



OPEN ACCESS

EDITED BY

Luciano Luporini Menegaldo,
Federal University of Rio de Janeiro, Brazil

REVIEWED BY

Fernando Lizarralde,
Federal University of Rio de Janeiro, Brazil
Feihu Zhang,
Northwestern Polytechnical University, China

*CORRESPONDENCE

Wen Wu

✉ wuwen66@163.com

RECEIVED 08 September 2023

ACCEPTED 15 January 2024

PUBLISHED 29 January 2024

CITATION

Xiao M, Zhang X, Zhang T, Chen S, Zou Y and Wu W (2024) A study on robot force control based on the GMM/GMR algorithm fusing different compensation strategies. *Front. Neurobot.* 18:1290853. doi: 10.3389/fnbot.2024.1290853

COPYRIGHT

© 2024 Xiao, Zhang, Zhang, Chen, Zou and Wu. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

A study on robot force control based on the GMM/GMR algorithm fusing different compensation strategies

Meng Xiao¹, Xuefei Zhang¹, Tie Zhang², Shouyan Chen³, Yanbiao Zou² and Wen Wu^{1,4*}

¹Department of Rehabilitation, Zhujiang Hospital, Southern Medical University, Guangzhou, China, ²School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, China, ³School of Mechanical and Engineering, Guangzhou University, Guangzhou, China, ⁴Rehabilitation Medical School, Southern Medical University, Guangzhou, China

To address traditional impedance control methods' difficulty with obtaining stable forces during robot-skin contact, a force control based on the Gaussian mixture model/Gaussian mixture regression (GMM/GMR) algorithm fusing different compensation strategies is proposed. The contact relationship between a robot end effector and human skin is established through an impedance control model. To allow the robot to adapt to flexible skin environments, reinforcement learning algorithms and a strategy based on the skin mechanics model compensate for the impedance control strategy. Two different environment dynamics models for reinforcement learning that can be trained offline are proposed to quickly obtain reinforcement learning strategies. Three different compensation strategies are fused based on the GMM/GMR algorithm, exploiting the online calculation of physical models and offline strategies of reinforcement learning, which can improve the robustness and versatility of the algorithm when adapting to different skin environments. The experimental results show that the contact force obtained by the robot force control based on the GMM/GMR algorithm fusing different compensation strategies is relatively stable. It has better versatility than impedance control, and the force error is within $\sim\pm 0.2$ N.

KEYWORDS

robot force control, impedance control, reinforcement learning, deep Q-network (DQN), Gaussian mixture model/Gaussian mixture regression (GMM/GMR)

1 Introduction

The applications of robot-skin contact are diverse, including uses in robotic medical-aided diagnosis, massage, aesthetic nursing, and other scenarios (Christoforou et al., 2020). In these scenarios, robots can work continuously without rest, and simultaneously, they can maintain highly consistent movements, strength, and speed, so they can partially replace human labor (Kerautret et al., 2020). Good robot force control is essential for efficient and comfortable robot-skin contact experiences. Robot force control requirements must address safety, precision, and variability; if the robot applies too little force, it may fail to achieve the intended effect, and if it applies excessive force, it may cause skin pain or injury. The biological characteristics of the skin determine differences in the mechanical characteristics of the skin of different individuals (Zhu et al., 2021); therefore, the robot usually faces unknown contact environments. Ensuring the accuracy of robot interaction considering the characteristics of different people's skin is the focus of current research.

Many researchers and institutions have studied robot force strategies, and impedance control plays an important role in these strategies. Impedance control constructs a contact model between a robot and human skin and flexibly changes dynamic characteristics during interactive tasks (Jutinico et al., 2017). Some scholars, such as Li S. et al. (2017) and Sheng et al. (2021) conducted experimental research on the contact process between the robot and skin based on impedance control. The control parameters of impedance control, such as stiffness and damping, require utilizing manual adjustment or trial and error, and the controller is insensitive to the uncertainty of the external environment. To adapt robots to the flexible environment of human skin, other scholars, such as Liu et al. (2021), Khoramshahi et al. (2020), Li et al. (2020), Ishikura et al. (2023), Huang et al. (2015), and Stephens et al. (2019) used adaptive algorithms and intelligent algorithms for optimizing the impedance control parameters. The skin, being a living tissue, has biomechanical properties, such as elasticity, viscoelasticity, non-linearity, and anisotropy (Joodaki and Panzer, 2018). The mechanical characteristics of the flexible contact environment faced by the robot are often dynamic, and traditional force controllers cannot explore unknown environments.

Reinforcement learning can be used to explore control strategies in robots. Through reinforcement learning, robots can learn how to adjust their control strategies to perform better and adapt to external environmental changes by interacting with that environment (Suomalainen et al., 2022). Many scholars have used reinforcement learning to explore the optimal control strategy; for example, Luo et al. (2021) proposed a method based on Q-learning to optimize online stiffness and damping parameters. Ding et al. (2023) used reinforcement learning to analyze and optimize the impedance parameters. Bogdanovic et al. (2020) used a deep deterministic policy gradient to learn the robot output impedance strategy and the required position in the joint space. Meng et al. (2021) adaptively adjusted the inertia, damping, and stiffness parameters through the proximal policy optimization algorithm. These reinforcement learning algorithms have good versatility and self-adaptability in the interaction process and perform well in the simulation environment, but when used in practical applications, they must often address multiple interactions. Therefore, some scholars have begun using the model-based method to reduce the number of actual interactions and improve the utilization rate of the algorithm (Hou et al., 2020). For example, Zhao et al. (2022) proposed a model-based actor-critic learning algorithm to safely learn strategy and optimize the impedance control. Anand et al. (2022) used a model-based reinforcement learning algorithm, which integrates probabilistic inference for learning force control and motion tracking. Roveda et al. (2020) proposed a variable impedance controller with model-based reinforcement learning, and Li Z. et al. (2017) identified adaptive impedance parameters based on the linear quadratic regulator. In most of the aforementioned studies, the contact environments are rigid, and the established models are relatively stable. These models can predict the dynamic evolution of the environment and the generation of rewards. Furthermore, reinforcement learning agents can identify and make better decisions, so the quality and accuracy of the model directly affect the performance results of reinforcement learning. While

the contact between the robot and human skin is flexible, this environment is more uncertain than the rigid environment, and using reinforcement learning to quickly and efficiently find the optimal strategy in practice has not been achieved (Weng et al., 2020).

Compared to traditional control for robot massage, the main contributions of this work are as follows.

- (1) A robot force controller based on the Gaussian mixture model/Gaussian mixture regression (GMM/GMR) algorithm fusing different compensation strategies is proposed, which combines a traditional robot force controller and reinforcement learning algorithm.
- (2) Two environmental dynamics models of reinforcement learning are constructed to simulate the contact process between the robot and the skin. The number of actual interactions of the reinforcement learning is reduced. At the same time, the practicability of the reinforcement learning algorithm is improved.
- (3) The GMM/GMR algorithm fuses online and offline compensation strategies to improve the robustness and versatility of the algorithm and to adapt to different skin environments.

The remainder of the paper is structured as follows: in the second section, the impedance control strategy is constructed in the contact process of the robot. In the third section, two robot force control compensation strategies based on a deep Q-network (DQN) with dynamic models are proposed, and the strategy of reinforcement learning is learned offline. In the fourth section, an online compensation strategy is built based on a skin mechanics model. In the fifth and sixth sections, the experimental platform is built and experiments are conducted to verify the feasibility of the algorithm. A list of variables used in the paper are shown in Table 1.

2 Robot force control based on impedance control

In robot-skin interaction scenarios, the robot end-effector is equipped with a probe, which makes skin contact and moves along a set trajectory, and the force signal is collected through the sensor between the robot and the probe. To ensure safety during the contact process, the reference force of the contact force must be set and a force controller must be used to adjust the contact state of the robot and ensure that the robot follows the reference force. Impedance control can be used to ensure reasonable contact between robots and human skin; it simplifies the contact model between the robot and the human into a linear second-order system contact model with inertia, damping, and stiffness characteristics. The contact model adjusts the robot displacement based on the difference between the actual measured force and the reference force, while the characteristics of the contact model are adjusted using the inertia, damping, and stiffness parameters (Song et al., 2017). In the Cartesian coordinate system, in the normal direction of the contact between the robot and

TABLE 1 List of variables used in the paper.

| Placement | Variable | Description |
|------------------------------------|----------------------------|---|
| Impedance control | m_d | Inertia parameter of impedance control |
| | b_d | Damping parameter of impedance control |
| | k_d | Stiffness parameter of impedance control |
| | $\Delta\ddot{x}$ | Acceleration of the robot end-effector |
| | $\Delta\dot{x}$ | Velocity of the robot end-effector |
| | Δx | Offset displacement of the robot end-effector |
| | f_r | Reference contact force |
| | f_e | Actual contact force |
| | k | k -th sampling period |
| | T_s | Sampling period |
| | e | Difference between reference force and actual force |
| DQN | s | State |
| | a | Robot action, i.e., robot offset displacement |
| | τ | Trajectory of reinforcement learning |
| | r | Reward |
| | \dot{e}_t | Change of the force error |
| | R | Discounted return |
| | γ | Discount factor |
| | π^* | Optimal strategy |
| | $Q(s, a, \theta^-)$ | Target value deep neural network |
| | $Q(s, a, \theta)$ | Predicted value deep neural network |
| | L | Loss function |
| | y | Value of the target network |
| | G | Experience samples |
| | N | Training iterations |
| | Z | Net activation value |
| | U^l | Activation value |
| | b^l | Bias of the l -th layer |
| | W^l | Weight of the l -th layer |
| | φ | Activation function, the ReLU activation function is selected. |
| | δ^l | Error term for the l -th layer |
| α | Learning rate | |
| λ | Regularization coefficient | |
| BP neural network dynamics model | $NeT1.W$ | Weight in the BP neural network |
| | $NeT1.b$ | Bias in the BP neural network |
| | φ | BP neural network |
| | a^1 | Compensation displacement obtained by DQN with BP neural network dynamics model |
| LSTM neural network dynamics model | ϕ | LSTM neural network |
| | $NeT2.W$ | Weight in the LSTM neural network |
| | $NeT2.b$ | Bias parameters in the LSTM neural network |
| | C_t | Memory state |

(Continued)

TABLE 1 (Continued)

| Placement | Variable | Description |
|-------------------|----------------------|--|
| | o_t | Output gate |
| | i_t | Input gate |
| | f_t | Forget gate |
| | X_t | Input at the current moment |
| | $Net2.W_f$ | Weights of the forget gate |
| | $Net2.W_i$ | Weights of the input gate |
| | $Net2.W_c$ | Weights of estimated state |
| | $Net2.W_o$ | Weights of output gate |
| | \odot | Hadamard product |
| | σ | Logistic function with an output interval |
| | H_t | Hidden state |
| | a^2 | Compensation displacement obtained by DQN with LSTM neural network dynamics model |
| | Skin mechanics model | f_s |
| x | | Coordinate of the robot when it is deformed |
| x_e | | Initial coordinates of the skin |
| k_s | | Elasticity coefficients |
| b_s | | Damping coefficients |
| u^s | | Compensation displacement based on displacement compensation with skin mechanics model |
| GMM/GMR algorithm | t | Time information |
| | n | Number of samples |
| | u | Represents the three kinds of compensation displacements |
| | $P(t, u)$ | Joint probability distribution |
| | M | Number of Gaussian components in the GMM |
| | π_m | Prior probability of the m -th Gaussian component |
| | μ_m | Mean of the m -th Gaussian component |
| | Σ_m | Covariance of the m -th Gaussian component |
| | t^* | The predicted time |
| | u^* | Predicted compensation displacement |
| | u_f | Central distribution of f , final fusion strategy |

the skin, analysis is performed from only one dimension, and the position and contact force of the robot meet the following conditions (Li et al., 2018):

$$m_d \Delta \ddot{x} + b_d \Delta \dot{x} + k_d \Delta x = f_r - f_e \tag{1}$$

where m_d , b_d , and k_d are the inertia, damping, and stiffness parameters of impedance control, respectively; $\Delta \ddot{x}$, $\Delta \dot{x}$, and Δx are the acceleration, velocity and offset displacement of the robot end-effector, respectively; f_r is the reference contact force; and f_e is the actual contact force, which obtained after filtering. In the actual sampling system, the difference can be calculated as follows (Song et al., 2019):

$$\begin{aligned} \Delta \dot{x}(k) &= \frac{\Delta x(k) - \Delta x(k-1)}{T_s} \\ \Delta \ddot{x}(k) &= \frac{\Delta \dot{x}(k) - \Delta \dot{x}(k-1)}{T_s} \end{aligned} \tag{2}$$

where k is used to represent the k -th sampling period, and T_s represents the sampling period. Substituting Equation 2 into Equation 1, can be calculated online as

$$\Delta x(k) = \frac{eT_s^2 + b_d T_s \Delta x(k-1) + m_d (2\Delta x(k-1) - \Delta x(k-2))}{m_d + b_d T_s + k_d T_s^2} \tag{3}$$

where $e = f_r - f_e$. If the parameters of the contact environment are well-defined, the contact force can be well-tuned by selecting appropriate impedance parameters. However,

the skin environment is usually unknown, and simply maintaining target impedance parameters does not guarantee a well-controlled contact force.

Therefore, a robot force control algorithm is proposed to compensate for the offset displacement of the robot $\Delta x(k)$. A deep reinforcement learning algorithm and a traditional compensation algorithm based on a physical model of the skin are integrated into the proposed algorithm. The flow chart of robot force control is shown in Figure 1. The actual force f_e is processed by a first-order low-pass filter to remove high-frequency noise. The difference between the actual force and the reference force is passed through the impedance controller to obtain the offset displacement of the robot. The offset displacement is compensated by integrating the DQN strategy and a compensation strategy based on the physical model of the skin. The compensations of the two different DQNs are a^1 and a^2 , the compensation based on the physical model of the skin is u^s , and the compensation after fusing offset displacement and strategy is u_f . This compensation is sent to the internal displacement controller of the robot, thereby indirectly adjusting the contact state between the robot and the outside world.

3 Decision-making process of different strategies

3.1 Robot displacement compensation process with DQN strategies

Manually optimizing the compensation displacement selection is very tedious and time-consuming, whereas the reinforcement learning algorithm can independently identify the optimal control strategy. The reinforcement learning algorithm uses the Markov decision process as its theoretical framework. In the Markov decision process, the contact force state between the robot and the skin is denoted by s , the agent selects the robot action a according to the current contact state, and the robot executes action a to change the robot state. Simultaneously, the agent obtains an immediate reward r and then continues to choose the action according to the state at the next moment. The final trajectory τ obtained by the agent is $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \dots, s_T, a_T, r_T\}$, where r_t is the instant reward at the t -th moment, $t \in [0, T]$. The robot-skin interaction process is used to maintain the actual force within a certain range, so the instant reward can be set as the distance between the actual force and the reference force:

$$r_t = -k_r^* |f_r^t - f_e^t| \quad (4)$$

where k_r is the proportional factor. The contact state s can be set as the force error and the change of the force error, namely,

$$s_t = [e_t, \dot{e}_t] \quad (5)$$

where e_t is the force error at time t , $e_t = f_r^t - f_e^t$, \dot{e}_t is the change of the force error, $\dot{e}_t = e_t - e_{t-1}$. The robot action a is the impedance control compensation. Given a policy π , the discounted reward

received by the trajectory τ of an interaction between the agent and the environment is:

$$R(\tau) = \sum_{t=0}^{T-1} \gamma^t r_{t+1} = \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t, s_{t+1}) \quad (6)$$

where γ is a discount factor between 0 and 1. When the time is t , the contact state is s_t , and the action selection is a_t , the expectation $E(R_t | s_t, a_t)$ of the defined discounted return R is the state-action value function, that is, the Q-function:

$$Q(s_t, a_t) = E[R(t) | S_t = s | A_t = a] \quad (7)$$

where E is the expectation and S and A are the sets of states and actions, respectively.

In the Q-learning algorithm, for each state s , the agent adopts the ε -greedy strategy. In the first action value function table, an action a_t is selected, and then the action a_t is executed and transferred to the next state s_t . In the second action value function table, an action a_{t+1} that maximizes $Q(s_{t+1}, a_{t+1})$ is selected according to the state s_{t+1} , and the predicted value and target value are used to update the Q-value function. The prediction value uses the current state and the known Q-value function to estimate the Q-value of an action being taken in the current state, and the prediction value is $Q(s_t, a_t)$. The target value updates the Q-value function, which is $r_t + \gamma \max Q(s_{t+1}, a_{t+1})$, and the Q-value function gradually adjusts the Q-value through the difference between the predicted value and the target value:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_t + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (8)$$

where γ is the discount factor ($0 \leq \gamma \leq 1$) and α represents the learning rate of the model.

Through the learned Q-value function, the agent selects the action with the highest Q-value according to the current state to be the optimal strategy π^* :

$$\pi^* = \arg \max Q(s_t, a_t) \quad (9)$$

However, the state space of robot-skin contact is high-dimensional. To calculate the value function $Q(s, a)$ in the state and action space, the neural network fitting method can be used to fit the action value function. However, if directly using one neural network updates the Q-learning algorithm, that is, the Q-value $r_t + \gamma \max Q(s_{t+1}, a_{t+1})$ and target Q value $Q(s_t, a_t)$ are the same network structure with the same parameters, the predicted value and the target value will change together, which increases the possibility of model oscillation and divergence to some extent. To address this, the predicted value deep neural network $Q(s, a, \theta)$ and the target value deep neural network $Q(s, a, \theta^-)$ are used. When training parameters, samples are usually strongly correlated and non-static; if the data are applied directly, the model will have difficulty converging and the loss values will constantly fluctuate. The DQN algorithm introduces a mechanism for replaying experience: at each stage, the predicted

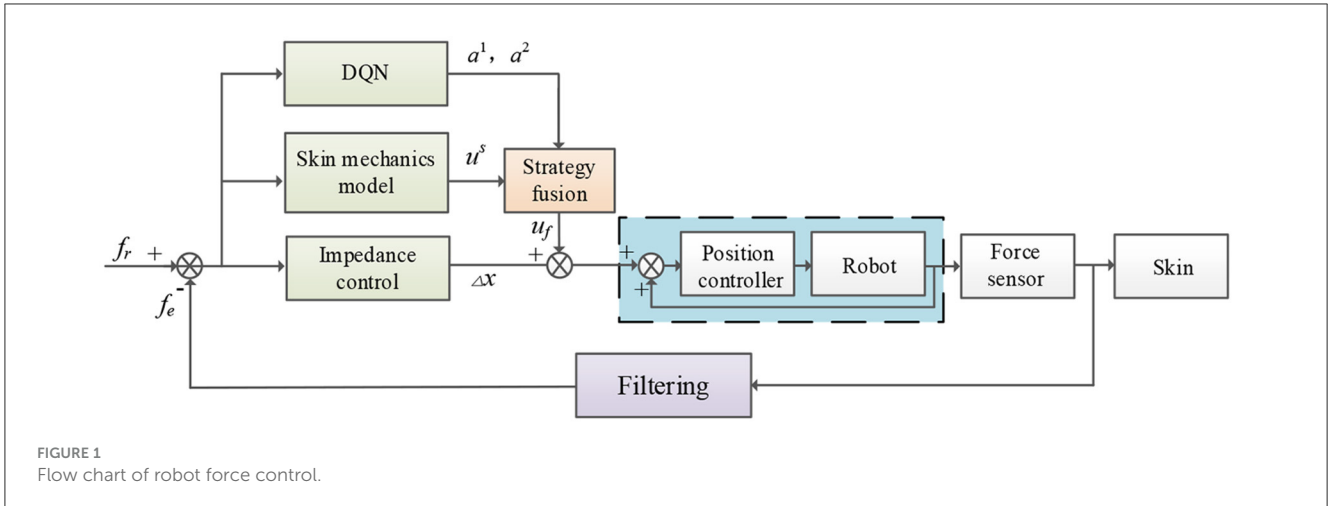


FIGURE 1
Flow chart of robot force control.

value deep neural network executes action a through the ϵ -greedy strategy, namely:

$$a = \begin{cases} \arg \max Q(s, a, \theta) & l - \epsilon \\ \text{random action} & \epsilon \end{cases} \quad (10)$$

After the experience sample data are obtained, the state and action data are stored in the experience pool. When the predictive value network needs to be trained, minibatch data are randomly selected from the experience pool for that training. On the one hand, introducing the experience pool replay mechanism makes backing up rewards easy; on the other hand, using a small number of random samples helps eliminate the correlation and dependence between samples. The loss function of the deep neural network for the predicted value is set to Mnih et al. (2015):

$$L(\theta) = E[(y - Q(s, a, \theta))^2] \quad (11)$$

where L represents the loss function and y is the value of the target network, as follows:

$$y_t = r_t + \gamma^* \max \left[\hat{Q}(s_{t+1}, a_{t+1}, \theta^-) \right] \quad (12)$$

In the initial state, the parameter θ of the predicted value network is the same as the parameter θ^- of the target network. Equation 11 is used to optimize the parameters of the predicted value network by gradient descent, and the parameter θ in the predicted network is updated. After the agent collects G group experience samples and N training iterations, the θ of the prediction network is copied to the θ^- of the target network, i.e., $\hat{Q} = Q$. As the above steps are repeated, the parameters of the predictor network are continuously updated to improve the predictive power and performance of the network, whereas the parameters of the target value network are relatively stable and are only periodically copied from the predictor network. The fitting ability of the Q-value function is

gradually optimized, and the agent selects the action with the highest Q-value as the current optimal decision according to the current state.

The neural network is constructed by a multilayer feedforward neural network, which consists of an input layer, multiple hidden layers, and an output layer. The contact state of the robot is passed through the input layer to the output layer along with connections between neurons in the hidden layer. Finally, the Q-value is output. In the hidden layer, the neural network first calculates the net activation value Z^l of the neurons in the l -th layer according to the activation value U^{l-1} of neurons in layer $(l-1)$ -th and then uses an activation function to obtain the activation value of neurons in the l -th layer. Let the input be the state value of the robot, that is, $U^0 = s$; information is disseminated by continuously iterating the following equation (Li et al., 2012):

$$\begin{aligned} Z^l &= W^l U^{l-1} + B^l \\ U^l &= \varphi(Z^l) \end{aligned} \quad (13)$$

where W^l is the weight of the l -th layer; b^l is the bias of the l -th layer; Z^l is the net activation value of the l -th layer; U^l is the activation value of the l -th layer; and φ is the activation function. The ReLU activation function is selected:

$$\text{ReLU}(Z) = \begin{cases} Z & Z \geq 0 \\ 0 & Z < 0 \end{cases} \quad (14)$$

The parameters of the neural network are trained by backpropagation, and the partial derivative of the loss function for each parameter in the network is calculated. Then, the chain rule is used to backpropagate these partial derivatives to each layer in the network, thereby updating the parameters to minimize the loss function. The error term δ^l for the l -th layer is calculated by backpropagation, and the sensitivity of the final loss to the neurons in layer l is defined as Shi (2021):

$$\delta^l \triangleq \frac{\partial L(\theta)}{\partial Z^l} \quad (15)$$

The derivative of each layer parameter is:

$$\begin{aligned} \frac{\partial L(\theta)}{\partial W^l} &= \delta^l (U^{l-1})^T \\ \frac{\partial L(\theta)}{\partial b^l} &= \delta^l \end{aligned} \quad (16)$$

where δ^l is the error term of neurons in the l layer. Finally, the neural network parameters are updated:

$$\begin{aligned} W^l &\leftarrow W^l - \alpha \delta^l (U^{l-1})^T + \lambda W^l \\ b^l &\leftarrow b^l - \alpha \delta^l \end{aligned} \quad (17)$$

where α is the learning rate and λ is the regularization coefficient.

3.2 Dynamics models of reinforcement learning

The agent of reinforcement learning must go through trial and error when improving the policy and conducting multiple experiments in the actual interaction to achieve the desired result. However, frequent trial and error processes will not only negatively impact the interactive experience but also cause damage and pain to the human skin due to repeated friction. Therefore, fast convergence of the algorithm during robot-skin contact is crucial. Since the DQN algorithm is a model-free algorithm, it must conduct multiple experiments to obtain sufficient data. To accelerate the convergence, dynamic models of the reinforcement learning environment can be constructed so that DQN can iteratively train in a virtual environment, reducing the number of actual training and improving the practicality of the algorithm.

3.2.1 BP neural network dynamics model

Since skin has biological characteristics, the mechanical characteristics of skin are non-linear. The dynamic model of the robot is also non-linear, so the contact process between the two can be set as a non-linear system; the BP neural network has non-linear mapping capabilities, so it can construct the relationship between the contact state and robot displacement. The network inputs the contact state e_t , \dot{e}_t , and the compensation displacement a of the robot, and the output state is e_{t+1} , \dot{e}_{t+1} . The dynamics model constructed by the BP neural network is composed of the data of multiple impedance algorithms, and the fitted model is as follows:

$$s_{t+1} = \varphi(s_t, a_t, NeT1.W, NeT1.b) \quad (18)$$

where $NeT1.W$ and $NeT1.b$ are the weight and bias parameters in the BP neural network. The network can be updated through Equations 13–17. After the BP neural network constructs the environmental dynamics model, the DQN algorithm can be used to train the strategy offline in this model. Once the compensation strategy satisfies Equation 9, the output compensation displacement a^l can be obtained.

3.2.2 LSTM neural network dynamics model

The presence of noise information in the robot state data is likely to lead to inaccurate information in the network results. A recurrent neural network can establish the correlation of state model information in time series and integrate multiple state information according to the characteristics of spatiotemporal context information; through doing so, the network can reduce noise interference and purify the sample set so that a more accurate state model can be obtained. A certain connection exists between the robot state data; the long short-term memory (LSTM) neural network has short-term memory ability, so it can build further connections between the data. Neurons in LSTM can receive information not only from other neurons but also from themselves, forming a network structure with loops. The LSTM better aligns with the structure of the biological neural network than with the feedforward neural network, and the fitted model is as follows:

$$S_{t+1} = \phi(S_t, A_t, NeT2.W, NeT2.b) \quad (19)$$

LSTM can effectively capture and store long-term dependencies by introducing memory units and gating mechanisms. The gating mechanism controls the path of information transmission; the forget gate f_t determines whether to retain the memory unit C_{t-1} at the previous moment, and the input gate controls how much information must be saved at the current moment. The output gate o_t controls how much information the memory state C_{t-1} at the current moment must output to the hidden state H_t . The memory unit in LSTM is a linear structure that can maintain the chronological flow of information. When $f_t = 0$ and $i_t = 1$, the memory unit clears the historical information; when $f_t = 1$ and $i_t = 0$, the memory unit copies the content of the previous moment, and no new information is written. The key operations of LSTM are expressed as follows (Shi et al., 2015):

$$\begin{aligned} i_t &= \sigma(NeT2.W_{xi}X_t + NeT2.W_{hi}H_{t-1} \\ &\quad + NeT2.W_{ci} \odot C_{t-1} + NeT2.b_i) \\ f_t &= \sigma(NeT2.W_{xf}X_t + NeT2.W_{hf}H_{t-1} \\ &\quad + NeT2.W_{cf} \odot C_{t-1} + NeT2.b_f) \\ o_t &= \sigma(NeT2.W_{xo}X_t + NeT2.W_{ho}H_{t-1} \\ &\quad + NeT2.W_{co} \odot C_{t-1} + NeT2.b_o) \\ C_t &= f_t \odot C_{t-1} + i_t \odot \tanh(NeT2.W_{xc}X_t \\ &\quad + NeT2.W_{hc}H_{t-1} + NeT2.b_c) \\ H_t &= o_t \odot \tanh(C_t) \end{aligned} \quad (20)$$

where, i_t , f_t , and o_t represent the input gate, forget gate, and output gate in the LSTM, respectively; t represents the period, X_t denotes the input at the current moment, C_t represents the memory state, H_t represents the hidden state, and $NeT2.W_f$, $NeT2.W_i$, $NeT2.W_c$, and $NeT2.W_o$ are the weights of the forget gate, input gate, estimated state, and output gate, respectively. \odot denotes the Hadamard product. σ is a logistic function with an output interval of (0,1), and H_{t-1} is the external state at the previous moment.

After the LSTM neural network constructs the dynamics model, the DQN algorithm can also be used to train offline in the

constructed model, and the output compensation displacement a^2 can be obtained.

3.3 Robot displacement compensation strategy with a skin mechanics model

For the skin contact environment, the amount of skin extrusion deformation first increases and then slowly increases as pressure increases, which has the non-linear elastic characteristics of compliant materials. The Hunt-Crossley skin mechanics model defines the relationship between the force on the skin and the depth of extrusion as a power function, which can conform to the non-linear elastic and viscous mechanical properties of skin-like soft material objects. In the one-dimensional direction, when the skin is squeezed, the deformation force of the skin is [Schindeler and Hashtrudi-Zaad \(2018\)](#):

$$f_s = k_s(|x - x_e|)^\beta + b_s(|\dot{x} - \dot{x}_e|)^\beta \quad (21)$$

where f_s is the force generated by skin deformation; x is the coordinate of the robot when it is deformed; x_e are the initial coordinates of the skin when it is not deformed by force; $|x - x_e|$ is the amount of deformation; k_s and b_s are the elasticity and damping coefficients, respectively; and β is the power exponent, determined by the nature of the skin in the local contact area. The parameters of the skin of different parts of the human body differ in certain ways, and the parameters in [Equation 21](#) also change, so directly using [Equation 21](#) to calculate the parameters online is cumbersome. Therefore, when the robot moves along the skin, the axis is fine-tuned in the Z-axis direction, that is, $\dot{x} \approx 0$; for calculation ease, [Equation 21](#) is simplified to:

$$f_s = k_s(|x - x_e|)^\beta \quad (22)$$

The parameters k_s and β are fitted by an offline collection of deformation and contact force data of different parts of the body by using the least square method. Therefore, the online compensation displacement of the robot is:

$$u^s = \sqrt[\beta]{\frac{f_e}{k_s}} - \sqrt[\beta]{\frac{f_r}{k_s}} \quad (23)$$

where u^s is the compensation displacement based on displacement compensation with skin mechanics.

4 Force control strategy fusion process based on the GMM/GMR algorithm

All strategies for the environment dynamics model built by the BP neural network or the LSTM neural network are offline training strategies, and some errors will still exist in the actual process regardless of which strategy is chosen. Although the robot

displacement compensation strategy under the physical model of skin mechanics is an online strategy, experience data cannot improve it. Therefore, the fusion strategy is employed to effectively fuse the prediction results of different data sources or models to improve the accuracy and robustness of the overall prediction.

The GMM/GMR algorithm is flexible, highly efficient, adaptable to multivariate data, interpretable and robust. These advantages can support the fusion of robot force control strategies. GMM is a probability model based on a Gaussian distribution that assumes the data are a mixture of several Gaussian distributions. By training the data, the GMM can learn the parameters (mean and covariance matrix), as well as the weight, of each Gaussian distribution. These parameters can be used to describe the data distribution and to generate new samples.

Under the three strategies, the robot may obtain three different predicted robot force trajectories, that is, $\{t_n, a_n^1\}_{n=1}^{N_m}$, $\{t_n, a_n^2\}_{n=1}^{N_m}$, $\{t_n, u_n^s\}_{n=1}^{N_m}$, and the predicted values of the deep neural network model and the skin mechanics model. Here, n is the number of samples, N_m is the length of the trajectory, t is the time information, a^1 , a^2 , and u^s are the output compensation displacements of the robot, u represents the three kinds of compensation displacements, and the GMM can model the joint probability distribution $P(t, u)$ of the input and output variables in the sample as follows ([Man et al., 2021](#)):

$$p(t, u) \sim \sum_{m=1}^M \pi_m N(\mu_m, \Sigma_m) \quad (24)$$

where M is the number of Gaussian components in the GMM. π_m , μ_m , and Σ_m represent the prior probability, mean and covariance of the m -th Gaussian component, respectively, and μ_m and Σ_m are defined as follows:

$$\mu_m = \begin{bmatrix} \mu_{t,m} \\ \mu_{u,m} \end{bmatrix} \quad \Sigma_m = \begin{bmatrix} \Sigma_{tt,m} & \Sigma_{tu,m} \\ \Sigma_{ut,m} & \Sigma_{uu,m} \end{bmatrix} \quad (25)$$

The parameters of the GMM are iteratively optimized through the expectation-maximization (EM) algorithm ([Hu et al., 2023](#)), the posterior probability of each sample point belonging to each Gaussian component is calculated, and the mean value, covariance matrix and mixing coefficient of the Gaussian component are updated. After obtaining the trained GMM model, GMR is used to make a regression prediction on the robot force trajectory. The posterior probability of each Gaussian component is first calculated, and the weighted sum of the posterior probability is used to obtain the weighted Gaussian component mean and covariance matrix. A new trajectory point is then obtained by sampling from each Gaussian component. GMR is used to predict the conditional probability distribution of the corresponding trajectory of a new input:

$$p(u^*|t^*) = \sum_{m=1}^M h_m(t^*) N(\bar{\mu}_m(t^*), \bar{\Sigma}_m) \quad (26)$$

where t^* and u^* are the predicted time and compensation displacement, respectively, and h_m , $\bar{\mu}_c$, and $\bar{\Sigma}_m$ are calculated as follows:

$$\begin{aligned}
 h_m(t^*) &= \frac{\pi_m N(t^* | \mu_{t,m}, \Sigma_{tt,m})}{\sum_{i=1}^M \pi_i N(t^* | \mu_{t,i}, \Sigma_{tt,i})} \\
 \bar{\mu}_c(t^*) &= \mu_{u,m} + \Sigma_{tt,m} \Sigma_{tt,m}^{-1} (t^* - \mu_{t,m}) \\
 \bar{\Sigma}_m &= \Sigma_{uu,m} - \Sigma_{ut,m} \Sigma_{tt,m}^{-1} \Sigma_{tu,m}
 \end{aligned}
 \tag{27}$$

For calculation convenience, Equation 26 can be approximated as

$$p(u^* | t^*) \approx N(\hat{\mu} \hat{\Sigma})
 \tag{28}$$

where $\hat{\mu} = \sum_{m=1}^M h_m(t^*) \bar{\mu}_c(t^*)$, $\hat{\Sigma} = \sum_{m=1}^M h_c(t^*) \bar{\mu}_m^T(t^*) + \bar{\Sigma}_m - \hat{\mu} \hat{\mu}^T$, the central distribution of u^* is obtained according to the probability distribution in $p(u^* | t^*)$, and u_f is the final fusion strategy.

5 Experimental setup of the force control based on the GMM/GMR algorithm

A schematic diagram of the experiment is shown in Figure 2. In this experiment, the robot squeezes the skin vertically along the Z direction at a speed of 2 mm/s. When the robot reaches the reference force f_r along the Z direction, i.e., point Q_a in the figure, the robot stops moving in the Z direction, enters force control mode to move horizontally along the X direction at a speed of 2 mm/s for 5 s until reaching point Q_b , the robot then leaves the human skin vertically. The second trajectory is in the opposite direction, starting from Q_b to Q_a . The force sensor is an ME-FKD40, and the force signal is collected by a backoff module and transmitted to the robot controller, the control system works at a frequency of 50 Hz, and the robot force control only tested while moving from point Q_a to Q_b or from point Q_b to Q_a .

The force control based on the GMM/GMR algorithm experimental process is shown in Figure 3. Multiple sets of impedance data parameters are used to obtain the robot contact states and displacements in the Z-direction to get experience data. When different impedance strategies are implemented, the difference between the force on the end of the robot and the reference force e_t , the rate of change of the error \dot{e}_t and the offset displacement Δx_t of the robot are collected, which can be used for fitting the BP and LSTM neural network model. The least squares algorithm is used to fit parameters in the skin mechanics model. The DQN strategy is obtained through offline training, and the compensation strategy based on the skin mechanics model is obtained through online calculation. If the force error obtained by the force control based on the GMM/GMR algorithm is greater than the expected threshold



FIGURE 2 Schematic diagram of the robot tracking process along the skin.

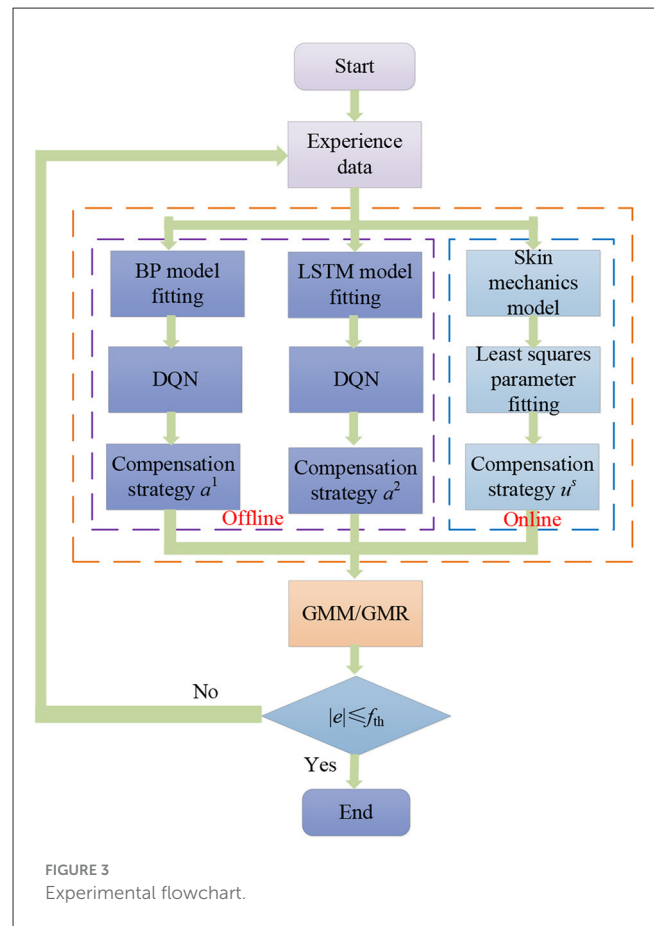
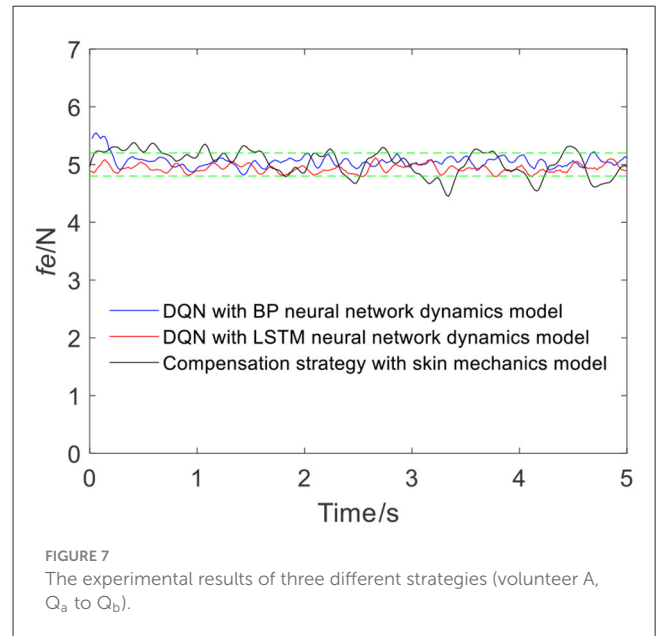
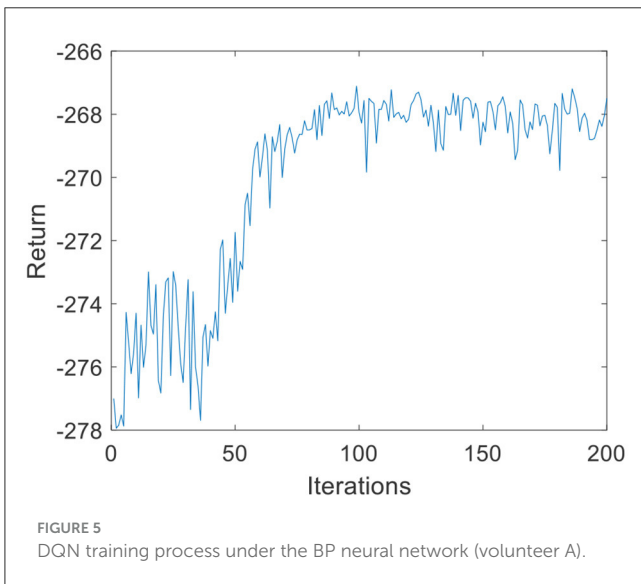
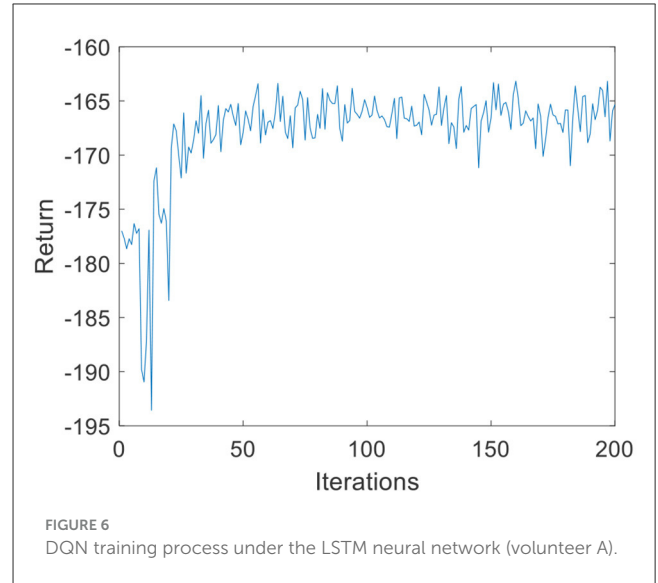
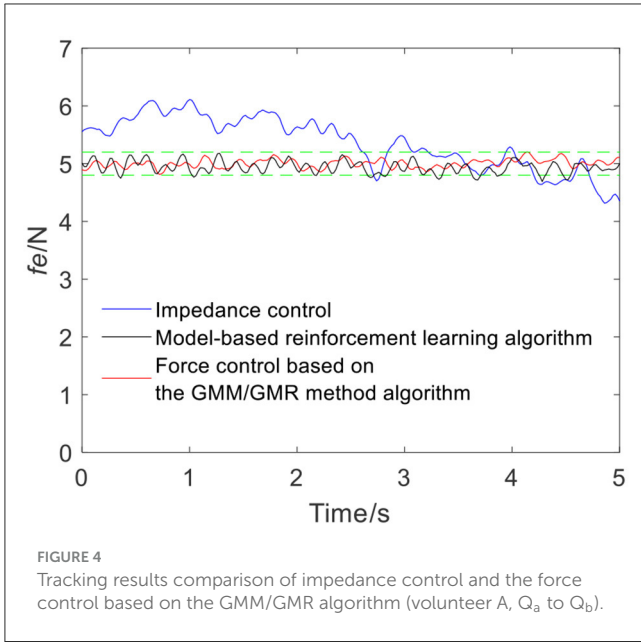


FIGURE 3 Experimental flowchart.

± 0.2 N, the obtained data can be added to the database. Then, the BP neural network can be updated again, and experiments can be iterated until the error between the force in the Z-direction and the reference force is within the set range, namely, ± 0.2 N.

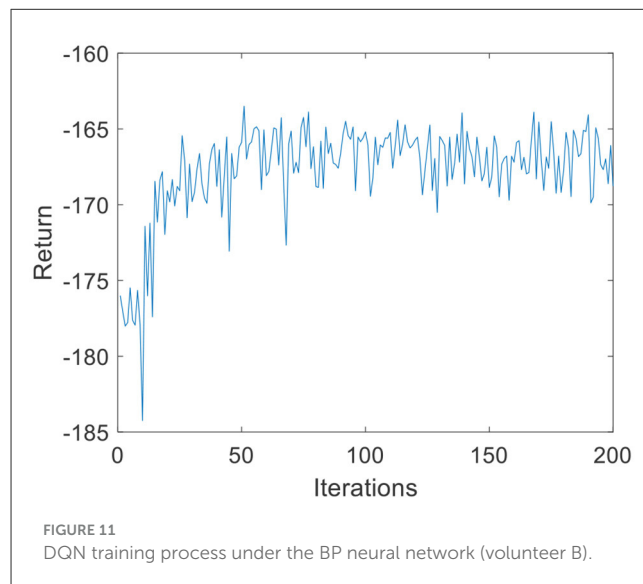
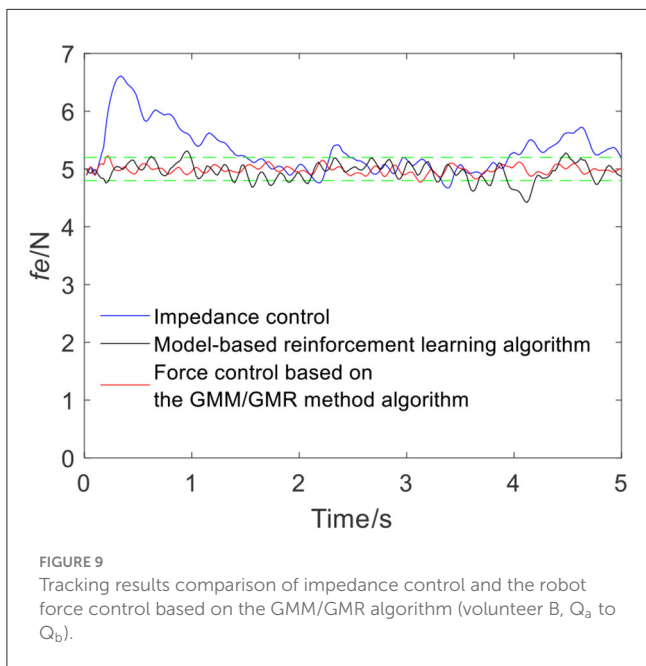
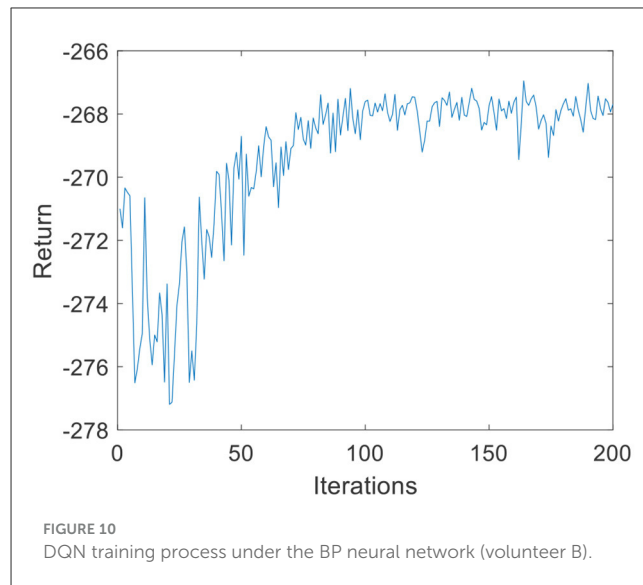
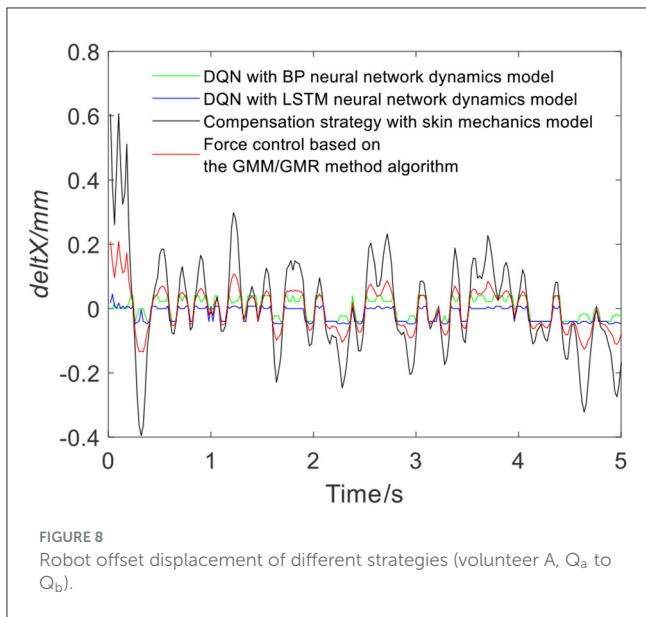


6 Robot-skin contact experiment results and analysis

To ensure the volunteers' safety, when the robot applies force on the skin surface, a gentle force application strategy is adopted, and the reference force of the robot is set to 5 N, i.e., $f_r = 5$ N. In the impedance control strategy, the parameters are manually adjusted to $m_d = 10$, $b_d = 6$, and $k_d = 700$ according to experience. When the robot moves along the skin from point Q_a to Q_b , the tracking force obtained by impedance control is illustrated by the blue line in Figure 4. It can be seen from the force signal that the robot maintains contact with volunteer A, meanwhile, the force exhibits certain fluctuations. The comparison between impedance control and the force control based on the GMM/GMR algorithm fusing different compensation strategies is shown in

Figure 4. The force control based on the GMM/GMR algorithm is significantly smoother than impedance control, and the control effect is significantly improved.

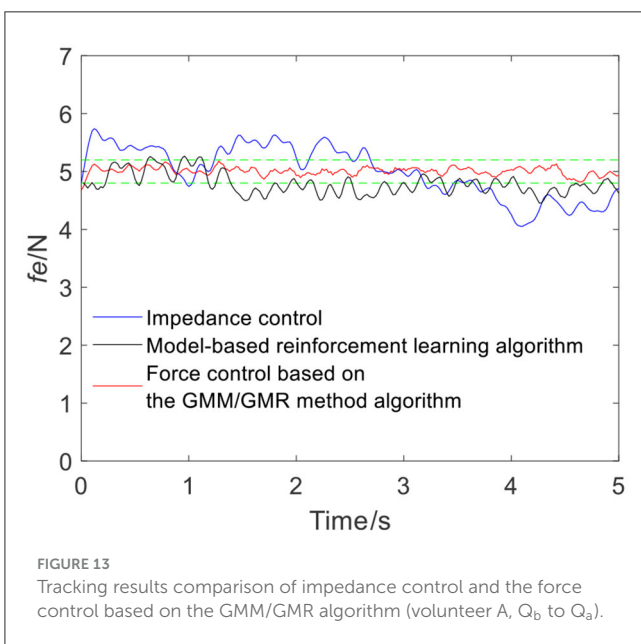
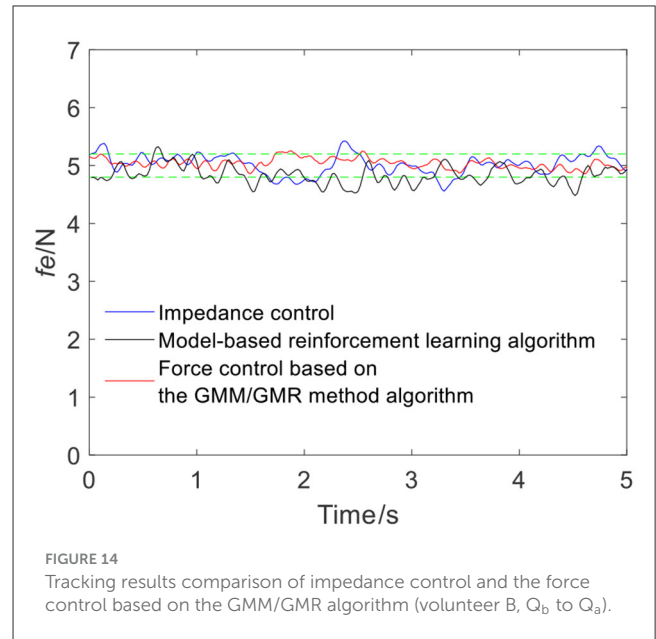
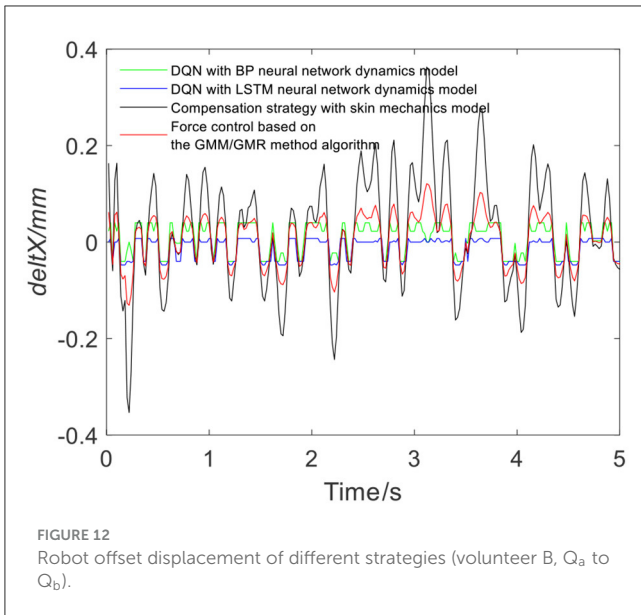
Due to the small amount of input and output data, in the environmental dynamics model constructed by the BP neural network, the range of action a is $[0:0.01:0.2]$ with a total of 20 actions. When the force error is negative, a chooses the opposite direction, which can reduce invalid searches. The output is the state at the next moment. The middle node of the neural network is set to 30, and the number of layers of the neural network is set to 2. In the LSTM neural network, the intermediate nodes of the neural network are set to 20. For the input data s of the DQN, the Q-values, which are 1-dimensional data, are the output. Due to the parameter dimensions and the small amount of data, the deep neural network is much smaller than the image dimension; therefore, the number of layers of the neural network is set to 2, and the number of nodes



in each layer is set to 30. The step size of the DQN is set to $T = 200$, $G = 200$, $k_r = 10$ in Equation 3, ϵ is 0.1 in Equation 11, and the total number of iterations N is 200. The iterative process of the DQN algorithm under the environmental dynamics model of the BP neural network is shown in Figure 5. As the number of iteration data increases, the algorithm converges after ~ 50 iterations. The iterative process of the DQN algorithm under the environmental dynamics model of the LSTM neural network is shown in Figure 6. As the number of iteration data increases, the algorithm converges after ~ 40 iterations. In the online strategy based on the skin mechanics model, the force data of different skins are chosen to fit the parameters of the skin mechanics model, namely, $k_s = 0.015$ and $\beta = 2.5$ in Equation 24. In the GMM/GMR algorithm, M is equal to 2, and the length of the trajectory $N_m = 20$.

Figure 7 shows the results of three different force control strategies that are run separately. All three algorithms achieve good results, but they exhibit relatively large fluctuations. Figure 8 depicts the offset displacement strategies of three different strategies under the robot force control based on the GMM/GMR algorithm. The DQN with the BP neural network dynamics model and the LSTM dynamics model are relatively conservative, while the algorithm based on the skin mechanics model is relatively radical.

To verify the versatility of the proposed algorithm, the arms of different volunteers are tracked with the robot force control based on the GMM/GMR algorithm. The parameters are consistent with the first experiment. The comparison results of the impedance control process and robot force control based on the GMM/GMR algorithm are shown in Figure 9. Similar to the effect of volunteer A, the obtained force also fluctuates to a certain extent with impedance control. The robot force control based on the GMM/GMR algorithm's force signal is significantly smoother than



that of the impedance control strategy, the error against the reference force is stable within a certain range, and the control effect is significantly improved. The return values of reinforcement learning with the BP neural network and the LSTM neural network dynamics model are shown in Figures 10, 11. Both gradually converge after ~ 50 iterations. The robot offset displacement of different strategies can be computed online as shown in Figure 12. The experiment of robot trajectory Q_b to Q_a is shown in Figures 13, 14, although the contact force signal obtained with impedance control under different external conditions has good contact effect, the force signal of force control strategy fusion algorithm is relatively smoother.

The intelligent algorithm for comparison is a model-based reinforcement learning algorithm, which is constructed by

combining a neural network and a cross-entropy method for control parameter search. The obtained force is shown as the black line in Figures 4, 9, 13, 14. Compared with the impedance control algorithm in the four groups of experiments, the model-based reinforcement learning algorithm has better results. However, the force signal of the model-based reinforcement learning algorithm exceeds the threshold in some trajectories, such as in the second half of the force tracking on volunteer B in Figure 9, and the robot force control based on the GMM/GMR algorithm is more stable and has better versatility.

The error comparison between the impedance control, model-based reinforcement learning algorithm and robot force control based on the GMM/GMR algorithm is shown in Table 2. The error of force tracking with the robot force control based on the GMM/GMR algorithm includes the maximum absolute value $|e|_{max}$, the mean absolute error $|\bar{e}|$ and the standard deviation of error σ_e . In the robot force control experiment of the trajectory from Q_a to Q_b on different volunteers, the mean absolute errors $|\bar{e}|$ of the robot force control based on the GMM/GMR algorithm were significantly reduced by 87.5 and 80%, respectively, compared with that of the impedance control strategy. In the robot force control experiment of the trajectory from Q_b to Q_a on different volunteers, the mean absolute errors $|\bar{e}|$ of the robot force control based on the GMM/GMR algorithm were reduced by 85.7 and 45.7%, respectively. And all three types of errors had been significantly reduced, too. Compared with model-based reinforcement learning, the mean absolute errors $|\bar{e}|$ of the robot force control based on the GMM/GMR algorithm were reduced by 35.7, 65.7, 74.4, and 60%, respectively.

The reason why the robot force control based on the GMM/GMR algorithm is better than the traditional impedance control is that the impedance control adjustment range is small. Although impedance control can ensure that the robot and skin remain in contact facing volunteers A and B, a fixed impedance parameter cannot ensure the accuracy of the robot-skin contact

TABLE 2 Error comparison of force control algorithms between impedance control, model-based reinforcement learning algorithm and the force control based on the GMM/GMR algorithm.

| Algorithm | $ e _{max}/N$ | $ \bar{e} /N$ | σ_e/N |
|---|---------------|---------------|--------------|
| Impedance control (volunteer A, Q_a to Q_b) | 1.1 | 0.49 | 0.46 |
| Model-based reinforcement learning algorithm (volunteer A, Q_a to Q_b) | 0.31 | 0.095 | 0.1 |
| Robot force control based on the GMM/GMR algorithm (volunteer A, Q_a to Q_b) | 0.21 | 0.061 | 0.074 |
| Impedance control (volunteer A, Q_b to Q_a) | 0.95 | 0.38 | 0.45 |
| Model-based reinforcement learning algorithm (volunteer A, Q_b to Q_a) | 0.58 | 0.13 | 0.16 |
| Robot force control based on the GMM/GMR algorithm (volunteer A, Q_b to Q_a) | 0.26 | 0.054 | 0.068 |
| Impedance control (volunteer B, Q_a to Q_b) | 1.6 | 0.33 | 0.39 |
| Model-based reinforcement learning algorithm (volunteer B, Q_a to Q_b) | 0.54 | 0.25 | 0.18 |
| Robot force control based on the GMM/GMR algorithm (volunteer B, Q_a to Q_b) | 0.37 | 0.064 | 0.087 |
| Impedance control (volunteer B, Q_b to Q_a) | 0.44 | 0.14 | 0.17 |
| Model-based reinforcement learning algorithm (volunteer B, Q_b to Q_a) | 0.51 | 0.19 | 0.15 |
| Robot force control based on the GMM/GMR algorithm (volunteer B, Q_b to Q_a) | 0.25 | 0.076 | 0.087 |

process. The accuracy of the model-based reinforcement learning strategy depends on whether the model conforms to reality. When the robot contact state exceeds the range of the model, there will be an error between the offline reinforcement learning strategy and the actual demand. However, when the robot force control based on the GMM/GMR algorithm faces unknown skin environments, the skin mechanics model can propose compensation strategies online and modify the robot state in real time, at the same time, the DQN with the BP and LSTM neural network models can provide the historical experience of offline learning. When the GMM/GMR algorithm integrates the two, the robot can obtain the advantages of both. The fusion strategy for volunteers A and B is relatively stable and has relatively good versatility.

7 Conclusions and future work

A robot force controller based on the GMM/GMR algorithm is proposed that combines different compensation strategies and is applied to robot-skin contact scenarios. The initial robot force control strategy is established by impedance control, the reinforcement learning algorithm and traditional control strategy are fused to compensate for the impedance control. Two environmental dynamics models of reinforcement learning are constructed to simulate the contact process between the robot and the skin, and accelerate the offline convergence of the reinforcement learning algorithm. The GMM/GMR algorithm fuses online and offline compensation strategies to improve the robustness and versatility of the algorithm and to adapt to different skin environments.

The experimental results show that the robot force control based on the GMM/GMR algorithm has good versatility and accuracy. Under 100 offline iterations, the reinforcement learning algorithm can select effective control parameters. The force can quickly converge to the reference force, and its error is stable within the range of $\pm 0.2N$. The method has also achieved good results with different volunteers. Furthermore, for the force obtained

by using the reinforcement learning algorithm, the maximum absolute value, the mean absolute error and the standard deviation of error are lower than those of the method of impedance control and the model-based reinforcement learning algorithm, the mean absolute errors of the force signal in the four groups are significantly reduced, further illustrating the strong stability of the proposed algorithm.

In the current work, we use constant force control, which is suitable for some scenarios of robot-skin contact, such as auxiliary treatment and robot local massage. In future research, we will study variable force to make the use range of the force controller wider.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Ethics statement

The studies involving humans were approved by School of Mechanical and Engineering, Guangzhou University. The studies were conducted in accordance with the local legislation and institutional requirements. The participants provided their written informed consent to participate in this study.

Author contributions

MX: Conceptualization, Methodology, Software, Validation, Writing—original draft, Writing—review & editing. XZ: Conceptualization, Methodology, Validation, Writing—original draft, Writing—review & editing. TZ: Conceptualization, Methodology, Writing—review & editing. SC: Conceptualization, Methodology, Software, Validation,

Writing—review & editing, YZ: Methodology, Software, Writing—review & editing. WW: Funding acquisition, Methodology, Validation, Writing—original draft, Writing—review & editing.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. This work was supported by the National Natural Science Foundation of China (NNSFC), China; Contract grant number: 82172526 and Guangdong Basic and Applied Basic Research Foundation, China; Contract grant number: 2021A1515011042.

References

- Anand, A. S., Hagen Myrestrand, M., and Gravidahl, J. T. (2022). "Evaluation of variable impedance- and hybrid force/motioncontrollers for learning force tracking skills," in *2022 IEEE/SICE International Symposium on System Integration (Narvik)*, 83–89.
- Bogdanovic, M., Khadiv, M., and Righetti, L. (2020). Learning variable impedance control for contact sensitive tasks. *IEEE Robot. Automat. Lett.* 5, 6129–6136. doi: 10.1109/LRA.2020.3011379
- Christoforou, E. G., Avgousti, S., Ramdani, N., Novales, C., and Panayides, A. S. (2020). The upcoming role for nursing and assistive robotics: opportunities and challenges ahead. *Front. Digit. Health* 2:585656. doi: 10.3389/fgth.2020.585656
- Ding, Y., Zhao, J., and Min, X. (2023). Impedance control and parameter optimization of surface polishing robot based on reinforcement learning. *Proc. Inst. Mech. Eng. B* 237, 216–228. doi: 10.1177/09544054221100004
- Hou, Y., Xu, H., Luo, J., Lei, Y., Xu, J., and Zhang, H.-T. (2020). Variable impedance control of manipulator based on DQN. *Intelligent Robotics and Applications. ICIRA 2020. Lect. Not. Comput. Sci.* 12595, 296–307. doi: 10.1007/978-3-030-66645-3_25
- Hu, Y., Wang, Y., Hu, K., and Li, W. (2023). Adaptive obstacle avoidance in path planning of collaborative robots for dynamic manufacturing. *J. Intell. Manuf.* 34, 789–807. doi: 10.1007/s10845-021-01825-9
- Huang, Y., Li, J., Huang, Q., and Souères, P. (2015). Anthropomorphic robotic arm with integrated elastic joints for TCM remedial massage. *Robotica* 33, 348–365. doi: 10.1017/S0263574714000228
- Ishikura, T., Kitamura, Y., Sato, W., Takamatsu, J., Yuguchi, A., Cho, S. G., et al. (2023). Pleasant stroke touch on human back by a human and a robot. *Sensors* 23:1136. doi: 10.3390/s23031136
- Joodaki, H., and Panzer, M. B. (2018). Skin mechanical properties and modeling: a review. *Proc. Inst. Mech. Eng. H* 232, 323–343. doi: 10.1177/0954411918759801
- Jutinico, A. L., Jaimes, J. C., Escalante, F. M., Perez-Ibarra, J. C., Terra, M. H., and Siqueira, A. A. G. (2017). Impedance control for robotic rehabilitation: a robust Markovian approach. *Front. Neurobot.* 11:43. doi: 10.3389/fnbot.2017.00043
- Kerautret, Y., Di Rienzo, F., Eyssautier, C., and Guillot, A. (2020). Selective effects of manual massage and foam rolling on perceived recovery and performance: current knowledge and future directions toward robotic massages. *Front. Physiol.* 11:598898. doi: 10.3389/fphys.2020.598898
- Khoramshahi, M., Henriks, G., Naef, A., Salehian, S. S. M., Kim, J., and Billard, A. (2020). "Arm-hand motion-force coordination for physical interactions with non-flat surfaces using dynamical systems: toward compliant robotic massage," in *2020 IEEE International Conference on Robotics and Automation (Paris)*, 4724–4730.
- Li, H. Y., Dharmawan, A. G., Paranawithana, I., Yang, L., and Tan, U. X. (2020). A control scheme for physical human-robot interaction coupled with an environment of unknown stiffness. *J. Intell. Robot Syst.* 100, 165–182. doi: 10.1007/s10846-020-01176-2
- Li, J., Cheng, J., Shi, J., and Huang, F. (2012). Brief introduction of back propagation (BP) neural network algorithm and its improvement. *Adv. Comput. Sci. Inform. Eng.* 169, 553–558. doi: 10.1007/978-3-642-30223-7_87
- Li, S., Li, J., Li, S., and Huang, Z. (2017). Design and implementation of robot serial integrated rotary joint with safety compliance. *J. Cent. South Univ.* 24, 1307–1321. doi: 10.1007/s11771-017-3536-3
- Li, Y., Ganesh, G., Nathanael Jarrass, é, Haddadin, S., Albu-Schaeffer, A., and Burdet, E. (2018). Force, impedance, and trajectory learning for contact tooling and haptic identification. *IEEE Trans. Robot.* 34, 1170–1182. doi: 10.1109/TRO.2018.2830405
- Li, Z., Liu, J., Huang, Z., Peng, Y., Pu, H., and Ding, L. (2017). Adaptive impedance control of human-robot cooperation using reinforcement learning. *IEEE Trans. Ind. Electron.* 64, 8013–8022. doi: 10.1109/TIE.2017.2694391
- Liu, X., Ge, S. S., Zhao, F., and Mei, X. (2021). Optimized interaction control for robot manipulator interacting with flexible environment. *IEEE/ASME Trans. Mechatron.* 26, 2888–2898. doi: 10.1109/TMECH.2020.3047919
- Luo, Y., Xu, D., Zhu, J., and Lei, Y. (2021). "Impedance control of slag removal robot based on Q-Learning," in *2021 China Automation Congress (Beijing)*, 1338–1343.
- Man, Z., Fengming, L., Wei, Q., Yibin, L., and Rui, S. (2021). "Robot bolt skill learning based on GMM-GMR," in *Lecture Notes in Computer Science: Intelligent Robotics and Applications (Cham: Springer)*, 235–245.
- Meng, Y., Su, J., and Wu, J. (2021). "Reinforcement learning based variable impedance control for high precision human-robot collaboration tasks," in *2021 6th IEEE International Conference on Advanced Robotics and Mechatronics (Chongqing)*, 560–565.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236
- Roveda, L., Maskani, J., Franceschi, P., Abdi, A., Braghini, F., Molinari Tosatti, L., et al. (2020). Model-based reinforcement learning variable impedance control for human-robot collaboration. *J. Intell. Robot. Syst.* 100, 417–433. doi: 10.1007/s10846-020-01183-3
- Schindeler, R., and Hashtrudi-Zaad, K. (2018). Online identification of environment Hunt-Crossley models using polynomial linearization. *IEEE Trans. Robot.* 34, 1–12. doi: 10.1109/TRO.2017.2776318
- Sheng, Q., Geng, Z., Hua, L., and Sheng, X. (2021). "Hybrid vision-force robot force control for tasks on soft tissues," in *2021 27th International Conference on Mechatronics and Machine Vision in Practice (Shanghai)*, 705–710.
- Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W., and Woo, W. (2015). *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. Advances in Neural Information Processing Systems* (Red Hook, NY: Curran Associates, Inc.), 28.
- Shi, Z. (2021). *Intelligence Science* (Amsterdam: Elsevier), 89–115.
- Song, P., Yu, Y., and Zhang, X. (2017). "Impedance control of robots: an overview," in *2017 2nd International Conference on Cybernetics, Robotics and Control (Chengdu)*, 51–55.
- Song, P., Yu, Y., and Zhang, X. (2019). A tutorial survey and comparison of impedance control on robotic manipulation. *Robotica* 37, 801–836. doi: 10.1017/S0263574718001339
- Stephens, T. K., Awasthi, C., and Kowalewski, T. M. (2019). "Adaptive impedance control with setpoint force tracking for unknown soft environment interactions," in *2019 IEEE 58th Conference on Decision and Control (Nice)*, 1951–1958.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Suomalainen, M., Karayiannidis, Y., and Kyrki, V. (2022). A survey of robot manipulation in contact. *Robot. Autonom. Syst.* 156:104224. doi: 10.1016/j.robot.2022.104224

Weng, L., Tian, L., Hu, K., Zang, Q., and Chen, X. (2020). "Overview of robot force control algorithms based on neural network," in *2020 Chinese Automation Congress* (Shanghai), 6800–6803.

Zhao, X., Han, S., Tao, B., Yin, Z., and Ding, H. (2022). Model-based actor-critic learning of robotic impedance control in complex interactive environment. *IEEE Trans. Industr. Electr.* 69, 13225–13235. doi: 10.1109/TIE.2021.3134082

Zhu, X., Gao, B., Zhong, Y., Gu, C., and Choi, K.-S. (2021). Extended Kalman filter for online soft tissue characterization based on Hunt-Crossley contact model. *J. Mech. Behav. Biomed. Mater.* 123:104667. doi: 10.1016/j.jmbbm.2021.104667