamma Q\left(g - y_0\right)$, $\Gamma_{22} = \bar{d}_t^2 G K_n \Xi_2$, $\Gamma_{23} = -G\left(K_n\Xi_2 + \left(g - y_0\right)\Gamma Q\tau\right)$, $\Gamma_{33} = \bar{d}_t^2 G^2 \Xi_2$.

Then the estimation error ΔY and $\tilde{\Theta}$ will converge to 0.

Remark 1: The proof of system convergence is presented in Section IV. A. However, it should be noted that the observer (10) is constructed based on the same K_1 and K_2 as those in (11), demonstrated by humans. This implies that the start and end points of the trajectory are known before teleoperation. The update of Θ in (17) is to facilitate the predicted movements $\hat{Y}$ in (10) to track human teleoperation actions Y in (9).

B. High-gain observer in the leader side

As the movement of robots cannot be regularized as humans' movement in DMPs, we build a high-gain nonlinear observer to estimate the actions of robots. According to (1), we have

$$\ddot{q}_f = M_f\left(q_f\right)^{-1}\left(\tau_f - J_f^T(q_f)F_e - C_f\left(q_f, \dot{q}_f\right)\dot{q}_f - G_f\right) . \tag{20}$$

Set $Q = \left[q_f, \dot{q}_f\right]^T$, $\hat{Q} = \left[x_l,\ v_l\right]^T$, and the left formula of $\ddot{q}_f$ in (20) as $\Phi\left(q_f, \dot{q}_f, \tau_f, F_e\right)$, where x_l and v_l are observations of q_f and $\dot{q}_f$ from the leader view, then the desired values of the observations $\hat{Q}^d$ can be expressed as

$$\begin{aligned} \dot{\hat{Q}}^d &= \begin{bmatrix} \dot{x}_l \\ \dot{v}_l \end{bmatrix}^d = \begin{bmatrix} \dot{q}_f \\ \Phi\left(x_l, v_l, \tau_f, F_e\right) \end{bmatrix} \\ &= \begin{bmatrix} v_l \\ M_f\left(x_l\right)^{-1}\left(\tau_f - J_f^T(x_l)F_e - C_f v_l - G_f\right) \end{bmatrix} . \end{aligned} \tag{21}$$

The desired value is $\hat{Q}^d = Q$. Defining $\eta_l = q_f - x_l$ as the estimation errors, the following high-gain observer is

$$\dot{\hat{Q}} = \begin{bmatrix} \dot{x}_l \\ \dot{v}_l \end{bmatrix} = \begin{bmatrix} v_l + \ell k_1 \eta_l^{dt} \\ \hat{\Phi}\left(x_l, v_l, \tau_f, F_e\right) + \ell^2 k_2 \eta_l^{dt} \end{bmatrix}, \tag{22}$$

where ℓ is a high gain of the observer, and k_1 and k_2 are two positive factors. According to [15], the essential Lipschitz-like condition holds for a constant factor $L_\delta > 0$ with Q and its estimation $\hat{Q}$ to ensure the asymptotic stability of the feedback system using the observer in (22) as:

$$\left\| \left[0, \Phi\left(q_f, \dot{q}_f, \tau_f, F_e\right) - \hat{\Phi}\left(x_l, v_l, \tau_f, F_e\right)\right]^T \right\| \le L_\delta \left|Q - \hat{Q}\right| . \tag{23}$$

According to Assumptions 3 and 4 and the properties of the Lagrangian system (1), $M_f\left(q_f\right)^{-1}$, $C_f\left(q_f, \dot{q}_f\right)\dot{q}_f$, $J_f^T(q_f)$ and G_f are bounded. Meanwhile, F_e and control torque τ_l are also bounded. Then, $\Phi\left(q_f, \dot{q}_f, \tau_f, F_h\right)$ and $\hat{\Phi}\left(x_l, v_l, \tau_f, F_e\right)$ are bounded and there exists a L_δ satisfying condition in (23).

Setting $\tilde{Q} = \left[q_f - x_l, \ell^{-1}\left(\dot{q}_f - v_l\right)\right]^T$ and $\bar{K}_e = \begin{bmatrix} -k_1 e^{-dt} & 1 \\ -k_2 e^{-dt} & 0 \end{bmatrix} < 0$, and using (21) and (22), we can get

$$\begin{aligned} \dot{\tilde{Q}} &= \left[\dot{q}_f - \dot{x}_l, \ell^{-1}\left(\ddot{q}_f - \dot{v}_l\right)\right]^T \\ &= \begin{bmatrix} \dot{q}_f - v_l - \ell k_1 e^{-dt}\eta_l \\ \ell^{-1}\left(\Phi\left(q_f, \dot{q}_f, \tau_f, F_e\right) - \hat{\Phi}\left(x_l, v_l, \tau_f, F_e\right) - \ell^2 k_2 \dot{\bar{d}}_t e^{-dt}\eta_l\right) \end{bmatrix} \\ &= \ell \begin{bmatrix} -k_1 e^{-dt} & 1 \\ -k_2 e^{-dt} & 0 \end{bmatrix}\tilde{Q} + \begin{bmatrix} 0 \\ \ell^{-1}\left(\Phi\left(q_f, \dot{q}_f, \tau_f, F_e\right) - \hat{\Phi}\left(x_l, v_l, \tau_f, F_e\right)\right) \end{bmatrix} \\ &= \ell \bar{K}_e \tilde{Q} + \Delta\left(q_f, \dot{q}_f, x_l, v_l, \tau_f, F_e\right) \end{aligned} \tag{24}$$

So, for a high gain $\ell \gg 1$, (24) will be dominated by term $\ell \bar{K}_e \tilde{Q}$, and the term $\Delta\left(q_f, \dot{q}_f, x_l, v_l, \tau_f, F_e\right)$ satisfies

$$\left|\Delta\left(q_f, \dot{q}_f, x_l, v_l, \tau_f, F_e\right)\right| \le \frac{L_\delta}{\ell}\left|\tilde{Q}\right| \le L_\delta \left|\tilde{Q}\right| , \tag{25}$$

to ensure that the estimation error $\tilde{Q}$ converge to 0 finally. The stability conditions and measurements for eliminating negative effects of high gain can refer to [18].

C. Controller design in the follower and leader side

In Sections III.A and B, two observers are built to estimate the timely motions of the agents on both sides in teleoperation. These estimations will be used as current control references in local site. Define two errors in the follower side $e_f := q_f - \hat{q}_f$ and $\eta_f := q_l - \hat{q}_f$ to represent tracking errors to estimations and estimating errors to the desired positions, then we can set two error terms $r_f := \dot{\eta}_f + k_\eta^f \eta_f$, $\varepsilon_f := \dot{e}_f + k_e e_f$, where k_η^f and k_e are constants, then the control torque τ_f in the follower side is designed as

$$\begin{aligned} \tau_f = {} & M_f\left(q_f\right)\left(\ddot{\hat{q}}_f - k_e \dot{e}_f\right) + C_f\left(q_f, \dot{q}_f\right)\left(\dot{\hat{q}}_f - k_e e_f\right) + \\ & G_f - \xi_1 \varepsilon_f - \left(1 - \xi_1\right) r_f^{dt} + J_f^T(q_f)\bar{F}_e + \sigma_f \end{aligned} . \tag{26}$$

where $\bar{F}_e$ represents the measurement of F_e and $\hat{q}_f$ is estimated based on (10) and detailed as:

$$\tau^2 \ddot{\hat{q}}_f = K\left(g - \hat{q}_f\right) - D\tau\dot{\hat{q}}_f - K_n r_f^{dt} + \left(g - q_0\right) f^u\left(s\right) , \tag{27}$$

where $f^u\left(s\right) = \left(\theta^u\right)^T \Psi(s)$ and θ^u is updated by

$$\dot{\theta}^u = \frac{g - q_0}{\tau^2}\Psi(s) r_f , \tag{28}$$

where $\xi_1 \in \left(0, 1\right)$ is a constant factor for shared control. The ξ_1 is for robot autonomy and $1 - \xi_1$ is for human delayed inputs, and their values are determined by Theorem 2. Set the finalized weight learned in (11) as constant θ^n, similar to (18), the factor $\tilde{\theta} := \theta^n - \theta^u$ is updated by

$$\dot{\tilde{\theta}} = -\frac{g-q_0}{\tau^2}\Psi(s)r_f, \tag{29}$$

and σ_f is a saturated term for encountering with the contact force measuring errors:

$$\sigma_f = \phi_f\left(\alpha_f, \varepsilon_f, o_f\right) = \begin{cases} -\varepsilon_f / \|\varepsilon_f\| \cdot \alpha_f, \; if \; \alpha_f \geq o_f \\ -\varepsilon_f / o_f \cdot \alpha_f, \quad if \; \alpha_f < o_f \end{cases}. \tag{30}$$

where α_f satisfies $\left|J_f^T(q_f)\left(\bar{F}_e - F_e\right)\right| \leq \left|J_f^T(q_f)\right|\left|\bar{F}_e - F_e\right| < \alpha_f$ and the $\left|J_f^T(q_f)\right|$ and the force error term $\left|\bar{F}_e - F_e\right|$ are bounded, and o_f denotes a small scalar term. Taking τ_f into (1), we can get

$$\begin{aligned} &M_f\left(\ddot{q}_f - \ddot{\hat{q}}_f + k_e\dot{e}_f\right) + C_f\left(\dot{q}_f - \dot{\hat{q}}_f + k_e e_f\right) \\ &= M_f\dot{\varepsilon}_f + C_f\varepsilon_f \\ &= -\xi_1\varepsilon_f - (1-\xi_1)r_f^{dt} + J_f^T(q_f)\left(\bar{F}_e - F_e\right) + \sigma_f \end{aligned}. \tag{31}$$

The leader controller τ_l is designed with new-defined terms $\eta_l^x := q_f - x_l, \eta_l^v := \dot{q}_f - v_l, e_l := x_l - q_l, \dot{e}_l := v_l - \dot{q}_l, \varepsilon_l := \dot{e}_l + k_e e_l$, and $r_l^{dt} := \eta_l^v(t-dt) + k_\eta^l\eta_l^x(t-dt)$, k_η^l is constant. Then τ_l is

$$\begin{aligned} \tau_l = &M_l(q_l)(\dot{v}_l - k_e\dot{e}_l) + C_l(q_l,\dot{q}_l)(v_l - k_e e_l) + \\ &J_l^T(q_l)\beta\bar{F}_h^e + G_l + \xi_2\varepsilon_l + (1-\xi_2)r_l^{dt} + \sigma_l \end{aligned}, \tag{32}$$

where $\bar{F}_h^e$ represents the haptic force feedback in the leader side, simulated F_e by estimation of $\bar{F}_h^e = \hat{K}_x J_f^T(q_f)e_i + \hat{D}_x\dot{e}_i$, to generate a virtual force generated by position errors e_i, and the stiffness and damping factors $\hat{K}_x$ and $\hat{D}_x$. ξ_3 is a shared control parameter in the leader side, similar to ξ_3 in (26). σ_l is a robust term with $\sigma_l = \phi_l(\alpha_l, \varepsilon_l, o_l)$ and $\left|J_l^T(q_l)\left(\bar{F}_h - F_h\right)\right| < \alpha_l$.

The system stability condition is presented in Theorem 2 and the proof of system stability is presented in Section IV. B.

Theorem 2: For the system (1) with controller (26) and (32), if there exist positive matrices E_1 and E_2 such that the following LMIs holds:

$$\mathrm{E}_1 = \begin{bmatrix} \mathrm{H}_{11} & 0 & \mathrm{H}_{13} \\ * & \mathrm{H}_{22} & \mathrm{H}_{23} \\ * & * & \mathrm{H}_{33} \end{bmatrix} < 0, \mathrm{E}_2 = \begin{bmatrix} \Gamma_{11} & 0 & \Gamma_{13} \\ * & \Gamma_{22} & 0 \\ * & * & \Gamma_{33} \end{bmatrix} < 0, \tag{33}$$

where $\mathrm{H}_{11} = -\xi_1, \mathrm{H}_{22} = k_2 - 2k_1K/\tau^2, \mathrm{H}_{33} = -k_2(1-\mu_t)$, $\mathrm{H}_{13} = (\xi_1 - 1)/2, \mathrm{H}_{23} = k_1K_n/\tau^2, \Gamma_{11} = -\xi_2, \Gamma_{13} = (\xi_2 - 1)/2$, $\Gamma_{22} = 2k_3k_\eta^l + k_4 + 2k_3L_\delta$, $\Gamma_{33} = -k_4(1-\mu_t)$, then the system (1) is robust asymptotic stable with the following parameters $k_\eta^f = \dfrac{D\tau}{\tau^2 + K}$, $k_\eta^l = \dfrac{\ell}{2}\left(\pm\sqrt{k_1^2 - 4k_2} - k_1\right)$ and $k_1, k_2, k_3, k_4 > 0$.

IV. CONVERGENCE AND STABILITY PROOFS

A. Proof of convergence of DMPs-based observations

Set a Lyapunov function $V_o = V_1 + V_2$ for the observer (10):

$$\begin{cases} V_1 = \int\limits_{t-dt}^{t} \Delta Y^T\Xi_1\Delta Y + \int\limits_{-dt}^{0}\int\limits_{t+\theta}^{t} \Delta\dot{Y}^T\Xi_2\Delta\dot{Y} \\ V_2 = \Delta Y^T\Xi_3\Delta Y + \tilde{\Theta}^T Q\tilde{\Theta} \end{cases}. \tag{34}$$

Using $0 \leq \left|\dot{d}_t\right| \leq \mu_t < 1$, then the time derivative of V_1 is

$$\begin{aligned} \dot{V}_1 \leq &\Delta Y^T\Xi_1\Delta Y - \left(1-\dot{d}_t\right)\Delta Y(t-d_t)^T\Xi_1\Delta Y(t-d_t) + \\ &d_t^2\Delta\dot{Y}^T\Xi_2\Delta\dot{Y} - d_t\int\limits_{t+d_t}^{t}\Delta\dot{Y}(\tau)^T\Xi_2\Delta\dot{Y}(\tau)d\tau \\ \leq &\Delta Y^T\Xi_1\Delta Y - (1-\mu_t)\Delta Y(t-d_t)^T\Xi_1\Delta Y(t-d_t) + \\ &d_t^2\Delta\dot{Y}^T\Xi_2\Delta\dot{Y} - d_t\int\limits_{t+d_t}^{t}\Delta\dot{Y}(\tau)^T\Xi_2\Delta\dot{Y}(\tau)d\tau \end{aligned}. \tag{35}$$

Following Lemma 2, we have

$$\begin{aligned} &-d_t\int\limits_{t+d_t}^{t}\Delta\dot{Y}(\tau)^T\Xi_2\Delta\dot{Y}(\tau)d\tau \\ &\leq -\left(\int\limits_{t+d_t}^{t}\Delta\dot{Y}(\tau)d\tau\right)^T\Xi_2\left(\int\limits_{t+d_t}^{t}\Delta\dot{Y}(\tau)d\tau\right) \\ &\leq -\left(\Delta Y(t) - \Delta Y(t-d_t)\right)^T\Xi_2\left(\Delta Y(t) - \Delta Y(t-d_t)\right) \end{aligned}. \tag{36}$$

Using (16), (35) can be further rewritten as

$$\begin{aligned} \dot{V}_1 \leq &\Delta Y^T\Xi_1\Delta Y - (1-\mu_t)\Delta Y(t-d_t)^T\Xi_1\Delta Y(t-d_t) + \\ &d_t^2\left(K_1\Delta Y - K_n\Delta Y(t-d_t) + G\tilde{\Theta}^T\Psi(s)\right)^T \bullet \\ &\Xi_2\left(K_1\Delta Y - K_n\Delta Y(t-d_t) + G\tilde{\Theta}^T\Psi(s)\right) - \\ &\left(\Delta Y - \Delta Y(t-d_t)\right)^T\Xi_2\left(\Delta Y - \Delta Y(t-d_t)\right). \end{aligned} \tag{37}$$

Using (18), the time derivative of V_2 is

$$\begin{aligned} \dot{V}_2 &= 2\Delta\dot{Y}^T\Xi_3\Delta Y + 2\dot{\tilde{\Theta}}^T Q\tilde{\Theta} \\ &= 2\left(K_1\Delta Y - K_n\Delta Y(t-d_t) + G\tilde{\Theta}^T\Psi(s)\right)\Xi_3\Delta Y - \\ &\quad 2(g-y_0)\Delta Y^T Q\tilde{\Theta}\Gamma\Psi(s). \end{aligned} \tag{38}$$

Using (37) and (38), we can get

$$\begin{aligned} \dot{V}_1 + \dot{V}_2 \leq &\Delta Y^T\left(\Xi_1 + 2K_1\Xi_3 + \left(\bar{d}_t^2K_1^2 - I\right)\Xi_2\right)\Delta Y + \\ &\Delta Y(t-d_t)^T\left(\bar{d}_t^2K_n^2\Xi_2 - (1-\mu_t)\Xi_1 - \Xi_2\right)\Delta Y(t-d_t) + \\ &2\Delta Y^T\left(\Xi_2 - \bar{d}_t^2K_1K_n\Xi_2 - K_n\Xi_3\right)\Delta Y(t-d_t) + \\ &2\Psi(s)\tilde{\Theta}\left[\left(G\bar{d}_t^2K_1\Xi_2 + G\Xi_3 + \Gamma Q(g-y_0)\right)\Delta Y - \right. \\ &\left.\bar{d}_t^2GK_n\Xi_2\Delta Y(t-d_t)\right] + \bar{d}_t^2G^2\Xi_2\left(\Psi(s)\tilde{\Theta}\right)^T\Psi(s)\tilde{\Theta} \end{aligned}. \tag{39}$$

Set a vector $\mathrm{Z} = \left[\Delta Y, \Delta Y(t-d_t), \Psi(s)\tilde{\Theta}\right]^T$, then (39) can be expressed as

$$\dot{V}_1 + \dot{V}_2 \leq \mathrm{Z}^T\Xi\mathrm{Z}, \tag{40}$$

where Ξ is represented in (19), then the Theorem 1 is proved.

B. Proof of stability of the shard-control framework

We build the following Lyaponov function

$$V = V_f^e + V_f^p + V_l^e + V_l^p, \tag{41}$$

where V_i^e is for position tracking to $\hat{q}_i$ and V_i^p is for position tracking to the real value $q_i, i=l,f$. Taking the functions in the following controller as an example first, and V_f^e is

$$V_f^e = \frac{1}{2}\varepsilon_f{}^T M_f \varepsilon_f , \tag{42}$$

where M_f is a positive diagonal matrix. The time derivative of V_f^e is

$$\dot{V}_f^e = \dot{\varepsilon}_f{}^T M_f \varepsilon_f + \frac{1}{2}\varepsilon_f{}^T \dot{M}_f \varepsilon_f . \tag{43}$$

Following (31) and using Assumption 4, we have

$$\begin{aligned}\dot{V}_f^e &= \varepsilon_f{}^T\left(-\xi_1\varepsilon_f - (1-\xi_1)r_f^{dt} + J_f^T(q_f)\left(\bar{F}_e - F_e\right) - C_f\varepsilon_f + \sigma\right) + \\ &\quad \varepsilon_f{}^T \dot{M}_f \varepsilon_f / 2 \\ &= -\xi_1\varepsilon_f{}^T\varepsilon_f - (1-\xi_1)\varepsilon_f{}^T r_f^{dt} - \varepsilon_f{}^T J_f^T(q_f)\left(\bar{F}_e - F_e\right) - \varepsilon_f{}^T\sigma \\ &\le -\xi_1\varepsilon_f{}^T\varepsilon_f - (1-\xi_1)\varepsilon_f{}^T r_f^{dt}\end{aligned} . \tag{44}$$

For the tracking errors between the leader and the follower, we refer the Lyaponov function in (34) and set

$$V_f^p = k_1 r_f^T r_f + k_2\int_{t-d_t}^{t} r_f^T r_f \, d\tau + \tilde{\theta}^T k_1 \tilde{\theta} . \tag{45}$$

where $k_1 > 0$. Set $r_f^{dt} := r_f(t-d_t)$, the time derivative of V_f^p is

$$\dot{V}_f^p = 2k_1\dot{r}_f{}^T r_f + k_2 r_f{}^T r_f - k_2\left(1-\dot{d}_t\right)\left(r_f^{dt}\right)^T r_f^{dt} + 2\dot{\tilde{\theta}}^T k_1\tilde{\theta} . \tag{46}$$

As the trajectories $\hat{q}_l$ and q_l are generated by DMPs function in (5), according to (10), we can achieve following equations:

$$\begin{cases}\tau^2\ddot{\hat{q}}_f = K\left(g-\hat{q}_f\right) - D\tau\dot{\hat{q}}_f + \left(g-\hat{q}_0\right)f^u(s) - K_n r_f^{dt} \\ \tau^2\ddot{q}_l = K\left(g-q_l\right) - D\tau\dot{q}_l + \left(g-q_0\right)f^n(s)\end{cases}, \tag{47}$$

where $\hat{q}_0 = q_0$, representing the initial position for estimations and real values are the same. According to the definition of η_f and defining $\Delta f(s) = f^n(s) - f^u(s)$, we have

$$\begin{aligned}\ddot{\eta}_f &= \ddot{q}_l - \ddot{\hat{q}}_f \\ &= \frac{1}{\tau^2}\left(-K\left(q_l - \hat{q}_f\right) - D\tau\left(\dot{q}_l - \dot{\hat{q}}_f\right) + \left(g-q_0\right)\Delta f(s) + K_n r_f^{dt}\right) \\ &= -\frac{K}{\tau^2}\eta_f - \frac{D}{\tau}\dot{\eta}_f + \frac{K_n r_f^{dt}}{\tau^2} + \frac{g-q_0}{\tau^2}\Delta f(s)\end{aligned} . \tag{48}$$

Considering $f^n(s) = \left(\theta^n\right)^T\Psi(s)$ and $f^u(s) = \left(\theta^u\right)^T\Psi(s)$, $\Delta f(s) = f^n(s) - f^u(s) = \left(\theta^n - \theta^u\right)^T\Psi(s) = \tilde{\theta}^T\Psi(s)$, then

$$\ddot{\eta}_f = -\frac{K}{\tau^2}\eta_f - \frac{D}{\tau}\dot{\eta}_f + \frac{K_n r_f^{dt}}{\tau^2} + \frac{g-q_0}{\tau^2}\tilde{\theta}^T\Psi(s) . \tag{49}$$

So for the term $\dot{r}_f = \ddot{\eta}_f + k_\eta^f\dot{\eta}_f$, we have

$$\dot{r}_f = -\frac{K}{\tau^2}\eta_f + \left(k_\eta^f - \frac{D}{\tau}\right)\dot{\eta}_f + \frac{K_n r_f^{dt}}{\tau^2} + \frac{g-q_0}{\tau^2}\tilde{\theta}^T\Psi(s) , \tag{50}$$

If we define factor k_η^f in (50) satisfying $\left(k_\eta^f - \frac{D}{\tau}\right) = -k_\eta^f\frac{K}{\tau^2}$, then we can get $k_\eta^f = \frac{D\tau}{\tau^2+K}$, and (50) can be simplified as

$$\begin{aligned}\dot{r}_f &= -\frac{K}{\tau^2}\eta_f - k_\eta^f\frac{K}{\tau^2}\dot{\eta}_f + \frac{K_n r_f^{dt}}{\tau^2} + \frac{g-q_0}{\tau^2}\tilde{\theta}^T\Psi(s) \\ &= -\frac{K}{\tau^2}r_f + \frac{K_n}{\tau^2}r_f^{dt} + \frac{g-q_0}{\tau^2}\tilde{\theta}^T\Psi(s)\end{aligned} . \tag{51}$$

Then (46) can be expressed as

$$\begin{aligned}\dot{V}_f^p &\le 2k_1\dot{r}_f{}^T r_f + k_2 r_f{}^T r_f - k_2\left(1-\mu_t\right)\left(r_f^{dt}\right)^T r_f^{dt} + 2\dot{\tilde{\theta}}^T k_1\tilde{\theta} \\ &= 2k_1\left(-\frac{K}{\tau^2}r_f + \frac{K_n}{\tau^2}r_f^{dt} + \frac{g-q_0}{\tau^2}\tilde{\theta}^T\Psi(s)\right)^T r_f + \\ &\quad k_2 r_f{}^T r_f - k_2\left(1-\mu_t\right)\left(r_f^{dt}\right)^T r_f^{dt} - 2\frac{g-q_0}{\tau^2}k_1\tilde{\theta}^T\Psi(s) r_f \\ &= \left(k_2 - 2k_1\frac{K}{\tau^2}\right)r_f^T r_f + 2k_1\frac{K_n}{\tau^2}r_f^T r_f^{dt} - k_2\left(1-\mu_t\right)\left(r_f^{dt}\right)^T r_f^{dt}\end{aligned} . \tag{52}$$

So the time derivative of $V_f = V_f^e + V_f^p$ satisfies

$$\begin{aligned}\dot{V}_f &\le -\xi_1\varepsilon_f{}^T\varepsilon_f - \left(1-\xi_1\right)\varepsilon_f{}^T r_f^{dt} + \left(k_2 - 2k_1\frac{K}{\tau^2}\right)r_f^T r_f + \\ &\quad 2k_1\frac{K_n}{\tau^2}\left(r_f^{dt}\right)^T r_f - k_2\left(1-\mu_t\right)\left(r_f^{dt}\right)^T r_f^{dt}\end{aligned} . \tag{53}$$

Similarly, we build Lyapunov functions for the leader side:

$$\begin{aligned}V_l &= V_l^e + V_l^p,\ V_l^e = \frac{1}{2}\varepsilon_l{}^T M_l\varepsilon_l \\ V_l^p &= k_3 r_l^T r_l + k_4\int_{t-d_t}^{t} r_l^T r_l \, d\tau\end{aligned} , \tag{54}$$

and the time derivative of V_l^e and V_l^p are calculated as

$$\dot{V}_l^e = \dot{\varepsilon}_l{}^T M_l\varepsilon_l + \frac{1}{2}\varepsilon_l{}^T\dot{M}_l\varepsilon_l \le -\xi_2\varepsilon_l^T\varepsilon_l - \left(1-\xi_2\right)\varepsilon_l{}^T r_l^{dt} , \tag{55}$$

$$\dot{V}_l^p = 2k_3\dot{r}_l^T r_l + k_4 r_l^T r_l - k_4\left(1-\dot{d}_t\right)\left(r_l^{dt}\right)^T r_l^{dt} . \tag{56}$$

According to the definition of r_l and (24), we have

$$\begin{aligned}\dot{r}_l &= \dot{\eta}_l^v + k_\eta^l\dot{\eta}_l^x , \\ \dot{\eta}_l^x &= \dot{q}_f - \dot{x}_l = \dot{q}_f - v_l - \ell k_1\eta_l^x = \eta_l^v - \ell k_1\eta_l^x , \\ \dot{\eta}_l^v &= \ddot{q}_f - \dot{v}_f = \ell\Delta\left(q_f, \dot{q}_f, x_l, v_l, \tau_f, F_e\right) - \ell^2 k_2\eta_l^x , \\ \dot{r}_l &= \ell\Delta\left(q_f, \dot{q}_f, x_l, v_l, \tau_f, F_e\right) - \ell^2 k_2\eta_l^x + k_\eta^l\left(\eta_l^v - \ell k_1\eta_l^x\right) \\ &= k_\eta^l\eta_l^v - \left(\ell k_1 k_\eta^l + \ell^2 k_2\right)\eta_l^x + \ell\Delta\left(q_f, \dot{q}_f, x_l, v_l, \tau_f, F_e\right)\end{aligned} , \tag{57}$$

Set, $k_\eta^l = \frac{\ell}{2}\left(\pm\sqrt{k_1{}^2 - 4k_2} - k_1\right)$, then we can get

$$\begin{aligned}\dot{r}_l &= k_\eta^l\eta_l^v + \left(k_\eta^l\right)^2\eta_l^x + \ell\Delta\left(q_f, \dot{q}_f, x_l, v_l, \tau_f, F_e\right) \\ &= k_\eta^l r_l + \ell\Delta\Phi\end{aligned} , \tag{58}$$

Using (56) can be expressed as

$$\begin{aligned}\dot{V}_l^p &\le 2k_3\left(k_\eta^l r_l + \ell\Delta\Phi\right)^T r_l + k_4 r_l^T r_l - k_4\left(1-\dot{d}_t\right)\left(r_l^{dt}\right)^T r_l^{dt} \\ &\le \left(2k_3 k_\eta^l + k_4 + 2k_3 L_\delta\right)r_l^T r_l - k_4\left(1-\mu_t\right)\left(r_l^{dt}\right)^T r_l^{dt}\end{aligned} . \tag{59}$$

Set vectors as $\Lambda_f = \left[\varepsilon_f, r_f, r_f^{dt}\right]$ and $\Lambda_l = \left[\varepsilon_l, r_l, r_l^{dt}\right]$, the sufficient condition for $\dot{V} \leq 0$ is $\left(\Lambda_f\right)^T \mathrm{E}_1 \Lambda_f + \left(\Lambda_l\right)^T \mathrm{E}_2 \Lambda_l \leq 0$, where E_1 and E_1 are shown in Theorem 2.

V. EXPERIMENTS

A. Comparative simulations using Omni joystick

We use an Omni haptic phantom to interact with a simulation system built using MATLAB/Simulink for trajectory prediction and teleoperation control. The first simulation is to compare the DMPs-based prediction method with a comparative method for estimating human motion that is deduced from [4]:

$$\dot{\hat{X}}_f = \Upsilon_1\left(\hat{X}_f + X_f\right) + \Upsilon_n\left(X_l(t-d_t) - \hat{X}_f(t-d_t)\right), \quad (60)$$

where $\Upsilon_1 = \begin{bmatrix} -0.3 & 0 \\ 0 & -0.3 \end{bmatrix}$ and $\Upsilon_n = \begin{bmatrix} -20 & 0 \\ 0 & -20 \end{bmatrix}$.

Using a joystick illustrated in Fig. 2(a), we draw a trajectory from [2,3] to [12,10] crossing two obstacles, and the learned results are presented in Fig. 2(b). Figs. 2 (c) to (f) illustrate a new demonstration trajectory (red solid lines) from [4,3] to [11,9]. The trajectories are predicted by (10) and (60) with different time delays, shown as dashed red and solid blue lines. Regarding DMPs learned in the form of (5), the parameters are set $K = 200, D = 28, \tau = 0.01$ to learn $f(s)$ and transfer that to (10) to generate a new trajectory to approach a new target by overcoming two obstacles. For the observers in (10) and (60), we compare the predictive trajectories by choosing time delays as 0.02s, 0.08s, (0.08+0.01sin(t)+ 0.02cos(2t))s and 0.1s.

We can see from Figs.2(c) to (f) that, under the short constant time delay of 0.02s both trajectories estimated by two observers align well with human demonstrations. Upon increasing time delays to 0.08s or even 0.1s, the results of (60) suffer a heavier influence, resulting in larger tracking errors compared to those of (10). The fluctuation in time delays exacerbates the influence of time delays on the prediction accuracy of (60), but has a limited effect on the observer of (10). In addition, the observer of (60) is unable to guide the trajectory to the target and only reaches the surrounding region as indicated by the gray square. In contrast, the DMPs-based observer keeps stable observations, exhibits smaller prediction errors, approaches the target and is less influenced by time delays.

Here, we further compare the predicting complex trajectories of two predictors of writing letters A, C and E with different time delays in Fig.4. We can observe that for the short time delays like 0.02s and 0.1s, both the predicted trajectories have smaller errors to the demonstrations, which are presented in red solid lines. Only for the letter E, the tracking differences to the demonstrations using (60) are smaller than the results using (10). Comparatively, with the increment of time delays to 0.4s, the tracking trajectories of the predictor (60) seriously leave away from the demonstration and do not approach the target finally, while the proposed DMPs-based prediction can reach the destination even though the position tracking errors are much larger than those with small time delays.

The second simulation uses the Omni to get human inputs for a virtual teleoperation system consisting of a 2-DoF robot and a 2-DoF manipulator with parameters: $m_l^1 = 0.12kg, m_l^2 = 0.14kg$ $, m_f^1 = 0.23kg, m_f^2 = 0.46kg, l_f^1 = l_l^1 = 0.3m, l_f^2 = l_l^2 = 0.3m, I_l^1 = 0.01kgm^2, I_l^1 = 0.02kg.m^2, I_f^1 = 0.03kg.m^2, I_f^2 = 0.03kg.m^2$,

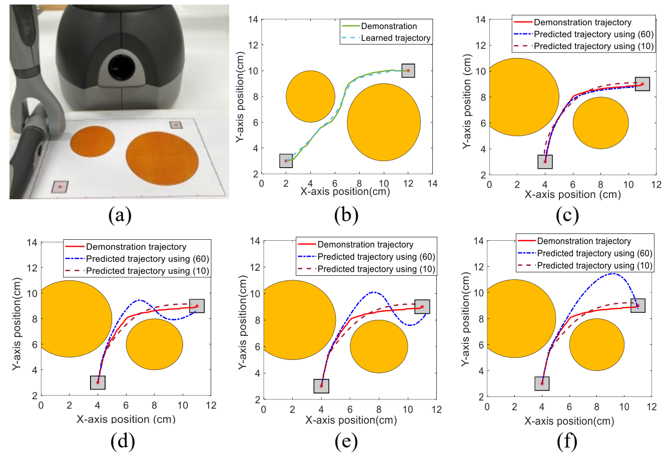

Fig. 2. Comparison of human demonstration and robot prediction trajectories. (a) Experimental setup (b) human demonstration trajectory and learned result using DMPs; (c-f) New task and trajectory predictions in a new environment using (10) (dashed red lines) and (60) (solid blue lines) with the time delays of 0.02s, 0.08s, 0.08+0.01sin(t)+ 0.02cos(2t)s and 0.1s;

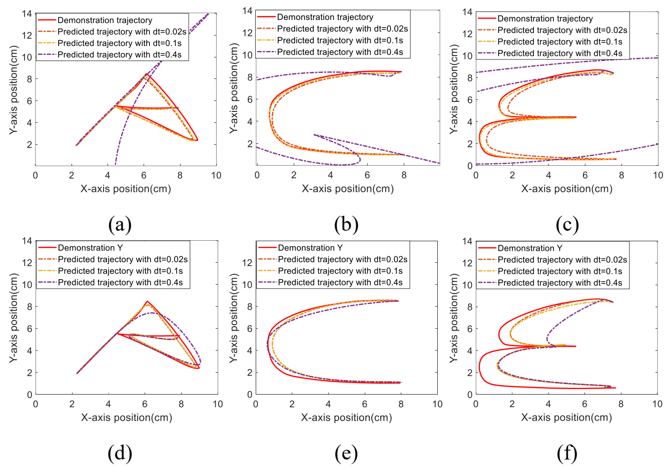

Fig. 3. Comparison of human demonstration and robot trajectories prediction of letters A, C and E with different time delays of 0.02s, 0.1s and 0.4s (a)-(c) Results using the predictor (60) (d)-(f) Results using the predictor (10)

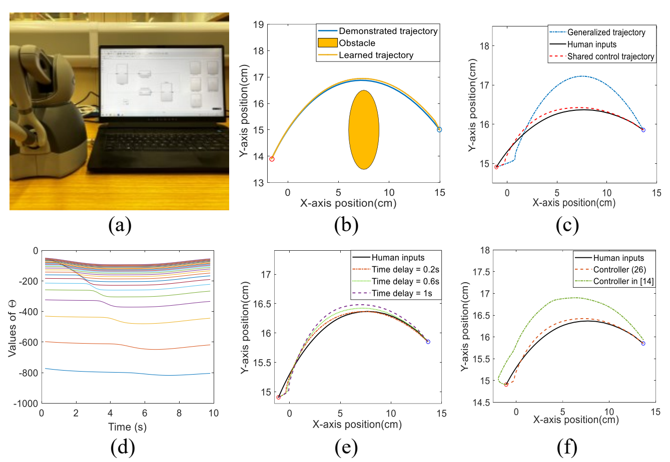

Fig. 4. Teleoperation based on the proposed framework (a) Experimental setup; (b) Human demonstrations; (c) Trajectories in the leader and follower sides with time delays of 0.2+0.05sin(t)+0.02cos(t/2); (d) Parameters changes for the weight vector; (e) Comparisons of trajectories with different time delays 0.2s. 0.6s, and 1.0s; (f) Comparison with the shared controller in [14]

where m_j^i , l_j^i and I_j^i, $i=1,2, j=l,f$ represent the mass, link length and inertia of Link i of Agent j, and the time delay is $d(t)=0.2+0.05sin(t)+0.02cos(t/2)$. Fig. 4(b) shows a demonstration of crossing an obstacle and the learned results using DMPs. The results are then generalized to create a new trajectory (blue dot line) in Fig. 4(c), connecting the start and the end of the new humans trajectory (Black solid line). Using controllers (26) and (32), the robot trajectory is depicted as the red dashed line in Fig. 3(c). Throughout the control process, we set $\xi_1=\xi_2=0.5$ to enable robots and humans to share control authority equally. The elements in the weight vector θ^u in (28) are updated timely and the weight parameters' changes are presented in Fig. 3(d). Fig. 3(e) illustrates the influence of different time delays on the position tracking performance. It can be observed that with the increase of time delays, the position tracking deviations become larger than those with smaller time delays. Fig. 3(f) illustrates a comparison of the proposed method with the shared controller in [14], which is a position-level shared controller integrating TP-HSMM and human inputs with the same sharing coefficient $\xi_1=0.5$ (α=0.5 in [14]). We used the same human inputs as those in the controllers (26) and (32). The generated trajectory (blue lines) deviates away from the human inputs due to the inaccurate predictions of human actions. In contrast, the method in (10) can correct predictions based on delayed human inputs and achieve smaller tracking errors (red dashed line).

B. Experiment on Franka robot

In this experiment, we apply the framework shown in Fig.1 to the Franka robot. Firstly, we demonstrate the Franka robot in a low-stiffness dragging state and hold the handle of the Franka to cross an obstacle as depicted in Fig.5(a). The trajectory is then learned using (5) to acquire the basic skill of overcoming an obstacle. Subsequently, we configure controllers for both the robot and the Omni joystick on the same laptop and simulate the communication channels and time delays through software programming. The control signals and haptic feedback are then published separately to the robot and joystick sides to generate robot actions and haptic force on human hands. Simultaneously, real-time actions of robots and human inputs are collected and delivered to the laptop to generate future control commands. Fig.5(b) presents the process in which an operator teleoperates the Franka to reach a new target position and Fig.5(c) illustrates a close view of the process of crossing a higher obstacle. For

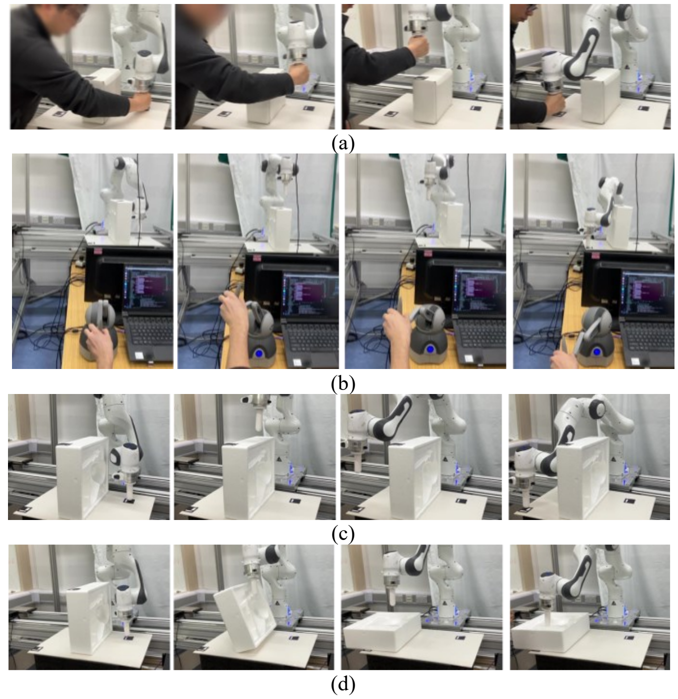

Fig. 5. Human demonstrations and DMPs-based shared control teleoperation (a) Human demonstrations, (b) Human shared control teleoperation (c) Robot execution under the proposed shared control teleoperation (d) Robot execution using autonomous robot control to follow the DMPs-generalized trajectory

comparison, we implement robot autonomous control using the DMPs-based generalized results [27], in a non-sharing control case. As shown in Fig. 5(d), there are conflicts during the robot crossing process and leading to the pushing down the obstacle. Bilateral control can also enable robots to reach to the target without conflicts, but due to the time delays, it requires a longer time to wait for robots to complete actions and feedback.

In Fig.6, we generalized the learned skill and applied that to a more complex scenario, where the robot is required to cross two obstacles and contact position 1 to position 4 in sequence, which are presented in Fig.6(a), and the two obstacles have different heights. We compare the results of using a trajectory generalized by DMPs and a trajectory based on shared control teleoperation with a DMP-based observer and the experimental process is illustrated in Fig.6(b) and (c). There are three crosses during the process in both two cases, the robot can overcome the higher obstacle and contact the designed positions. During the third cross, the robot end controlled by the shared control is closer to the top height of the lower obstacle, which is presented

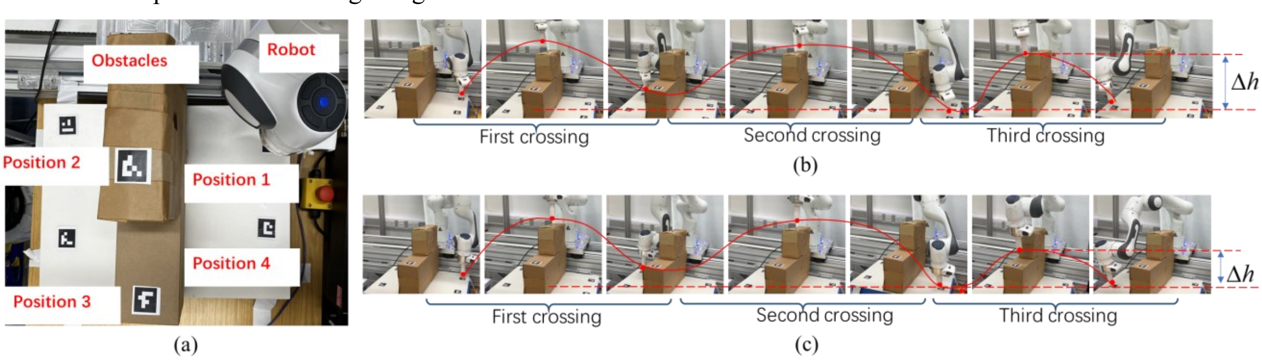

Fig. 6. Human demonstrations and DMPs-based shared control teleoperation (a) Experimental setup, (b) Robot execution using autonomous robot control to follow the DMPs-generalized trajectory (c) Robot execution using shared control teleoperation based on DMP-based observer

as the smaller Δh in Fig.6(c).

We conduct another experiment of writing letters. The robot was controlled in real-time teleoperation to write two letters: 'L' and 'C,' and the motions of the robot's ends were recorded for training purposes. The motions learned from one letter using ordinary DMPs were transferred to write the other letter. New writings of one letter were generated through shared control between the delayed teleoperation motions of the other letter and the generalized DMPs trajectory using equations (26) and (32). The results are presented in Figs. 7(a2) and 7(b2). These results are further detailed in Figs. 7(a3) and 7(b3) to illustrate the influence of time delays more clearly. It is evident that due to the time delays, in the initial stage, the trajectories exhibit varying levels of deviation. The deviations in trajectories with larger time delays (2s) are more significant than those with smaller time delays (0.1s), as depicted in the zoomed figures.

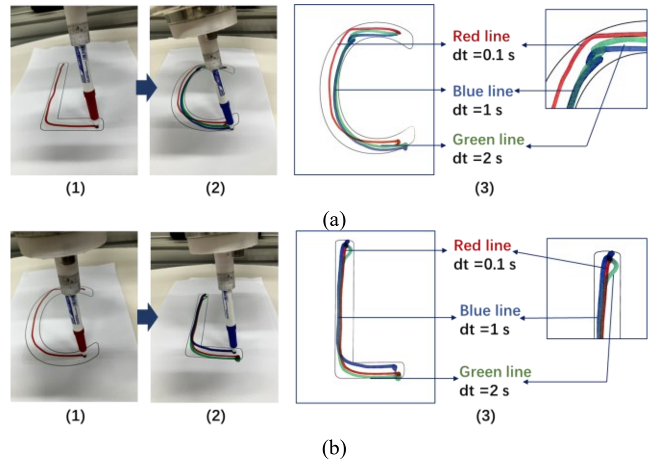

Fig. 7. Human demonstrations and DMPs-based shared control teleoperation (a) trajectory transfer from L to (b) Human shared control teleoperation

VI. Conclusion

In this paper, we develop a DMPs-based observer to predict human motion intentions and apply this observer for the shared control of teleoperation. The experimental results present that, compared with other observers and shared control frameworks, the proposed observer can accurately predict long-term human action intentions and correct prediction errors using the delayed signals to establish consistency between the predicted actions of robots and the actual human operational actions. The DMPs -based observer contains human operational features, ensuring stable operational outputs despite the changes in time delays, This is particularly beneficial for long-distance operations with time delays. We prove the convergence of the estimation and system stability of the control framework with two observers by building two Lyapunov functions as well. Future work has two directions. First, we aim to consider varying sharing factors in (26) based on the objective conditions or requirements of robot manipulation. Second, we aim to extend the DMPs-based shared control teleoperation framework for a wider range, such as multi-robot coordination.

References

[1] Siciliano, B., Khatib, O. and Kröger, T. eds., *Springer handbook of robotics* (Vol. 200). Berlin: springer. 2008.

[2] Z. Chen, F. Huang, C. Yang and B. Yao, "Adaptive fuzzy backstepping control for stable nonlinear bilateral teleoperation manipulators with enhanced transparency performance," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 746-756, Jan. 2020

[3] D. Sun and Q. Liao, "Asymmetric bilateral telerobotic system with shared autonomy control," *IEEE Transactions on Control Systems Technology,* vol. 29, no. 5, pp. 1863-1876, Sept. 2021.

[4] J. Luo, Z. Lin, Y. Li and C. Yang, "A teleoperation framework for mobile robots based on shared control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 377-384, April 2020

[5] S. Islam, P. X. Liu, A. E. Saddik, R. Ashour, J. Dias and L. D. Seneviratne, "Artificial and virtual impedance interaction force reflection-based bilateral shared control for miniature unmanned aerial vehicle," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 329-337. 2019.

[6] Laghi, M., Ajoudani, A., Catalano, M.G. and Bicchi, A., "Unifying bilateral teleoperation and tele-impedance for enhanced user experience", *The International Journal of Robotics Research*, vol. 39, no. 4, pp.514-539, 2020.

[7] Y. Zhu, C. Yang, Z. Tu, Y. Ling and Y. Chen, "A haptic shared control architecture for tracking of a moving object," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 5, pp. 5034-5043, 2023.

[8] de Jonge, A.W., Wildenbeest, J.G., Boessenkool, H. and Abbink, D.A., "The effect of trial-by-trial adaptation on conflicts in haptic shared control for free-air teleoperation tasks", *IEEE transactions on haptics*, vol. 9, no. 1, pp.111-120. 2015.

[9] G. Li, Q. Li, C. Yang, Y. Su, Z. Yuan and X. Wu, "The Classification and New Trends of Shared Control Strategies in Telerobotic Systems: A Survey," *IEEE Transactions on Haptics*, vol. 16, no. 2, pp. 118-133, 2023.

[10] C. Ezeh, P. Trautman, L. Devigne, V. Bureau, M. Babel, and T. Carlson, "Probabilistic vs linear blending approaches to shared control for wheelchair driving," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, 2017, pp. 835–840

[11] M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero and P. R. Giordano, "A shared-control teleoperation architecture for nonprehensile object transportation," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 569-583, 2022.

[12] M. J. A. Zeestraten, I. Havoutis and S. Calinon, "Programming by demonstration for shared control with an application in teleoperation," *IEEE Robotics and Automation Letters,* vol. 3, no. 3, pp. 1848-1855, 2018

[13] A. Gottardi, S. Tortora, E. Tosello and E. Menegatti, "Shared control in robot teleoperation with improved potential fields," *IEEE Transactions on Human-Machine Systems*, vol. 52, no. 3, pp. 410-422, 2022.

[14] B. Xi, S. Wang, X. Ye, Y. Cai, T. Lu, and R. Wang, "A robotic shared control teleoperation method based on learning from demonstrations, " *International Journal of Advanced Robotic Systems*, vol.16, no. 4, pp.1 - 13, 2019.

[15] Z. Lu, P. Huang and Z. Liu, "Predictive approach for sensorless bimanual teleoperation under random time delays with adaptive fuzzy control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 3, pp. 2439-2448, 2018

[16] H. El-Hussieny, S. F. M. Assal, A. A. Abouelsoud and S. M. Megahed, "A novel intention prediction strategy for a shared control tele-manipulation system in unknown environments," in *IEEE International Conference on Mechatronics (ICM)*, Nagoya, Japan, 2015.

[17] K. T. Ly, M. Poozhiyil, H. Pandya, G. Neumann and A. Kucukyilmaz, "Intent-Aware Predictive Haptic Guidance and its Application to Shared Control Teleoperation," in *30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN),* Vancouver, BC, Canada, 2021, pp. 565-572.

[18] Andrieu, V., Prieur, C., Tarbouriech, S. and Zaccarian, L., "A hybrid scheme for reducing peaking in high-gain observers for a class of nonlinear systems, " *Automatica*, no. 72, pp.138-146. 2016.

[19] Z. Li, Y. Xia, D. Wang, D. -H. Zhai, C. -Y. Su and X. Zhao, "Neural network-based control of networked trilateral teleoperation with geometrically unknown constraints," *IEEE Transactions on Cybernetics*, vol. 46, no. 5, pp. 1051-1064, 2016

[20] Z. Lu, P. Huang and Z. Liu, "Relative impedance-based internal force control for bimanual robot teleoperation with varying time delay," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 1, pp. 778-789, 2020,

[21] Z. Li, Y. Xia and F. Sun, "Adaptive fuzzy control for multilateral cooperative teleoperation of multiple robotic manipulators under random network-induced delays," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 2, pp. 437-450, 2014,

[22] J. I. Mulero-MartÍnez, "Uniform bounds of the Coriolis/Centripetal matrix of serial robot manipulators," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1083-1089, Oct. 2007
[23] S. Heshmati-alamdari, A. Nikou, K. J. Kyriakopoulos, and Dimarogonas, D.V., "A robust force control approach for underwater vehicle manipulator systems, " *IFAC-PapersOnLine*, vol 50, no.1, pp.11197-11202. 2017.
[24] Ijspeert, A. J., Nakanishi, J., Hoffmann, H., Pastor, P. and Schaal, S., "Dynamical movement primitives: learning attractor models for motor behaviors, " *Neural computation*, vol 25, no. 2, pp.328-373. 2013.
[25] Schaal, S., "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive motion of animals and machines*. Tokyo: Springer Tokyo, pp. 261-280, 2006.
[26] S. Schaal and C. G. Atkeson, "Constructive incremental learning from only local information," *Neural Comp.,* vol. 10, pp. 2047-2084, 1998.
[27] C. Yang, C. Zeng, Y. Cong, et al., "A Learning Framework of Adaptive Manipulative Skills From Human to Robot," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1153-1161,. 2019

Zhenyu Lu (M’21) He is currently working as a senior research fellow in the Bristol Robotic Laboratory & University of the West of England, Bristol. He received the Ph.D. degree in Northwestern Polytechnical University, Xi’an, China in 2019. His research interests include teleoperation, human–robot interaction and intelligent learning method.

Weiyong Si received the M.S. degree from the Beijing Institute of Technology in 2018. He is currently pursuing the PhD degree in Bristol Robotics Laboratory and the University of the West of England. His research interests include robot skill learning, teleoperation, and robot control.

Ning Wang (S’07-M’11) is a Senior Lecturer in Robotics at the Bristol Robotics Laboratory, University of the West of England, United Kingdom. She received the M.Phil. and Ph.D. degrees in electronics engineering from the Department of Electronics Engineering, The Chinese University of Hong Kong, Hong Kong, in 2007 and 2011, respectively. Her research interests lie in signal processing, intelligent data analysis, human-robot interaction and autonomous driving.

Chenguang Yang (M'10-SM'16-F’23) received the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010, and postdoctoral training in human robotics from the Imperial College London, London, U.K. As the lead author, he won the IEEE Transactions on Robotics Best Paper Award (2012) and IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award (2022). He is the also Corresponding Co-Chair of IEEE Technical Committee on Collaborative Automation for Flexible Manufacturing, a Fellow of Institute of Engineering and Technology (IET), and a Fellow of British Computer Society (BCS). His research interest lies in human robot interaction and intelligent system design.