SCIENCE CHINA
Information Sciences

• LETTER •

November 2024, Vol. 67, Iss. 11, 219201:1–219201:2
https://doi.org/10.1007/s11432-023-4176-0

Teleoperation with automatic posture regulation and broad learning control for assembly tasks

Liwen SITU[1], Zhenyu LU[2] & Chenguang YANG[1*]

[1]*School of Automation Science and Engineering, South China University of Technology, Guangzhou* 510641*, China;*
[2]*Bristol Robotics Laboratory, University of the West of England, Bristol* BS34 8QZ*, UK*

Received 9 November 2023/Revised 6 June 2024/Accepted 10 August 2024/Published online 15 October 2024

In recent years, robot teleoperation systems have been widely used in assembly tasks due to their ability to combine human control with robotic execution, addressing limitations in precision, perception, and intelligence faced by fully automated robots. However, robot teleoperation for assembly tasks still encounters several challenges. In complex and unstructured environments, telerobots must regulate the posture of their manipulators to avoid obstacles and singularities, ensuring the safety of human operators [1]. Moreover, to ensure the effective execution of precise assembly tasks and to achieve rapid and accurate posture regulation, the tracking performance of the teleoperation system must be guaranteed [2].

To solve the abovementioned problems, an assembly teleoperation system based on shared control and a broad learning neural network is designed to reduce operator workload, enhance task efficiency, and ensure robot tracking performance, as shown in Figure 1. The proposed system comprises three controllers: the null-space controller for posture regulation, the barrier Lyapunov function-based joint controller for ensuring the predefined transient tracking performance, and the adaptive torque controller for compensating dynamic uncertainty and achieving precision assembly.

The main contributions of this study are listed below.

(1) Three functions for posture regulation are introduced: manipulability optimization, obstacle avoidance, and joint limit prevention, which are integrated through shared control, allowing telerobots to adjust joint positions automatically while the operator concentrates solely on assembly manipulation.

(2) A barrier Lyapunov function (BLF)-based joint controller is employed to confine transient tracking errors within a prescribed region.

(3) An improved broad learning neural network (BLNN)-based torque controller is developed. Compared to the traditional radial basis function neural network, which depends on empirically preset nodes, the BLNN dynamically generates nodes, making it more accurate for estimating manipulator uncertainties and reducing computation time.

Posture regulation. To minimize kinematic energy consumption during the manipulability optimization process, we define the manipulability performance criterion $C(\boldsymbol{\theta})$ as $\frac{1}{2}\dot{\boldsymbol{\theta}}^{\mathrm{T}}\dot{\boldsymbol{\theta}} - \boldsymbol{M}_d^{\mathrm{T}}\dot{\boldsymbol{\theta}} + \frac{1}{2}\boldsymbol{M}_d^{\mathrm{T}}\boldsymbol{M}_d$, where $\boldsymbol{M}_d$ is the derivative of joint angles $\boldsymbol{\theta}$ with respect to the manipulability measure. Thus, the optimizing problem can be transformed into minimizing $C(\boldsymbol{\theta})$ using the gradient descent technique to obtain the negative gradient of $C(\boldsymbol{\theta})$, denoted as $\nabla C(\boldsymbol{\theta})$. To avoid excessive joint accelerations, the convergence rate coefficient is designed as $K_m(t) = \frac{\exp(0.5t)-\exp(-0.5t)}{\exp(0.5t)+\exp(-0.5t)}$, where t is time variable. Consequently, the vector $\boldsymbol{z}_m$ for the manipulability optimization function is given by

$$\boldsymbol{z}_m = K_m \nabla C(\boldsymbol{\theta}). \tag{1}$$

In complex and unstructured environments, a point-cloud-based obstacle detection algorithm is introduced [3], leveraging the K-means clustering and the robot skeleton modeling to identify collision points, denoted as $\boldsymbol{p}_o$ for the obstacle and $\boldsymbol{p}_r$ for the robot arm. Hence, the desired collision avoidance velocity can be obtained as $\dot{\boldsymbol{x}}_o = \boldsymbol{p}_r - \boldsymbol{p}_o$. Additionally, to revert the manipulator to its initial posture after obstacle removal, a parallel inverse kinematic system provides the joint restoring velocity $\dot{\boldsymbol{x}}_r = \zeta \boldsymbol{e}_r$, where ζ is a positive coefficient and $\boldsymbol{e}_r$ denotes the position error between the parallel system and the real system. Thus, the vector $\boldsymbol{z}_o$ for obstacle avoidance and the vector $\boldsymbol{z}_r$ for posture restoration are derived and combined into $\boldsymbol{z}_{or}$ as

$$\begin{cases} \boldsymbol{z}_o = [\boldsymbol{J}_o(\boldsymbol{I}_6 - \boldsymbol{J}_e^{\dagger}\boldsymbol{J}_e)]^{\dagger}(\dot{\boldsymbol{x}}_o - \boldsymbol{J}_o\boldsymbol{J}_e^{\dagger}\dot{\boldsymbol{x}}_e), \\ \boldsymbol{z}_r = [\boldsymbol{J}_r(\boldsymbol{I}_6 - \boldsymbol{J}_e^{\dagger}\boldsymbol{J}_e)]^{\dagger}(\dot{\boldsymbol{x}}_r - \boldsymbol{J}_r\boldsymbol{J}_e^{\dagger}\dot{\boldsymbol{x}}_e), \\ \boldsymbol{z}_{or} = K_o\boldsymbol{z}_o + (1-K_o)\boldsymbol{z}_r, \end{cases} \tag{2}$$

where $\boldsymbol{J}_e$, $\boldsymbol{J}_e$, and $\boldsymbol{J}_r$ are the Jacobian matrices of the end-effector, the collision point $\boldsymbol{p}_r$, and the restoration point, respectively. $\boldsymbol{I}_6$ is a 6×6 identity matrix, † indicates the Moore-Penrose pseudo-inverse, $\dot{\boldsymbol{x}}_e$ is the real velocity vector, and K_o is a variable parameter dependent on the distance between the robot arm and the obstacle.

To prevent configurations of the manipulator from getting close to its mechanical joint limits, a vector $\boldsymbol{z}_j$ is designed. We define $\theta_{\max,i}$ ($\theta_{\min,i}$) as the maximum (minimum) angle and $q_{\max,i}$ ($q_{\min,i}$) as the user-defined maximum (minimum) angle for the ith joint, forming joint buffer zones. If the joint angles enter the ranges $(\boldsymbol{q}_{\max}, \boldsymbol{\theta}_{\max}]$ or $[\boldsymbol{\theta}_{\min}, \boldsymbol{q}_{\min})$, $\boldsymbol{z}_j$ increases, guiding the joint towards the safe interval $[\boldsymbol{q}_{\min}, \boldsymbol{q}_{\max}]$. Otherwise, $\boldsymbol{z}_j$ remains $\boldsymbol{0}$ and does not affect the manipulator's motion.

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

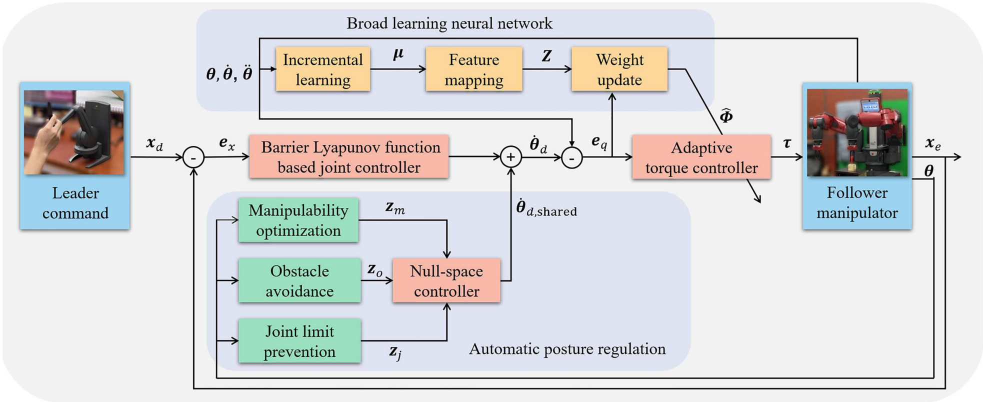

Figure 1 (Color online) Block framework of the proposed assembly teleoperation system.

To achieve automatic posture regulation, the principle of shared control is introduced. The three functions—manipulability optimization, obstacle avoidance, and joint limit prevention—are integrated by exploiting the null space of $\boldsymbol{J}_e$ to maintain the commanded robot arm's pose in the task space, formulated as

$$\dot{\boldsymbol{\theta}}_{d,\text{shared}} = (\boldsymbol{I}_6 - \boldsymbol{J}_e^{\dagger}\boldsymbol{J}_e)(w_1\boldsymbol{z}_m + w_2\boldsymbol{z}_{or} + w_3\boldsymbol{z}_j), \tag{3}$$

where w_1, w_2, and w_3 are positive weighting factors specified by users, reflecting the relative importance of each function in optimizing posture adaptation.

BLF-based joint controller. In this study, the desired end-effector command $\boldsymbol{x}_d$ for the robot arm is generated from the leader site. To ensure transient tracking performance and confine tracking errors within a prescribed region, a symmetric BLF with an exponential decaying error constraint [4] is selected for the joint controller. By taking the time derivative of this BLF, the virtual control law $\dot{\boldsymbol{\theta}}_d$, which considers the posture regulation component as formulated in Eq. (3), is derived as

$$\dot{\boldsymbol{\theta}}_d = \boldsymbol{J}_e^{\dagger}(\dot{\boldsymbol{x}}_d + \boldsymbol{\Gamma} - \boldsymbol{K}_1\boldsymbol{e}_x) + \dot{\boldsymbol{\theta}}_{d,\text{shared}}, \tag{4}$$

where $\boldsymbol{K}_1$ is a 6×6 diagonal positive definite matrix, $\boldsymbol{e}_x = \boldsymbol{x}_e - \boldsymbol{x}_d$ represents the task space error, and $\boldsymbol{\Gamma}$ is the transient tracking vector derived from the error constraint.

BLNN-based torque controller. To compensate for dynamic uncertainty and further enhance tracking performance, the compensation torque is defined as $\boldsymbol{\Phi} = \boldsymbol{M}\ddot{\boldsymbol{\theta}} + \boldsymbol{C}\dot{\boldsymbol{\theta}} + \boldsymbol{G} - \ddot{e}_q$, where $\boldsymbol{M}$, $\boldsymbol{C}$, and $\boldsymbol{G}$ are the robot arm's dynamic parameters and $\boldsymbol{e}_q = \boldsymbol{\theta} - \boldsymbol{\theta}_d$. Unlike [5], which requires separate estimation of different dynamic parameters, resulting in long convergence time, we regard them as a unified entity, reducing computation time and enhancing responsiveness for teleoperation assembly tasks. Thus, the torque controller can be designed as

$$\boldsymbol{\tau} = \hat{\boldsymbol{\Phi}} - (\boldsymbol{K}_{2d}\dot{\boldsymbol{e}}_q + \boldsymbol{K}_{2p}\boldsymbol{e}_q) + \boldsymbol{\tau}_{\text{ext}}, \tag{5}$$

where $\boldsymbol{\tau}$ and $\boldsymbol{\tau}_{\text{ext}}$ are the control and the external torque, $\hat{\boldsymbol{\Phi}}$ is the estimate of $\boldsymbol{\Phi}$ and the matrices $\boldsymbol{K}_{2d}$ and $\boldsymbol{K}_{2p}$ represent the proportional-derivative parameters.

We use the BLNN with $\boldsymbol{\theta}$ as the input vector to estimate $\boldsymbol{\Phi}$, yielding $\hat{\boldsymbol{\Phi}} = \hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{Z}$, where $\boldsymbol{Z}$ is the feature vector that covers the mapping layer and the enhanced layer with weights $\hat{\boldsymbol{W}}$. For the BLNN's update law, the initial joint configuration is set as the first node. In subsequent sampling periods T, if the distance between the current joint position and its neighboring nodes is below the threshold, nodes remain unaltered. Conversely, a new node $\boldsymbol{\mu}_{\text{new}}$ is added based on the current joint position, with updates to $\boldsymbol{Z}(t+T)$ and its weight $\boldsymbol{W}(t+T)$.

Conclusion. This study presents a robot teleoperation system for precise assembly tasks in unstructured environments. To enhance the robots' flexibility, adaptability and safety, an automatic posture regulation strategy is employed. To improve the robots' tracking accuracy and stability, a BLF-based joint controller and a BLNN-based adaptive torque controller are implemented. Additionally, we conducted a series of experiments and a comprehensive assembly task to validate the effectiveness of the proposed system (see Appendix E). In future work, we aim to enhance the posture regulation component, focusing on achieving continuous joint trajectories during obstacle avoidance in environments with multiple obstacles.

Acknowledgements This work was supported in part by the National Nature Science Foundation of China (Grant Nos. U20A20200, 92148204) and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023B1515120019).

Supporting information Appendixes A–E. The supporting information is available online at info.scichina.com and link.springer.com. The supporting materials are published as submitted, without typesetting or editing. The responsibility for scientific accuracy and content remains entirely with the authors.

References

1 Zhang Y, Yang C, Xu S, et al. Obstacle avoidance in human-robot cooperative transportation with force constraint. Sci China Inf Sci, 2023, 66: 119205
2 Zhang Y, Li M, Yang C. Robot learning system based on dynamic movement primitives and neural network. Neurocomputing, 2021, 451: 205–214
3 Wang X, Yang C, Ju Z, et al. Robot manipulator self-identification for surrounding obstacle detection. Multimed Tools Appl, 2017, 76: 6495–6520
4 Yang C, Huang D, He W, et al. Neural control of robot manipulators with trajectory tracking constraints and input saturation. IEEE Trans Neur Netw Learn Syst, 2021, 32: 4231–4242
5 Huang H, Zhang T, Yang C, et al. Motor learning and generalization using broad learning adaptive neural control. IEEE Trans Ind Electron, 2019, 67: 8608–8617