Deep reinforcement learning-based variable impedance control for grinding workpieces with complex geometry

Yanghong Li, Yahao Wang, Zhen Li, Lv Yingxiang, Jin Chai and Erbao Dong
Department of Precision Machinery and Precision Instrumentation, State Key Laboratory of Precision and Intelligent Chemistry, CAS Key Laboratory of Mechanical Behavior and Design of Materials, Humanoid Robotics Institute, University of Science and Technology of China, Hefei, China

Abstract

Purpose – This paper aims to design a deep reinforcement learning (DRL)-based variable impedance control policy that supports stability analysis for robot force tracking in complex geometric environments.

Design/methodology/approach – The DRL-based variable impedance controller explores and pre-learns the optimal policy for impedance parameter tuning in simulation scenarios with randomly generated workpieces. The trained results are then used as feedforward inputs to improve the force-tracking performance of the robot during contact. Based on Lyapunov's theory, the stability of the proposed control policy is analysed to illustrate the interpretability of the results.

Findings – Simulations and experiments are performed on different types of complex environments. The results show that the proposed method is not only theoretically feasible but also has better force-tracking effects in practice.

Originality/value – Compared with most other DRL-based control policies, the proposed method possesses stability and interpretability, effectively avoids the overfitting phenomenon and thus has better simulation-to-real deployment results.

Keywords Robotic machining, Model learning for control, Compliance and impedance control, Force tracking in workpieces

Paper type Research paper

1. Introduction

With the development of manipulators (Chien-Chern, 1998; Zheng, 2023), exoskeletons (Liu *et al.*, 2022), quadruped dogs (Wu *et al.*, 2023), etc., robots are widely used in various non-structured environments to perform contact tasks such as sanding, polishing, peg-in-hole assembly and human–robot interaction. Compliant control is necessary to ensure operation accuracy and safety (Ko *et al.*, 2022). As a kind of classical compliant control, impedance control has been a hot research topic since its proposal. The core idea of impedance control is establishing a dynamic relationship between force and position, so that the robot is compliant when it contacts a rigid environment. This effectively prevents the robot or the environment (human) from being damaged during the interaction. However, the classical constant impedance control has inherent limitations when facing complex and nonlinear environments. When in a low impedance state, the robot's switching from no-contact to contact with the environment often causes oscillations or even instability. When in a high impedance state, the robot's dynamic responsiveness to changes in the environment is reduced, let alone facing complex environments (Wu *et al.*, 2023). Therefore, the key to stabilizing force contact between the robot and the environment lies in the tuning of impedance parameters (Vanderborght *et al.*, 2013; Abu-Dakka and Matteo, 2020).

In the literature, research on variable impedance control for force-tracking task in complex geometric environments can be divided into the following two categories (Siciliano and Villani, 2000; Li, 2018):

1.1 Methods based on system analysis

The straightforward idea is to apply adaptive control law or optimization theory to update the impedance parameters based on the analysis of the controlled system. Duan *et al.* (2018) proposed an adaptive control for force-tracking task, which tunes the impedance parameters to adapt to the environmental changes through the force error. Kong and Huang (2023) used fuzzy logic analysis to reason about the force error affiliation in different states, which improved the robustness of the force-tracking process. Song *et al.* (2024) designed an adaptive control that corrects the grinding path by recognizing the

The current issue and full text archive of this journal is available on Emerald Insight at: **https://www.emerald.com/insight/2754-6969.htm**

Robotic Intelligence and Automation
45/1 (2025) 159–172
© Emerald Publishing Limited [ISSN 2754-6969]
[DOI 10.1108/RIA-09-2024-0207]

Funding: This research was supported by the National Key R&D Program of China (Grant number: 2018YFB1307400) and the Fundamental Research Funds for the Central Universities.

Received 12 July 2024
Revised 12 November 2024
20 December 2024
Accepted 20 December 2024

workpiece roughness online. In addition, many other scholars have proposed different control methods based on convex optimization theory (Deng *et al.*, 2022), iterative learning theory (Hailong *et al.*, 2022) or other adaptive control theories. However, most of these approaches aim to propose a generalized control method while simplifying or ignoring the dynamic characteristics of the robot. This leads to unavoidable tracking errors during the robot's interaction with the complex environment and thus is rarely used in recent work.

1.2 Methods based on data driven

The basic idea is to generate impedance tuning policies from a large number of data samples (Okada, 2023). Methods include two main types:

1 learning from demonstration (LfD) or imitation learning; and
2 reinforcement learning (RL).

Biomechanics showed that human neural networks can tune muscle tension to adapt to complex environments (Prendergast *et al.*, 2021). LfD looks to transfer this ability to robots. Based on this, Yang *et al.* (2018) collected electrical muscle signals to mimic the human process of tuning impedance gain, but the noise brought by the tremor leads to poor localization accuracy and is not suitable for fine contact tasks. Most LfD-based approaches rely on human demonstration data, and the differences in structure and actuation between human arms and robots make the learning process always biased or even negatively optimized.

An alternative approach to data-driven-based variable impedance control is RL, especially deep reinforcement learning (DRL) that introduces deep networks to estimate the value of state actions. The DRL-based agent requires only a reward function and explores the optimal behaviour for reward maximization through iterative trials. Beltran (2020) trained an RL agent for a peg-in-hole assembly manipulation task, which was validated on both simulation and real robots. Wu (2022a) used DRL agent to automatically configure the impedance parameters of the exoskeleton's knee joint for continuous walking on different terrains. Gangapurwala *et al.* (2022) proposed a DRL algorithm for quadrupedal control for walking, running and crossing on uneven environment. Although DRL-based methods have achieved good results in various control tasks, there is a need to avoid control instability that may result from differences between sample data and real robots.

In summary, the first category of methods is adaptive variable impedance control based on specific theory or *a priori* knowledge. However, this category lacks the self-learning ability to update the control policy in the face of new environments, resulting in very limited applicable environments (Seo *et al.*, 2023). The second category of methods is the data-driven method with learning capability, but the stability analysis and interpretability of the results need to be addressed to avoid the overfitting phenomenon in complex environments (Zhang *et al.*, 2023).

Aiming at the above problems, this paper proposes a DRL-based variable impedance control method applicable to complex geometric environments. By analysing the dynamic characteristics of the robot interacting in the complex environment, stability bounds are set for the exploratory actions based on Lyapunov's theory, which ensures the safe transfer of the learning results. To improve the learning efficiency of DRL policy networks in randomly generated environments, an energy loss term is included in the reward function to avoid impedance parameter oscillations.

The main contributions of this paper include theoretical and practical aspects:

- A DRL-based variable impedance control policy that supports stability analysis is proposed. The exploration boundary is set for the policy network based on the stability analysis, which takes into account the control effect while effectively avoiding the overfitting phenomenon of the deep network.
- For the force-tracking task of complex geometric workpieces, the proposed control policy shows better simulation-to-real deployment results than the classical control methods.

The rest of the paper is organized as follows. Section 2 details our proposed DRL-based variable impedance controller and constructs the force tracking formulation for the robot in complex geometric workpieces. Section 3 analyses the stability of the proposed control algorithm. Section 4 describes the *CoppeliaSim*-based simulations and real experiment results analysis. Section 5 gives the conclusion of this paper and discusses some further work.

2. Our method

In this section, we first formulate the equivalent model of our research problem. Then, we design our DRL-based variable impedance controller, which aims to serve for force-tracking tasks in unknown environments. Finally, we present the proposed controller reward function design and update policy.

2.1 Problem formulation

Consider a non-redundant manipulator with $n = 6$ revolute joints, the kinetic equation in joint space under contact force constraints (Campeau, 2019) are given by:

$$\overline{M}(q)\ddot{q} + \overline{C}(q,\dot{q})\dot{q} + g(q) = \tau - \tau_{\text{ext}} \qquad (1)$$

where $q \in \mathbb{R}^n$ is the joint angles, $\overline{M}(\cdot) \in \mathbb{R}^{n\times n}$ is the inertia matrix of the manipulator, $\overline{C}(\cdot,\cdot) \in \mathbb{R}^{n\times n}$ is the centrifugal and Coriolis force terms, $g(\cdot) \in \mathbb{R}^n$ is the gravity terms, $\tau \in \mathbb{R}^n$ is the control input torque and $\tau_{\text{ext}} \in \mathbb{R}^n$ is the external torque induced by external constraints.

Within the non-singular region of the manipulator, we chose the n-dimensional finite task space satisfying one-to-one mapping to study the contact motion. Let $x \in \mathbb{R}^n$ denote the end-effector position vector mapped from the joint space to the workspace, defined as $x = Map(q)$. Then, the derivation of x yields:

$$\dot{x} = \mathcal{J}(q)\dot{q} \tag{2}$$

$$\ddot{x} = \mathcal{J}(q)\ddot{q} + \dot{\mathcal{J}}(q)\dot{q} \tag{3}$$

where $\mathcal{J}(\cdot)\partial Map(\cdot)/\partial q \in \mathbb{R}^{n\times n}$ is the end-effector Jacobian matrix, describing the relationship between Cartesian velocity to joint velocity.

If the end-effector contacts a workpiece with elastic modulus, the stiffness relationship between the external force $f_{\text{ext}} \in \mathbb{R}^n$ at the contact point and x is described by:

$$f_{\text{ext}} = K_{\text{env}}(x - x_{\text{env}}) \tag{4}$$

where $K_{\text{env}} \in \mathbb{R}^{n\times n}$ denotes the positive definite matrix describing the stiffness of the environment (i.e. workpiece) and the vector $x_{\text{env}} \in \mathbb{R}^n$ is the rest position of the environment.

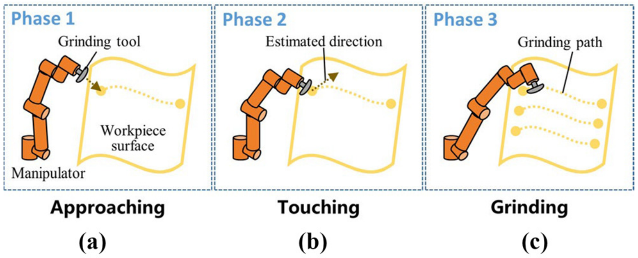
In this paper, we assume that the workpiece stiffness K_{env} and the rest position x_{env} cannot be known accurately. And from the principle of virtual work, the relationship between the external torque τ_{ext} under the joint space in equation (1) and the contact force f_{ext} under the task space in equation (4) satisfies $\tau_{\text{ext}} = \boldsymbol{J}^{\text{T}} f_{\text{ext}}$. Typically, the objective of impedance modelling is to establish a relationship between position error and force error. Variable impedance control does not directly track position trajectories or forces; rather, it indirectly controls position or forces by tuning the impedance relationship. The desired impedance relationship is established as follows:

$$f_{\text{ext}} - f_{\text{d}} = M_{\text{d}}(\ddot{x} - \ddot{x}_r) + B_{\text{d}}(\dot{x} - \dot{x}_r) + K_{\text{d}}(x - x_r) \tag{5}$$

where $t > 0$ denotes the time interval, $M_{\text{d}} \in \mathbb{R}^{n\times n}$, $B_{\text{d}} \in \mathbb{R}^{n\times n}$ and $K_{\text{d}} \in \mathbb{R}^{n\times n}$ are the desired inertia matrix, damping matrix and stiffness matrix in impedance model, respectively, $\ddot{x}_{\text{r}} \in \mathbb{R}^n$, $\dot{x}_{\text{r}} \in \mathbb{R}^n$ and $x_{\text{r}} \in \mathbb{R}^n$ are the reference acceleration, velocity and position of the manipulator in the task pace, respectively, and $f_{\text{d}} \in \mathbb{R}^n$ is the desired force.

A schematic diagram of a simplified contact model of a robot with a rigid unstructured environment is given in Figure 1. Most robotic contact tasks involve constraint switching from free motion to contact motion. Therefore, the contact process between the end-effector and the workpiece is divided into three phases. In the first phase, the end-effector is in unconstrained state and approaching the workpiece. In the second phase, the end-effector starts to contact the workpiece. After a short non-linear oscillation, a steady state is reached. In the third phase, the end-effector steadily contacts the workpiece and performs the task. As the environment is unknown, we consider workpiece geometry variations in the third phase to improve the robustness and generality of the policy network.

Figure 1 Grinding process of the robot in the complex geometry environment



Notes: (a) Approaching; (b) touching; (c) grinding
Source: Authors own work

Facing the challenges posed by multiple control processes, this paper aims to develop a unified learning controller for force-tracking tasks without complex switching.

2.2 Deep reinforcement learning-based variable impedance control

The environment considered in this paper is a more generalized workpiece. The uncertainty of the environment makes expressions for contact forces unusable or difficult to handle.

DRL-based control formulates the control process as a finite-state Markov decision process (MDP), which does not require full process information about the controlled system and the environment (Beltran, 2020). The MDP for RL typically consists of an observation space $\boldsymbol{O}$, an action space $\boldsymbol{A}$, a state transfer probability distribution P, a reward function r and a discount factor $\gamma \in (0,1]$. At each time interval t, the agent observes the current observation $o(t) \in \boldsymbol{O}$ from the controlled system and the action $a(t) \in \boldsymbol{A}$ is generated and executed according to policy π. The next observation $o(t+1) \in \boldsymbol{O}$ is then observed from the controlled system, whereas the agent receives a reward r for that step. To evaluate the cumulative reward $Vt = \sum_{t=0}^{\infty} \gamma^t r(t)$ of the action a_t output by the policy π in the observation o_t, the action-value function or Q-function is defined as $Q^{\pi}(o_t, a_t) = \mathbb{E}[V_t | o_t, a_t]$, which can be described by the Bellman equation, i.e.

$$Q^{\pi}(o_t, a_t) = r + \gamma Q^{\pi}(o_{t+1}, a_{t+1}) \tag{6}$$

The policy $\pi(a_t|o_t)$ is a distribution over actions from several recent observations, usually starting with a randomized exploration (Wu, 2022b). The policy that keeps the controlled system in the desired state is defined as the optimal policy π^*. In this case, the output error is minimized, whereas the cumulative expected reward is maximized.

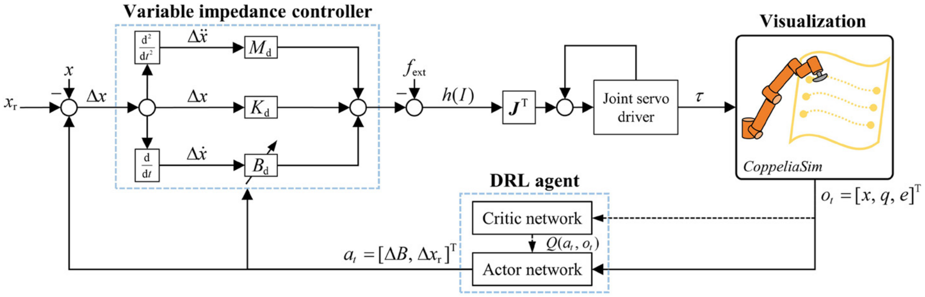
$$\pi * (o_t) = \arg\max_{a_t \in \mathbf{A}} Q * (o_t, a_t) \tag{7}$$

where Q^* is the optimal Q-function and satisfies $Q^* \geq Q^{\pi}$ for all π . We designed a DRL-based variable impedance control framework to obtain the optimal impedance tuning policy for the force tracking task, as shown in Figure 2.

2.2.1 Observations

The learning of impedance tuning policy requires a large amount of observation data generated by the robot interacting with the environment. If a real robot is applied, random exploration at the beginning of policy learning can easily lead to manipulator damage. Therefore, we perform the training in a visualization simulator called *CoppeliaSim*, which is a very well-developed simulator in the field of robotics. With *CoppeliaSim*'s virtual physics engine, manipulation can be simulated in either velocity control mode or position servo mode, allowing DRL agents to train and optimize impedance tuning policy. The observation space $\boldsymbol{O}$ includes a set of manipulator state information (including joint angles q and end-effector position x) and a set of

Figure 2 Scheme of the DRL-based variable impedance control



Source: Authors own work

force error information $e = f_{ext} - f_d$ obtained by force sensors. The observation $o \in \boldsymbol{O}$ is defined as $o = [q, x, e]^T$.

In order for the agent to transfer directly from the simulation to the real platform, the observations for training should be as similar as possible to the real robot (Ferro, 2023). Therefore, we added artificial Gaussian noise to the observations in *CoppeliaSim* to simulate sensor noise, with the sensor noise variance set to 30% of the performance tolerance. Physical properties such as stiffness, damping and collision were added to the contact process to make the simulation more realistic.

2.2.2 *Actions*

Without loss of generality, an example of one dimension in the task space is illustrated. The inertia term $M_d(t)$ in the impedance relation is set to zero to avoid unwanted oscillations. At each phase of force tracking, the impedance relation $I(t)$ (including reference trajectory x_r, stiffness term $K(t)$ and damping term $B(t)$) is tuned to the geometrical variations of the environment. Based on the reference trajectory, the desired trajectory command $h(I)$ is solved by the impedance relation. To ensure that the actions generated by the DRL agents can be thoroughly executed, the selected robot model is known with sufficient accuracy. Thus, the input joint torques τ in equation (1) for driving the contact motion are generated by the following principle (Siciliano and Villani, 2000):

$$\tau = \mathcal{J}^T h(I) + g(q) \tag{8}$$

Uncertain changes in the environment can lead to force errors. The designed DRL agent is intended to tune the impedance parameter, i.e. to output an action:

$$a = [\Delta B, \Delta x_r]^T \tag{9}$$

The impedance relation is updated to $I(t + 1) = I(t) + a(t)$. The DRL-based variable impedance controller then generates the new joint torque via equation (7). The above steps are repeated until the force error converges to zero.

2.2.3 *Reward*

To apply DRL-based method to the specific robotics task, it is useful to finely design the reward function. Considering the problem that rewards tend to fall into local optimality when DRL is applied to multi-control processes in unknown environments, a combined reward function is used to balance tracking results with reduced energy consumption. The reward function is defined as:

$$r = r_{goal} + r_{energy} \tag{10}$$

where r_{goal} is the goal reward function related to the tracking error and r_{energy} is the energy function related to the action.

The core metric of the force tracking task lies in the force error. The smaller the force error is, the better the control is. We establish a Gaussian-type reward function r_{goal} based on the error e between the external force f_{ext} and the desired force f_d. f_{ext} is obtained from the force sensor. The r_{goal} is expressed as:

$$r_{goal} = r_{done} \exp\left(-\frac{e^2}{f_d^2}\right) \tag{11}$$

where r_{done} is a positive real number. Thus, when the goal is reached, i.e. force tracking error $e = 0$, the task considers *success* corresponding to the reward r_{done}.

We expect the robot to perform the tracking task with minimal energy consumption. So the agent is encouraged to take small actions to minimize the extra energy loss from unnecessary actions of the manipulator. The energy reward r_{energy} takes the following quadratic form:

$$r_{energy} = -a^T R_{action} a \tag{12}$$

where R_{action} is a positive definite diagonal array.

2.3 Policy learning

To address the optimal impedance parameter tuning policy in equation (7), we use DRL algorithm for learning. The DRL algorithm uses the actor-critic structure, which is the comprehensive algorithm that combines policy and value iterations. We trained the critic network and the actor network as follows:

2.3.1 *Critic network and updating*

The critic network is used to estimate the Q function of the action a_t in the observation o_t and has an H_c-layer depth structure:

$$\widehat{Q}_\psi = f_{H_c}\left(\ldots f_1\left(e_{\sum}, \sigma\left(W^c_{1,a} a_t + W^c_{1,o} o_t\right)\right)\right) \tag{13}$$

where $\psi = \{W^c_{H_c}, \ldots, W^c_1\}$is the critic network weights, $f_h(s^2)$ with $(1 \le h \le H_c)$ is the hth layer of the recurrent neural network,

e_Σ is the weighted sum of the system output errors under different phases and $\sigma(s^2)$ is the activation function.

Calculate the regression loss function of the critic network:

$$L_c(\psi) = \mathbb{E}\left[\left(\gamma\widehat{Q}_t - \left(\widehat{Q}_{t-1} - r_{t-1}\right)\right)^2\right] \tag{14}$$

to update the weight parameter $\psi' \leftarrow \psi - \eta_c \nabla_\psi L_c(\psi)$ by the gradient error, η_c is the learning rate of critic network.

2.3.2 *Actor network and updating*

The actor network outputs the action a_t for impedance tuning through the observation o_t and has an H_a-layer depth structure:

$$a_t = \pi_\theta(o_t) = f_{H_a}\left(\dots f_1\left(\sigma\left(W_1^a o_t\right)\right)\right) \tag{15}$$

where $\theta = \{W_{H_a}^a, \dots, W_1^a\}$ is the actor network weights.

The actor network is updated using the error between the critic network's estimate of the current action-value function $\widehat{Q}_t$ and the desired goal reward. As we define "r_{done}" as the final goal reward and the non-positive nature of r_{energy}, the optimization objective of the actor network can be calculated directly from the policy goals, denoted as:

$$L_a(\theta) = \mathbb{E}\left[\log\pi_\theta(o_t)\left(r_{\text{done}} - \widehat{Q}_t\right)\right] \tag{16}$$

The actor network parameter is updated by calculating the policy gradient, $\theta' \leftarrow \theta - \eta_a \nabla_\theta L_a(\theta)$, where η_a is the learning rate of actor network.

Notably, this paper focuses on how to provide stability guarantees for the training results from a control system perspective, rather than on the tuning of hyperparameters, which is more of an engineering task. Therefore, all DRL algorithms use shared hyperparameter settings and a close number of network layers to reduce the number of variables. The specific control code will be open-sourced after the paper is published.

3. Grinding stability analysis

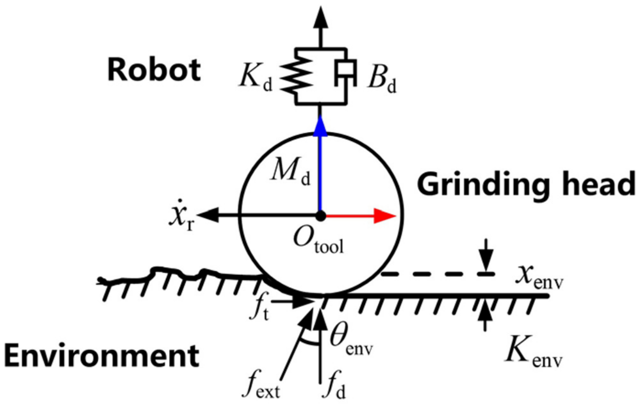
Considering the possible instability of the policy during stochastic exploration, this section analyses the Lyapunov stability of the proposed control policy in unknown environments. Based on the stability analysis, it aims to set boundaries for the exploration of the policy to ensure the safe transfer of learning results from simulation to real robot.

3.1 Grinding process

We modelled the grinding process as shown in Figure 3. The goal of DRL-based variable impedance control for force-tracking tasks is to develop impedance tuning policies such that the system response satisfies the specified characteristics. From equation (5), the reference trajectory error is transformed into the frequency domain:

$$\Delta X_{\text{r}}(s) = \left[Ms^2 + Bs + K\right]^{-1}\Delta F(s) \tag{17}$$

Figure 3 Grinding process model



Source: Authors own work

where s denotes the complex variable in the frequency domain and the corresponding variables [position error $\Delta X(s)$, force error $\Delta F(s)$, etc.] are capitalized in the frequency domain.

When in non-contact free motion (Phase 1), the end-effector of the manipulator follows the reference trajectory exactly. Without considering the underlying servo error, the reference trajectory error remains zero, i.e. $\Delta x_r = 0$.

When in contact motion (Phases 2 and 3), there will be non-linear error between the end-effector trajectory and the reference trajectory. Due to the uncertainty of the environmental information, the reference trajectory needs to be corrected according to the impedance model to compensate for the force error. From equation (4), the force error is written as:

$$\begin{aligned}\Delta F(s) &= F_{\text{ext}} - F_{\text{d}} = K_{\text{env}}(X - X_{\text{env}}) - F_{\text{d}} \\ &= K_{\text{env}}(X + \Delta X - X_{\text{env}}) - F_{\text{d}}\end{aligned} \tag{18}$$

Combined with equation (17), the steady-state force-tracking error can be calculated by:

$$\Delta f_{\text{ss}} = \lim_{s \to 0} \Delta F = K[K_{\text{env}}(X_{\text{r}} - X_{\text{env}}) - F_{\text{d}}](K - K_{\text{env}}) \tag{19}$$

Considering that x_{env} and K_{env} in the environment are not known beforehand, i.e. $K_{\text{env}}\ (X_{\text{r}} - X_{\text{env}}) - F_{\text{d}} \neq \boldsymbol{0}$. Therefore, the reference trajectory X_{r} will be updated by the RL agent to satisfy the steady-state zero error. Up to this point, the proposed DRL-based impedance control allows the development of a unified learning controller for force tracking – without the complex switching of learning models.

3.2 Action boundary analysis for stability

Some useful properties of the robotic dynamic model for analysing stability are presented below (Siciliano and Villani, 2000):

Property 1: The manipulator inertia $\overline{M}$ in equation (1) is symmetric positive definite matrix, it satisfies:

$$0 < \xi^{\text{T}}\overline{M}\xi < \infty, \forall \xi \in \mathbb{R}^n \ \& \ \xi \neq 0 \tag{20}$$

Property 2: From the passive nature of the dynamic model of the manipulator, we have:

$$\dot{\xi}^{\mathrm{T}}\left(\dot{\overline{M}}(\xi,\dot{\xi})-2\overline{C}(\xi,\dot{\xi})\right)\dot{\xi}^{\mathrm{T}}=0 \tag{21}$$

Now, we can give sufficient conditions for the stability of the DRL-based variable impedance controller to set bounds for the impedance learning results.

Theorem 1: Set the initial damping matrix $B(0)$ to be positive definite, then the DRL-based impedance controller is stable if:

$$\lambda_{\min}\{\Delta B(t)\} \geq 0,\ \text{for}\ t > 0 \tag{22}$$

where $\lambda_{\min}\{s^2\}$ denotes taking the minimum eigenvalue of the matrix, $\Delta B(t)$ is the output of the DRL policy network.

Proof: Considering the Lyapunov function candidate:

$$\mathcal{V}=\frac{1}{2}(\dot{e}_q^{\mathrm{T}}\overline{M}\dot{e}_q+e_x^{\mathrm{T}}K_{\mathrm{env}}e_x) \tag{23}$$

where $\dot{e}_q=\dot{q}-\dot{q}_{\mathrm{r}}$ is the angle velocity error in joint space, $e_x=x-x_{\mathrm{r}}$ is the position error in task space. Thanks to the positive characterization of $\overline{M}$ and K_{env}, it is easy to obtain.
$\mathcal{V}\geq\lambda_{\min}\{\overline{M}\}\|\dot{e}_q\|^2/2+\lambda_{\min}\{K_{\mathrm{env}}\}\|e_x\|^2/2>0$

Based on the analysis of the premise, choose and substitute it into equation (1). Associating equation (5), simplifying yields:

$$\overline{M}\ddot{e}_q=-\overline{C}\dot{e}_q-\mathcal{J}^{\mathrm{T}}(f_{\mathrm{ext}}+B(t)\dot{e}_x) \tag{24}$$

Calculating the time derivative of the candidate function yields:

$$\begin{aligned}\dot{\mathcal{V}}&=\frac{1}{2}\dot{e}_q^{\mathrm{T}}\dot{\overline{M}}\dot{e}_q+\dot{e}_q^{\mathrm{T}}\overline{M}\ddot{e}_q+\dot{e}_x^{\mathrm{T}}K_{\mathrm{env}}e_x\\&=\dot{e}_q^{\mathrm{T}}(\frac{1}{2}\dot{\overline{M}}-C)\dot{e}_q+\dot{e}_x^{\mathrm{T}}\left(K_{\mathrm{env}}e_x-f_{\mathrm{ext}}-B(t)\dot{e}_x\right)\\&=-\dot{e}_x^{\mathrm{T}}B(t)\dot{e}_x\end{aligned} \tag{25}$$

Due to the positive definiteness of initial damping matrix, we have $|B(t+1)|=|B(t)+\Delta B(t)|\geq\max\{|B(t)|,|\Delta B(t)|\}>0$. Using mathematical induction one can conclude that keeping $\Delta B(t)$ as semi-positive definite yields positive definite $B(t)$. This implies a negative definite value , i.e. the system will be asymptotically stable via the Lyapunov criterion. This sets the boundaries of higher sample utilization for the exploration of policy networks while ensuring the stability and interpretability of DRL-based controllers.

4. Simulations and experiments

n this paper, the UR5 manipulator is used to illustrate the implementation and the kinematic and dynamic parameters of UR5 simulation model is referenced from the real robot. The simulation and actual robot settings are kept the same. We evaluated our DRL-based impedance control agent on both simulation and real robot hardware. In simulations, we trained our agent with a combination of different DRL algorithmic models to select the one that performs best in terms of convergence time and force tracking during testing. For the real robot experiments, the best-performing agent in the simulation is applied to execute the force tracking task and compared with the constant impedance control method and other variable impedance control methods.

Algorithm 1: DRL algorithms for impedance control

```
Require: Environmental observations $\boldsymbol{O}$.
Ensure: Impedance tuning policy $\pi$.
Initialize network parameters and robot state.
for $episode$=1 to $Episode_{\max}$ do
    for $t$=1 to $T_{\max}$ do
        Obtain the observation $o_t$
        $a_t \leftarrow \pi(o_t)$ by (22)
        $o_{t+1}, r \leftarrow$ CoppeliaSim
        $D_{\mathrm{buffer}} \leftarrow D_{\mathrm{buffer}} \cup \{(o_t, a_t, r, o_{t+1})\}$
        if size of $(D_{buffer}) > D_{\max}$ then
            Update weight of policy networks
        end if
    end for
end for
```

Source: Authors own work.

4.1 Simulation process

In *CoppeliaSim*, the simulation model consists of a UR5 manipulator with a fixed base and different workpieces generated with random polynomial parameters. The training data is generated for the agent through the proprioception of the manipulator and the interaction forces of the end-effector with the different geometrical workpieces. We trained on three different DRL algorithms (DDPG, PPO, SAC) to validate the effectiveness of the proposed method. All DRL algorithms were trained on the same laptop equipped with an 8-core Intel i7 (16 GB RAM) and an NVIDIA RTX3060 GPU. Only DRL-based variable impedance controllers that work best in simulation and satisfy stability can be deployed in real robots. Table 1 illustrates the hyperparameters and training sample parameters shared by the agents during training. The training process of the policy network consists of the following steps: Initialize the relevant hyperparameters and the replay buffer D_{buffer}. Initialize the manipulator joints and generate the contact environment based on random polynomial interpolation in the finite task space. Obtain the observation o_t, select an action a_t and send it to the simulation robot. To guarantee the deployment effectiveness of sim2real, the action exploration boundary will be set based on the Lyapunov's stability analysis. Then obtain the reward r and the next observation o_{t+1}. Calculate the gradient error of different neural networks. Select three DRL algorithms to find the optimal action policy respectively. Calculate the impedance parameters for variable impedance control and then apply it to *CoppeliaSim* for force tracking

Table 1 Hyperparameters shared by different DRL agents

Hyperparameters	Value
Maximum training episodes *Episode*$_{\max}$	900
Learning epoques $T_{\max}$	350
Buffer size $D_{\max}$	256
Reward discount factor γ	0.99
Learning rate η_c, η_a	0.001, 0.001
Action dimension	3
Observation dimension	12

Source: Authors' own work

control and collect new data. Repeat Steps 1–4 until the desired force tracking error is less than the tolerance error. Algorithm 1 summarizes the entire process.

4.2 Simulation results

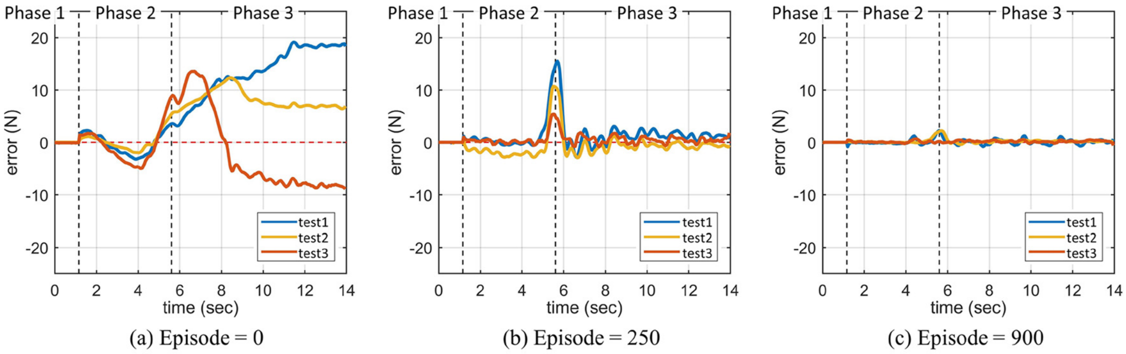
To validate the effectiveness of DRL agents for impedance tuning policy learning, three metrics are employed to assess the efficacy of these methodologies: force error, average reward and convergence speed. To assess the efficacy of the agent in reducing the tracking error, we selected controllers in disparate episodes and the outcomes are illustrated in Figure 4. The same scale of workpiece samples was used to remove the effects caused by workpiece differences.

When the *episode* = 0, the controllers can be equated to the impedance controller without DRL agent. In this case, the force tracking error is large, especially during Phase 3 of dynamic changes in the environment, as shown in Figure 4(a). As the episode increases, the force tracking error decreases, as shown in Figure 4(b) and (c). When the *episode* = 900, i.e. Figure 4(c), the force tracking error is already acceptable. This shows that as the number of iterations increases, the force tracking errors of the proposed DRL-based controllers with action boundary settings all converge to the steady state values.

To further validate the necessity of the action boundary setting and the reasonableness of the Theorem 1, we give the average reward variations during the iteration process of variable impedance controllers based on different DRL algorithms. The larger average reward implies better force tracking. As shown in Figure 5, the stochastic exploration policy without setting action boundaries (blue line) converges quickly in the three DRL algorithms, but fluctuates heavily after the initial convergence. This means that the agent still oscillates around the desired impedance parameter when approaching stabilization. What is worse, the unbounded stochastic exploration shows overfitting in Figure 5(a)–(c), which directly affects the safe transfer of learning results. On the other hand, the group (red line) with action bounds introduced based on Theorem 1 can be more stable after convergence, and the average rewards obtained by all three DRL-based control policies grow steadily.

Considering the effect of the action boundary setting on the speed of convergence of the algorithms, we tested the average force error of the controllers with and without the boundary condition in different episodes. We also tested the average force error of other DRL controller RLOC in different episodes, as shown in Figure 6. The same workpiece samples were used for

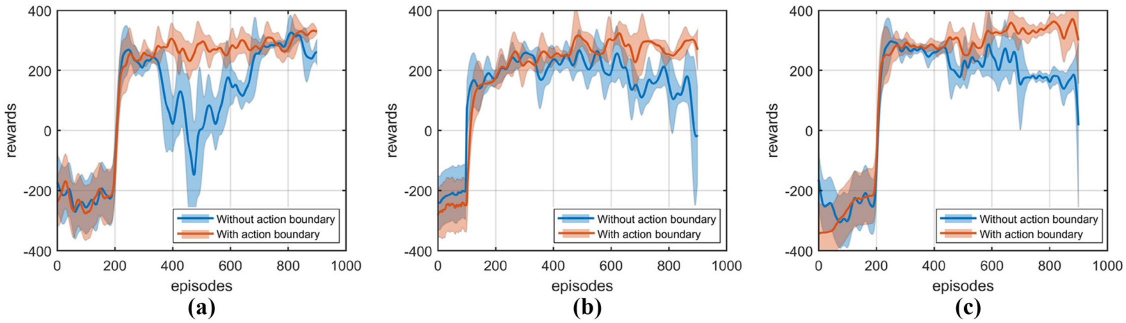
Figure 4 Force tracking error in the task space under different episodes



(a) Episode = 0
(b) Episode = 250
(c) Episode = 900

Notes: (a) Episode = 0; (b) episode = 250; (c) episode = 900
Source: Authors own work

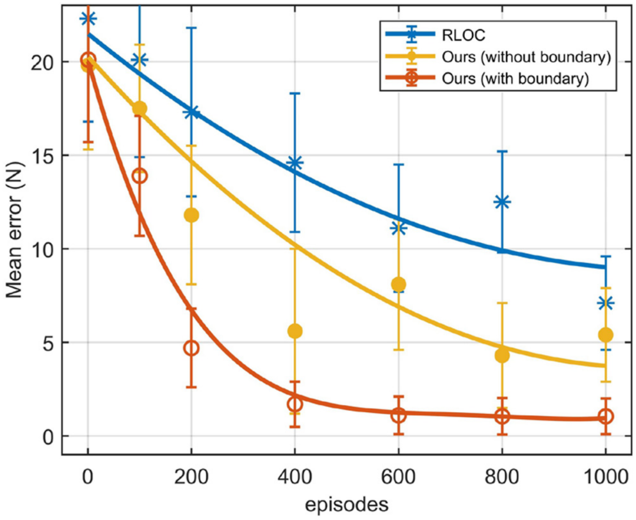
Figure 5 Schematic representation of the reward convergence process with and without action boundary setting



(a)
(b)
(c)

Notes: (a) DDPG; (b) PPO; (c) SAC
Source: Authors own work

Figure 6 Mean tracking error in different training episodes



Source: Authors own work

all agents. The results show that our method converges faster compared to the other methods. Because the positive definite restriction on damping, which acquires more stable samples and thus has higher sample utilization and convergence speed.

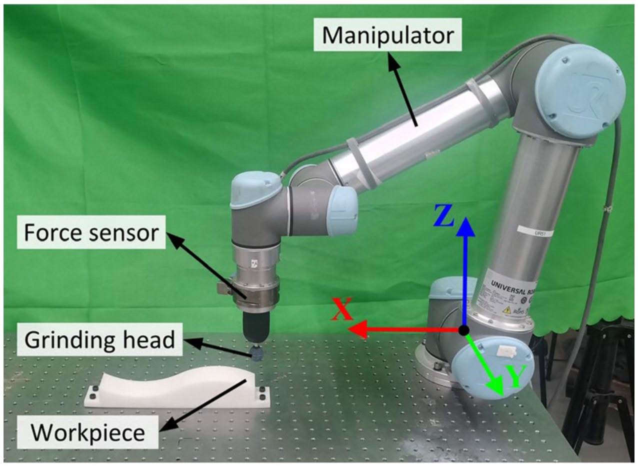
4.3 Experiments validation

4.3.1 Experimental setup

To verify the feasibility of the proposed method on the real robot, we conducted force-tracking experiments in different workpieces with complex shapes. The experimental setup is shown in Figure 7. A six-axis force sensor is installed at the end of the manipulator and the grinding head is connected to the force sensor. The grinding head is a 10 mm diameter sphere and the workpiece is fixed on an optical platform.

In the experiments, the trained DRL agent controls the real UR5 robot to interact with the environment. The control cycle was set to 0.05 s. The end of the robot was equipped with an ATI-Gamma force sensor (rated load at 400 N with an accuracy of 0.05 N) and pre-calibrated for compensation. Consider the robot approaching a workpiece surface from the negative z-direction and then moving 30 cm in the x-direction, maintaining constant force contact during the movement. The desired force was set to be 20 N, oriented perpendicular to the optical platform surface (world coordinate z-direction). We tested the force-tracking effect of the DRL-based variable impedance control on different workpieces. We compared our method with constant impedance control and adaptive variable impedance control. The implementation details of the control methods are as follows:

Figure 7 Experiment scene diagram



Source: Authors own work

- *Constant impedance control* (blue line): the stiffness and the damping parameters are chosen as constant diagonal matrices, i.e., $K = \mathrm{diag}\{k_1, \ldots, k_3\}$ and $B = \mathrm{diag}\{b_1, \ldots, b_3\}$.
- *Adaptive control* (cyan line): $K(t)$ and $B(t)$ are chosen according to the impedance adaptive adjustment principle proposed by Duan *et al.* (2018).
- *DRL-based control* (red line): $K(t)$ and $B(t)$ are chosen by our trained DRL agents. Note that only DRL agents that have converged in simulation could deploy to the real robot.

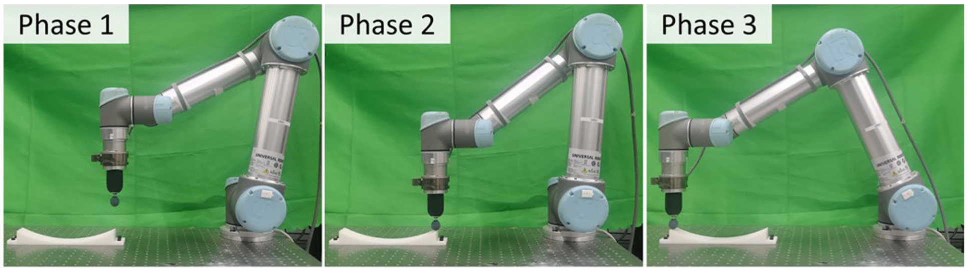
We carried out the experiments on three types of workpieces with different surfaces, including convex surfaces, concave surfaces and sine surfaces, using the above three methods. For the safety of the experiments, we select the agent trained by the SAC algorithm for the real experiments, which performs best in the simulation.

4.3.2 Experiments on convex surfaces

When the surface of the workpiece in contact is convex, the variation of the environmental position then satisfies $\dot{x}_{\mathrm{env}} \neq 0$, $\ddot{x}_{\mathrm{env}} > 0$. The dimensions of the selected workpiece are $300 \times 50 \times 35$ mm of resin material, where the surfaces to be grinded are the inner surfaces of circular arcs. To test the effect of different convexity of the environment on the force tracking, workpieces are made of circular arcs with different diameters. The material stiffness and surface shape of the workpiece are unknown. The same starting machining position and machining duration are used for all control methods. As an example, Figure 8 shows the force tracking process on a convex type workpiece with a radius r_{env} of 300 mm in cross-section.

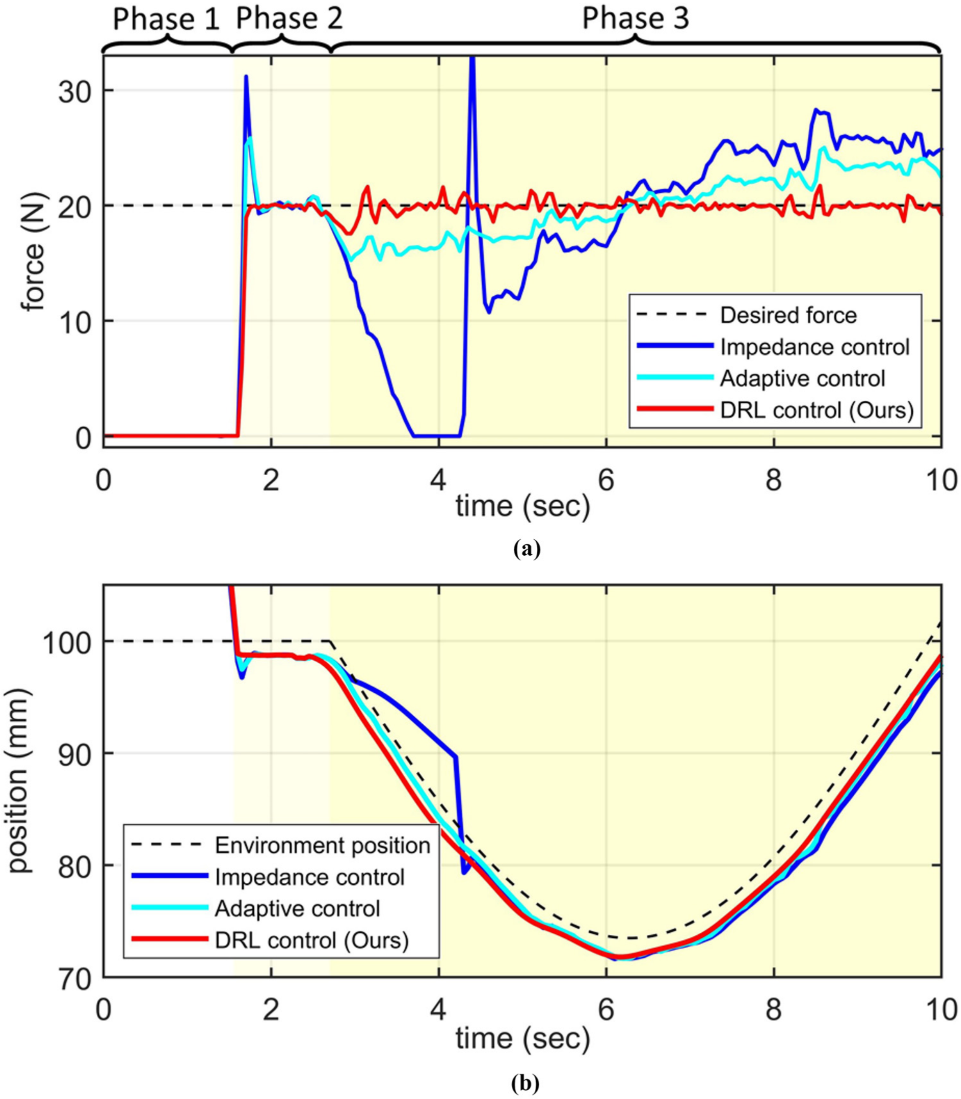
Figure 9 shows the results of force tracking on convex surfaces with different control methods. In the first phase (0–1.7 s), the manipulator moves in free space, during which the desired force guides the movement of the robot mainly. All three control methods were able to guide the end-effector to move quickly from the same initial position to the contact state with the workpiece surface. In the second phase (1.7–2.7 s), the robot starts to contact the workpiece and causes positional oscillations, which in turn lead to oscillations in the measured force. After the initial overshoot, all controllers were able to quickly track the desired force of 20 N, but the variable impedance controller and the constant impedance controller had larger overshoot errors. The reason that the DRL controller showed less overshoot is that the cost of training comes mainly from the force error. Thus, the proposed DRL controller shows stronger suppression of overshoot. In the third stage (2.7–10 s), the grinding head began to move in the x-

Figure 8 Experimental process on convex surface



Source: Authors own work

Figure 9 Results on convex surfaces



Note: (a) Force tracking results with different control methods; (b) Position results with different control methods
Source: Authors own work

direction to grind the workpiece. Faced with a sudden drop in the environmental rest position, all three methods showed force reductions of different magnitudes. Due to the fixed impedance parameter, the constant impedance control was unable to track the change of the environmental position in time, resulting in errors in the tracking force and even discontinuous grinding (3.6–4.2 s). The re-contact produced another overshoot, which is not allowed in the actual machining process as it may lead to damage to the workpiece. The adaptive control was able to adjust the impedance parameters to track the position drop, but it also suffered from a large tracking error, averaging about 2.5 N. The greater the convexity of the surface, the greater the force error of the adaptive control. Compared to adaptive control, our method can adjust the impedance parameters more quickly to re-achieve force tracking and quickly brings the force error asymptotically to 0.

To further analyse the effect of convexity variations on force tracking, we recorded the third phase force errors for grinding convex workpieces with three different cross-sectional arc radii (200, 300 and 400 mm). The same machining time taken for all tests and the statistical results shown in Table 2. Compared with constant impedance control and adaptive control, our method reduces the force error by 97.3% and 91.1%, respectively.

From the above experimental results, we can conclude that our method minimizes the force tracking error on convex workpieces compared to impedance control and adaptive control.

4.3.3 Experiments on concave surfaces

When the surface of the workpiece in contact is concave, the change in the position of the environment then satisfies $\dot{x}_{\text{env}} \neq 0$, $\ddot{x}_{\text{env}} < 0$. The dimensions of the selected workpiece are $300 \times 50 \times 65$ mm assemblage of rectangular and cylindrical side truncations. To test the effect of different concavity of the environment on the force tracking, cylindrical columns of different diameters were used for the surface to be machined of the workpiece. The workpiece material and cylindrical diameter are not given in advance. All control methods start contacting the workpiece from the same position. Figure 10 shows the force tracking process on concave type workpiece.

Table 2 Tracking results under different convex surfaces

Methods	Force error (N) $r_{\text{env}} = 200$ mm	300 mm	400 mm
Constant impedance	12.22 ± 9.62	8.85 ± 6.42	5.21 ± 3.62
Adaptive control	3.42 ± 2.71	2.67 ± 1.62	1.78 ± 1.06
Ours	*0.30 ± 0.23*	*0.21 ± 0.17*	*0.19 ± 0.16*

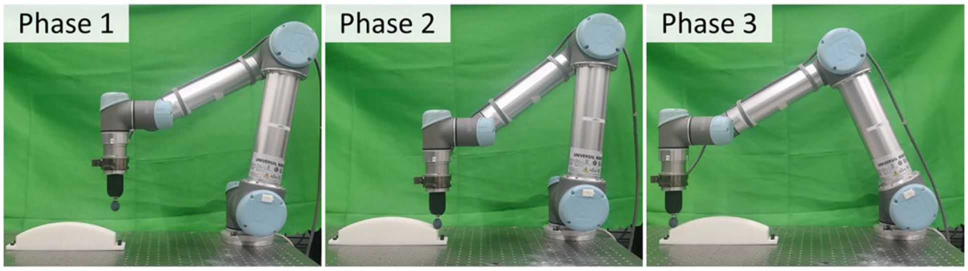
Source: Duan *et al.* (2018) proposed the adaptive control. Others are authors' own work

Figure 11 shows the results of force tracking on the concave surface with different control methods. The results in the first and second phases are similar to the previous ones. In the third phase (2.7–10 s), the rest position of the environment suddenly rises, and both the constant impedance control and the adaptive control experience a rise in the measured force of different magnitudes. Faced with the concave change in the environment, the constant impedance control was unable to re-achieve force tracking and there was always a constant tracking error (about 6.5 N on average). Both the adaptive control and our method were able to re-track the force and were able to eliminate the tracking error by adjusting the impedance parameters. However, the adjustment of the impedance parameter in the adaptive control always lags behind the change of the force error, resulting in a larger force error at the suddenly slope change. Compared with adaptive control, our method can adjust the impedance parameters more quickly to adapt to the concave changes through extensive simulation training, which effectively reduces the force error.

To further analyse the effect of different concavity variations on the force error, we statistics the third phase force tracking results under different concave surface, as shown in Table 3. The concave surface is made up of cylinders with different cross-sectional radii, including 200, 300 and 400 mm. The results show that compared to constant impedance control and adaptive control, the proposed DRL control can reduce the tracking error by 90.4%– 97.0% at different concave surfaces.

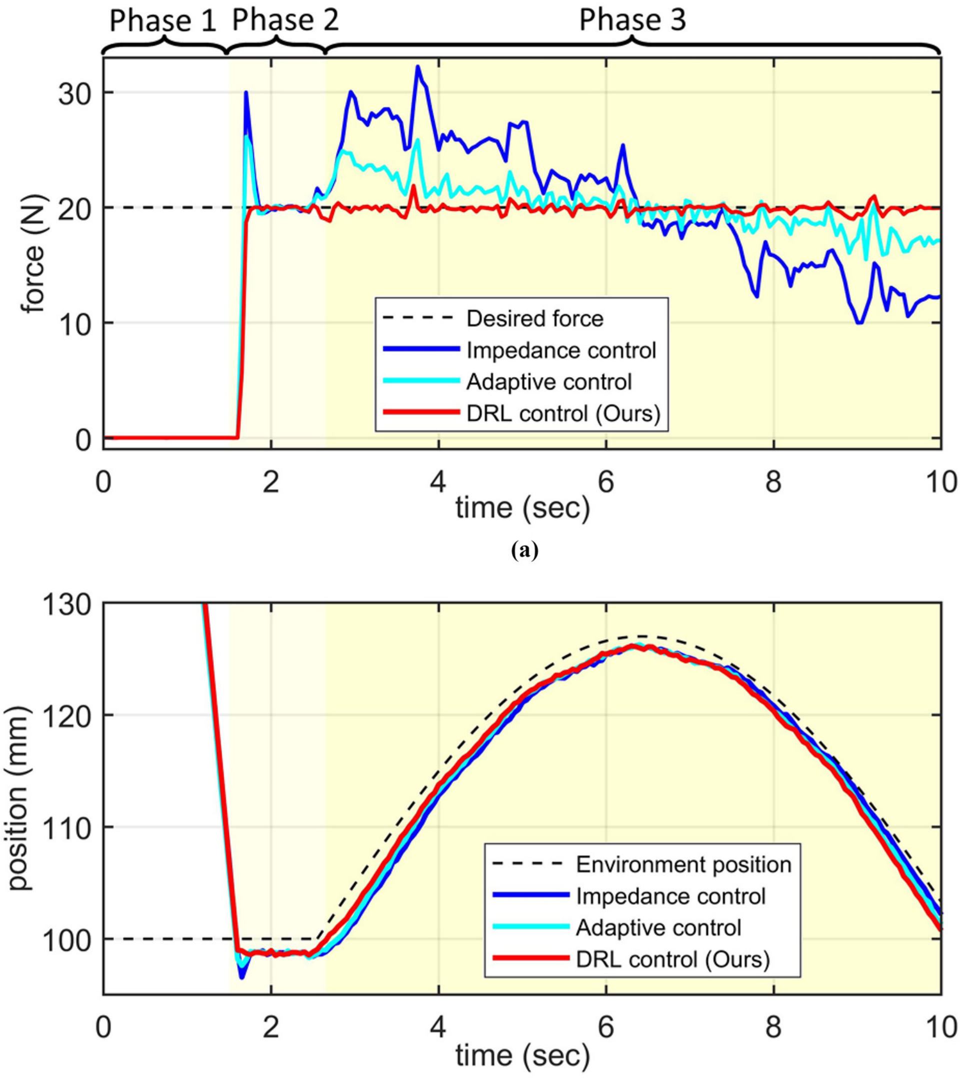
From the above experimental results, we can conclude that the constant impedance controller has a constant tracking error on the concave surface. The adaptive control can although the steady-state error is zero (force error decreases continuously), but the adjustment of impedance parameters always lags behind the change of force error, which leads to poor tracking effect when the slope is large. Thanks to the *a priori* training in the simulation environment, our method can compensate for the concave changes in the environment in time to quickly track the desired force.

Figure 10 Experimental process on concave surfaces



Source: Authors own work

Figure 11 Results on concave surfaces



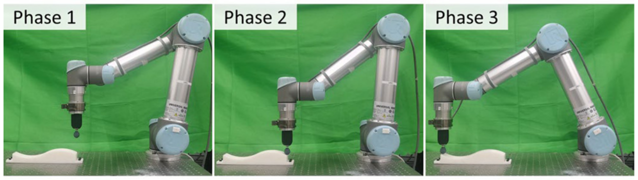
Note: (a) Force tracking results with different control methods; (b) position results with different control methods
Source: Authors own work

Table 3 Tracking results under different concave surfaces

	Force error (N)		
Methods	$r_{\text{env}} = 200$ mm	300 mm	400 mm
Constant impedance	9.55 ± 5.43	6.72 ± 3.99	3.41 ± 2.72
Adaptive control	3.14 ± 2.20	1.95 ± 1.42	1.17 ± 0.93
Ours	*0.23 ± 0.24*	*0.19 ± 0.18*	*0.18 ± 0.16*

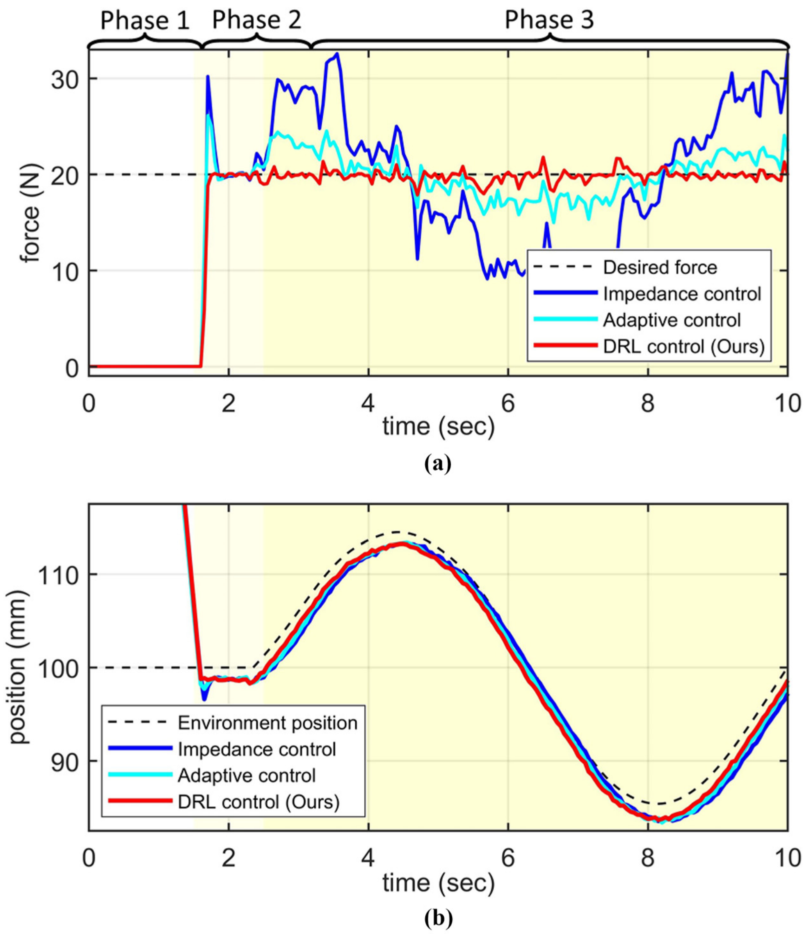
Source: Duan *et al.* (2018) proposed the adaptive control. Others are authors' own work

Figure 12 Experimental process on sine surface



Source: Authors' own work

Figure 13 Results on sine surfaceNote: (a) Force tracking results with different control methods; (b) position results with different control methods



(a)

(b)

Notes: (a) Force tracking results with different control methods;
(b) position results with different control methods
Source: Authors own work

4.3.4 Experiments on sine surfaces

When the surface of the workpiece in contact is sine, the variation of the environmental position then satisfies $\dot{x}_{\text{env}} \neq 0$, $\ddot{x}_{\text{env}} \neq 0$. The dimensions of the selected sine workpiece are $300 \times 50 \times 50$ mm, and it can be considered as a combination of convex and concave surfaces. Based on this, we study the effect of switching between convex and concave surfaces on force tracking. Any continuous surface can be decomposed into a combination of finite number of concave and convex surfaces, which lays the foundation for the next step of grinding complex surfaces. All control methods start contacting the workpiece from the same position and use the same machining time. Figure 12 shows the force tracking process on the sine type workpiece.

Figure 13 shows the results of force tracking on sine surface for different control methods. The third phase describes the effect of non-linear changes in the environment position on the force tracking results, especially at the position of the concave–convex surface switch. In the third phase (2.7–10 s), the machined surface always varies nonlinearly. Facing the nonlinear change of the environment, neither constant impedance control nor adaptive control can re-track the force. In contrast, our method can transition the concave-convex surface intersection stably (around 6.3 s). Due to the adaptation of the neural network to non-linear terms, our method exhibits strong robustness to nonlinear changes in the environment and can quickly re-track force.

To further validate the effectiveness of the proposed method in the nonlinear variations of the environment, we statistics the third-phase force tracking error ratios ϵ % for different desired forces ($f_{\text{d}} = 5$ N, 10 N, 20 N and 40 N), where ϵ % is defined as the ratio of the average force error to the desired force. The statistical results of the force tracking error ratios for different desired forces are given in Table 4. The results show that our method can adapt to the non-linear variations of the environment and the average force tracking error is less than 5% in the range from 5 to 40 N. Compared with constant impedance control and adaptive control, our method can reduce the tracking error by 89.0% and 78.7%, respectively.

From the above experimental results, we can conclude that our method can fully use the nonlinear fitting advantage of the deep network to adapt to the nonlinear changes of the environment.

Table 4 Force tracking results with different desired forces

	Force error ratio ϵ (%)			
Methods	$f_d = 5$ N	10 N	20 N	40 N
Constant impedance	39.5	33.7	35.4	47.1
Adaptive control	19.0	15.3	18.5	27.4
Ours	*4.28*	*3.79*	*4.11*	*4.87*

Source: Authors' own work

Especially for the strong nonlinearity at the concave–convex surface switching, our method shows stronger robustness than the constant impedance control and adaptive control.

4.3.5 Comparisons on complex surfaces

To verify the state-of-the-art of the proposed method, in addition to the classical control methods, we also compared it with other DRL-based control method (e.g. RLOC). New workpieces with different materials (resin, glass, aluminium) and different surfaces were selected as test objects. We performed 50 tests on new workpieces and the results are shown in Table 5.

Compared to the constant impedance method and variable impedance methods, our method reduces the average force tracking error by 89.1% and 79.4% on the new workpiece. Although the settling time increases by 0.045 s relative to the constant impedance method, for the overshoot, our method decreases by 24.3%–31.4% compared to the constant impedance and variable impedance methods, which more acceptable in the grind task. Compared to RLOC, our action boundaries set by stability analysis make the control more robust, reducing the standard deviation of force tracking by 76.0%.

5. Conclusion

In this paper, a DRL variable impedance control method supporting stability analysis is proposed for the force-tracking task. The method provides fast convergence and stable control results in uncertain environments. The main contribution of this paper is twofold. Firstly, a safe boundary is set for the exploration action by analysing the Lyapunov stability of the DRL controller, which allows the agent to obtain more positive samples and avoids ineffective exploration. Secondly, based on the theoretical analysis results and the combination reward function, a DRL-based variable impedance controller is designed, which can significantly improve the force tracking effect and ensure the safe transfer of learning results to the real robot. A simulation interaction model of the UR5 simulation robot and random parameter-generated workpiece was constructed in *CoppeliaSim*, and joint simulation training based on three DRL algorithms was performed. Finally, experimental validation was performed on the real robot. Compared with the constant impedance method and adaptive variable impedance control, the proposed DRL-based controllers can remain stable in unknown environments and the force tracking error is reduced by 70%–90%.

In the future, we will continue to study the convergence and result interpretability of the DRL controller to improve the sample utilization and generalization to extend it to more contact tasks.

Table 5 Results of different control method ($f_d = 20$ N)

Methods	Average force error (N)	Settling time (s)	Overshoot (%)
Constant impedance	6.85 ± 4.21	*0.161*	34.1
Adaptive control	3.63 ± 2.11	0.212	27.4
RLOC	0.79 ± 5.12	0.231	5.5
Ours	*0.57 ± 1.23*	0.189	*4.2*

Source: Duan *et al.* (2018) proposed the adaptive control. Gangapurwala *et al.* (2022) proposed RLOC. Others are authors' own work

References

Abu-Dakka, F. and Matteo, S. (2020), “Variable impedance control and learning–a review”, *Frontiers in Robotics and AI*, Vol. 7.

Beltran, C.C. (2020), “Learning force control for contact-rich manipulation tasks with rigid position-controlled robots”, *IEEE Robotics and Automation Letters*, Vol. 5 No. 4, pp. 5709-5716.

Campeau, A. (2019), “Intuitive adaptive orientation control for enhanced human–robot interaction”, *IEEE Transactions on Robotics*, Vol. 35 No. 2, pp. 509-520.

Chien-Chern (1998), “Learning impedance control for robotic manipulators”, *IEEE Transactions on Robotics and Automation*, Vol. 14 No. 3, pp. 452-465.

Deng, Y., Wang, G., Yue, X. and Zhou, K. (2022), ““A review of robot grinding and polishing force control mode”, *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1413-1418.

Duan, J., Gan, Y., Chen, M. and Dai, X. (2018), “Adaptive variable impedance control for dynamic contact force tracking in uncertain environment”, *Robotics and Autonomous Systems*, Vol. 102, pp. 54-65.

Ferro, M. (2023), “A CoppeliaSim dynamic simulator for the Da Vinci research kit”, *IEEE Robotics and Automation Letters*, Vol. 8 No. 1, pp. 129-136.

Gangapurwala, S., Geisert, M., Orsolino, R., Fallon, M. and Havoutis, I. (2022), “RLOC: terrain-aware legged locomotion using reinforcement learning and optimal control”, *IEEE Transactions on Robotics*, Vol. 38 No. 5, pp. 2908-2927.

Hailong, X., Qinghui, W. and Zipeng, C. (2022), “Adaptive human-robot collaboration for robotic grinding of complex workpieces”, *CIRP Annals*, Vol. 71 No. 1, pp. 285-288.

Ko, D., Lee, D., Chung, W.K. and Kim, K. (2022), “On the performance and passivity of admittance control with feed-forward input”, *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 11209-11215.

Kong, D. and Huang, Q. (2023), “Impedance force control of manipulator based on variable universe fuzzy control”, *Actuators*, Vol. 12 No. 8, p. 305.

Li, C. (2018), “Efficient force control learning system for industrial robots based on variable impedance control”, *Sensors*, Vol. 18 No. 8, p. 2539.

Liu, W., Wu, R., Si, J. and Huang, H. (2022), “A new robotic knee impedance control parameter optimization method facilitated by inverse reinforcement learning", *IEEE Robotics and Automation Letters*, Vol. 7 No. 4, pp. 10882-10889.

Okada, M. (2023), "Learning compliant stiffness by impedance control-aware task segmentation and multi-objective Bayesian optimization with priors", *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8155-8162.

Prendergast, J.M., Balvert, S. and Driessen, T. (2021), "Biomechanics aware collaborative robot system for delivery of safe physical therapy in shoulder rehabilitation", *IEEE Robotics and Automation Letters*, Vol. 6 No. 4, pp. 7177-7184.

Seo, C., Kim, H. and Jin, H. (2023), "Force control of a grinding robotic manipulator with floating base via model prediction optimization control", *IEEE/ASME Transactions on Mechatronics*, Vol. 28 No. 4, pp. 1911-1919.

Siciliano, B. and Villani, L. (2000), *Robot Force Control*, *Springer*, Berlin.

Song, H., Ye, R. and Xie, M. (2024), "Research on complex surface grinding path planning of grinding robot based on NX and Tecnomatix", *The International Journal of Advanced Manufacturing Technology*, Vol. 133 Nos 3/4, pp. 1961-1971.

Vanderborght, B., Albu-Schaeffer, A. and Bicchi, A. (2013), "Variable impedance actuators: a review", *Robotics and Autonomous Systems*, Vol. 61 No. 12.

Wu, R. (2022a), "Robotic knee tracking control to mimic the intact human knee profile based on actor-critic reinforcement learning", *IEEE/CAA Journal of Automatica Sinica*, Vol. 9 No. 1, pp. 19-30.

Wu, R. (2022b), "Reinforcement learning impedance control of a robotic prosthesis to coordinate with human intact knee motion", *IEEE Robotics and Automation Letters*, Vol. 7 No. 3, pp. 7014-7020.

Wu, H., Yang, H. and Li, Y. (2023), "Online estimation of 2D human arm stiffness for peg-in-hole tasks with variable impedance control", *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4436-4442.

Yang, C., Zeng, C. and Liang, P. (2018), "Interface design of a physical human–robot interaction system for human impedance adaptive skill transfer", *IEEE Transactions on Automation Science and Engineering*, Vol. 15 No. 1, pp. 329-340.

Zhang, T., Yuan, C. and Zou, Y. (2023), "Research on the algorithm of constant force grinding controller based on reinforcement learning PPO", *The International Journal of Advanced Manufacturing Technology*, Vol. 126 Nos 7/8, pp. 2975-2988.

Zheng, L. (2023), "An efficiently convergent deep reinforcement learning-based trajectory planning method for manipulators in dynamic environments", *Journal of Intelligent & Robotic Systems*, Vol. 107 No. 4, p. 50.

Corresponding author

Erbao Dong can be contacted at: ebdong@ustc.edu.cn