Intelligent control of flexible joint based on cooperative learning theory

Qun Shi\textsuperscript{a*} and Xiangjie Jia\textsuperscript{b*}

School of mechatronic engineering and automation, Shanghai University, Shanghai, 200444, China
\textsuperscript{a*}shiqun@staff.shu.edu.cn, \textsuperscript{b*}xiangjie_jia@163.com

Abstract. It is complex and time-consuming for obtaining accurate dynamics of industrial robots. The flexibility of the joint not only increases the adverse effects of nonlinear factors, but also makes the controller design of robot significantly difficult. In order to improve the performance of robots containing joint flexibility, cooperative learning (COL) theory is proposed in this paper. Based on this theory, a model-free intelligent controller is trained and successfully applied to the flexible joints. Compared with the conventional PID controller and RBF controller, the cumulative tracking error produced by the cooperative learning intelligent controller is reduced by 38.25\% and 31.08\%, respectively, and the robustness is improved in the trajectory tracking experiment and the robustness experiment. The experimental results demonstrate the feasibility and effectiveness of the cooperative learning theory.

1. Introduction

Varieties of industrial robots have been widely applied to extensive factory automation fields, where the fast and precise point-to-point and tracking performance of end effectors must be essential to provide the high productivity and quality in the manufacturing [1]. The controller of an industrial robot is designed in a situation where the joints are assumed to be rigid. In fact, vibration is excited when the industrial robot accelerates or decelerates rapidly. The flexibility of the joint is actively incorporated into the robot drive mechanism in order for the safety of the robot's interaction with the environment and humans in non-industrial areas of medical and service classes. As elastic joints become more prevalent in the field of robotics, modeling and identification of accurate dynamics models become more difficult, and the design of controller becomes more complex.

Several techniques have been developed to overcome the difficulty, such as feedback linearization [2], singular ingestion method [3], and adaptive theory [4]. In order to compensate for the flexibility and other nonlinearities of the robot's joints, accurate dynamics models are needed. However, accurate knowledge of the dynamics model of flexible joint is difficult, and factors such as friction, gaps, unmodeled high-frequency characteristics of the flexible joint, and uncertain model parameters of the system need to be considered. Although the control method based on H\textsubscript{\infty} theory [5,6] does not require a high-precision model, the application of the method is too complex, while the designed controller is conservative. Therefore, a model-free intelligent control method is proposed in order to avoid the problems associated with the dynamic modeling of flexible joints.

In recent years, due to the re-emergence of artificial intelligence, intelligent control has developed greatly in the field of robotics. Several intelligent control methods have been used in control systems for robots, including RBF neural network control, expert control, and fuzzy control [7,8], but these methods still have some limitations. First, there are problems such as dimensional disasters, large
information errors and few reliable samples, when machine learning techniques such as neural networks are used in the design of intelligent controllers. Second, as traditional training methods for machine learning-based intelligent controllers are generally used separately, different methods need to be reselected for different usage scenarios. The effectiveness of imitation learning [9,10] with dataset limitations depends on the size and features of the dataset, and models trained based on imitation learning have a certain confidence level in prediction and may not make the best decisions. The complex relationship between the cost function or reward function and the optimal decision needs to be determined in reinforcement learning [11], and the ideal cost function is difficult to implement in practice. Therefore, some researchers have tried to mix two or more machine learning methods. For example, Jonathan [12] used standard gradient descent methods based on imitation learning to find a stochastic policy that performs constrained optimization in the trust region. Although the performance of this strategy is the same compared to the expert strategy, it ignores the design of the loss function for apprentice learning, which can lead to the inability of the method to identify the global optimal solution. The hybrid approach of multiple machine learning methods can solve the shortcomings of a single machine learning method to a certain extent if there is a lack of logical order and recursive relationship among the methods, but the final global optimal training result is difficult to be guaranteed.

In order to solve the problem of using a single machine learning method to train the intelligent controller, a new machine learning theory is proposed: Collaborative Learning (COL), i.e., the intelligent controller is trained by different machine learning methods in the form of phases. COL differs significantly from the traditional hybrid machine learning approach in that it includes two major features: first, there is a strong logical relationship between the training methods in different stages; second, the training methods in each stage can be chosen flexibly. The purpose of this paper is to obtain an intelligent controller with better control performance through cooperative learning methods.

The rest of the paper is organized as follows. In Section II, the complete theory of cooperative learning is described in detail, which consists of an initial training stage, an evolutionary stage, and a retraining stage. In Section III, the cooperative learning theory is applied to the control of flexible joints. In Section IV, the control effects of different controllers are compared and the effectiveness of the cooperative learning algorithm is verified. In Section V, the whole paper is summarized.

2. Cooperative learning theory

The collaborative learning theory consists of three stages, which are the initial training phase, the evolutionary phase, and the retraining phase. Two objects are defined, the trainer and the learner, respectively. The structure of the collaborative learning theory is shown in figure 1, where the process of learner training by the coach and evolution of the coach is the core of the theory. Training a good learner is the purpose of the theory, i.e., learners are trained by coaches of different levels to achieve a stepwise progression. The process of each stage is described in detail.

2.1. Initial training stage

Before the training starts, the learner network structure and activation function need to be determined, and the input states \( S_k \) and output actions \( a_k \) of the network need to be defined; then the states \( S_k \) and actions \( a_k \) under the primary coach model are collected to form an expert data set \( \{ S_1, \tilde{a}_1, S_2, \tilde{a}_2, \ldots \} \); finally, the imitation learning method is used to train the learner network model parameters.
2.2. Coach evolution stage
The senior coach network model is established, and the junior coach is trained with reference to the optimal control theory and the reinforcement learning (RL) method to evolve the senior coach, and the retraining data set is obtained to set higher learning standards for the learner.

2.3. Retraining stage
Using the dataset obtained from the advanced trainer in the second stage, the learner is retrained using iterative learning or reinforcement learning methods. In which, the iterative function needs to be defined, and by continuously iterating, the learner network keeps improving and performing better in iterative learning methods. Reinforcement learning is an unsupervised learning method. In order to guide the learner in the direction of optimization, the appropriate reward function needs to be defined, in reinforcement learning. Although the choice of training methods is flexible, the ultimate goal of collaborative learning theory is to train learners with better performance. The complete process of collaborative learning is shown in figure 2.

3. Implementation of cooperative learning theory
In this section the flexible joints of the wire drive are used as the object of study. A servo control algorithm was designed which is based on cooperative learning theory.

3.1. Control structure of intelligent controller
There are two hidden layers in the network structure of the intelligent controller, which has three variables in the input layer, 32 neurons in the first hidden layer, 64 neurons in the second hidden layer, and one variable in the output layer. ReLU activation function and tanh activation function are used by the input layer and hidden layer, respectively.

The state of the load is considered as the input of the intelligent controller of the neural network, and the input state is defined as follows.

\[ S(k) = \{\theta_i(k), \omega_i(k), e_y(k)\} \]  \hspace{1cm} (1)

where \(\theta_i(k)\), \(\omega_i(k)\), \(e_y(k)\) denote the actual position, actual velocity and position error of the joint at the kth moment, respectively, and the intelligent control structure is shown in figure 3. \(Gp\) means the flexible joint model and the output \(y'_{cmd}\) of the network is the control quantity. The intelligent controller is trained in Tensorflow using back propagation technique and desired trajectory, AdamOptimizer optimizer is used to optimize the network parameters, and L2 parametric regularization is used to avoid over-fitting.
3.2. Implementation of collaborative learning theory
The neural network intelligent controller is defined in the previous section, cooperative learning theory is used for the training of flexible joint intelligent controllers in the next section.

3.2.1. Stage I, initial training. First, the PID controller is connected to the control system in series as a primary trainer, and discrete data of the control process are collected. Then, each discrete moment state value $S(k) = \{\theta_1(k), \omega_1(k), e_y(k)\}$, and action value $\hat{a}(k) = y_{cmd}(k) - y'_{cmd}(k)$ are saved to the database as primary expert training data, and the control structure is shown in figure 4.

\[ y'_{cmd} + e_y \rightarrow coach \rightarrow Gp \rightarrow \hat{a}(k) \rightarrow Action \]

\[ S(k) \]

Figure 4. Control structure of the initial training dataset.

The data in the database is used to train the intelligent controller using the imitation learning method, and the training process is specified as follows:

1) Determining the primary trainer model: Primary coach parameters are initialized under conditions that ensure system stability.

2) Acquisition of primary expert data: The state and action values of primary coach are collected and uniformly noted as $D(k) = \{S(k), \hat{a}(k)\}$, and the dataset $D(k)$ is taken as the primary expert dataset.

3) Imitation learning: The neural network intelligent controller $\pi_{\theta}$ is trained using primary expert data and behavioral cloning, and $\pi_{\theta}$ is denoted by the symbol $\pi_{\theta}^{' \prime}$ after the first stage of training. With the same desired input, the control performance of $\pi_{\theta}^{' \prime}$ can reach or approximate that of the primary trainer.
According to the control structure of intelligent controller shown in figure 3, the process of state input to action output is represented using the equation normalization. The input layer is considered as layer 0 of this neural network. The following formula is implemented in each layer:

\[
\begin{align*}
    z^{[i]} &= w^{[i]}v^{[i-1]} + b^{[i]} \\
    v^{[i]} &= f_D(z^{[i]}) \\
    v^0 &= S(k)
\end{align*}
\]

where the superscript \(i\) denotes the \(i\)th layer neuron, \(i \in N^+\), and \(i \leq 3\); \(f_D\) denotes the activation function matrix; \(w^{[i]}\) is the weighting matrix, and the dimensionality is related to the number of neurons; \(v^{[i]}\) denotes the output matrix of the node; \(b^{[i]}\) is the bias matrix.

The \(e(k)\) is obtained by subtracting the expected output \(\hat{a}(k)\) from the output \(a(k)\) of \(\pi_\theta\):

\[
e(k) = \hat{a}(k) - a(k)
\]

Define the error energy function of \(\pi_\theta\) as follows:

\[
E(k) = \frac{1}{2}e^2(k)
\]

The \(w^{[i]}, b^{[i]}\) are corrected using back propagation technique and gradient descent method. The iterative training will end until the function \(E(k)\) converges to the optimal or desired value.

After the simulation of the algorithm in the first stage of cooperative learning, the controller \(\pi_\theta'\) can achieve the control performance of the primary expert controller. As in figure 5, the step response of \(\pi_\theta'\) (red solid line) is compared with that of the primary expert controller (black dotted line).

### 3.2.2. Stage II, expert model evolution

Through the training of the first stage of collaborative learning, \(\pi_\theta\) has approached or reached the level of the primary coach, at which point the learner’s ability is limited by the level of the primary trainer, and therefore the coach needs to evolve. In the second stage, the coach is built as a three-layer neural network, which contains one unit in the input layer, 32 units in the hidden layer, and one unit in the output layer.

Referring to the typical second-order system control object, the second-order system input-output tracking error \(E_r = \{e(1), \ldots, e(k)\}\) is collected with the optimal damping ratio \(\xi = 0.707\), and \(E_r\) is used as the ideal training dataset for the advanced trainer. The transfer function of a typical second-order system is as follows.

\[
G_s = \frac{\omega_n^2}{s^2 + 2\xi\omega_n + \omega_n^2}
\]

In this section, the trainer is trained using reinforcement learning (RL). The system is rewarded or punished by a performance evaluation module in the RL method. During the control process, the reward to the intelligence is maximized depending on the state of the environment [13,14].

Define the relationship between the action \(a_t\) and the environmental state \(s_t\) at moment \(t\) as follows:

\[
Q(s_t, a_t) = r_t + \gamma \max \{Q(s_{t+1}, a_t)\}, a \in A
\]

The optimal control strategy \(\pi(s_t)\) and \(Q(s_t, a_t)\) satisfy the following equation:

\[
\pi(s_t) = \arg \max [Q(s_t, a_t)], a \in A
\]

where \(r_t\) denotes the agent reward value, \(\gamma \in [0,1]\) denotes the discount rate, and \(A\) denotes the set of actions corresponding to all knowledge reserves possessed by the agent. After receiving the reward function \(r_t\), the agent will update its knowledge reserve and adjust its strategy to output the action \(a_{t+1}\) at moment \(t + 1\). The computational process is iterated until a satisfactory target state is reached.

The coach is trained based on the RL method and the structure schematic is shown in figure 6, where the agent makes a strategy \(a_t\) (control quantity \(y_{cmd}\)) based on the environmental state \(s_t\) (tracking error \(e_p\)) and the reward value \(r_t\) at the current moment and acts in the environment \(G_p\).

It is desired that the actual system output will achieve the response of a typical second-order system at a damping ratio \(\xi = 0.707\), the reward function is defined as follows:

\[
r_t = 2 \frac{|s_p(k) - e(k)|}{\rho}
\]
where \( s_t(k) \) denotes the \( k \)th tracking error of the actual system, \( e(k) \) denotes the \( k \)th data of the ideal dataset \( E_r \), and \( \rho \) serves to adjust the convergence rate of the algorithm. From equation (10), it can be seen that when the actual position tracking error is closer to the ideal tracking error, the reward value \( r_1 \) will keep approaching 1, otherwise the reward value \( r_1 \) is close to 0. With the RL method, coach is trained and network weight is updated. The senior coach model and the control state sequence \( R_k = [y_t, v_t, e_t] \) are finally obtained. The control state sequence \( R_k \) is used for the retraining of learners in the third stage.

### 3.2.3. Stage III, retraining based on advanced expert models.

In this section, iterative learning is used as the retraining method, and the iterative function is defined as follows:

\[
 u(k+1) = u(k) + \alpha (y(k) - \hat{y}_t(k)) + \beta (v(k) - \hat{v}_t(k)) + \gamma (e(k) - \hat{e}_t(k)) \tag{11}
\]

where \( y(k) \) denotes the actual motion trajectory of the connecting rod, \( [\hat{y}_t(k), \hat{v}_t(k), \hat{e}_t] \) denotes the sequence of motion states \( R_k \) obtained in the second stage, \( u(k) \) denotes the output sequence before optimization by the intelligent controller, \( u(k+1) \) denotes the output sequence after iterative optimization by the intelligent controller, and \( \alpha, \beta, \gamma \) denote the iterative learning rate.

Based on the primary network model \( \pi_{\theta}' \), the state dataset obtained from the second stage of cooperative learning is combined with the third stage of cooperative learning for retraining. The iteration equation (11) is used to optimize the weight parameters and action sequences of the intelligent controller. After several iterations of calculation, the advanced network model \( \pi_{\theta}'' \), which is better than the primary network model \( \pi_{\theta}' \), is finally obtained, i.e. \( \pi_{\theta} \xrightarrow{\pi_{\theta}''} \pi_{\theta}'' \). The step response of the intelligent controller after the first stage of training was compared with the step response after the third stage of training, as shown in figure 7, after retraining by advanced experts. It can be seen that the system response is significantly better after retraining.

![Figure 7. Comparison of system response in the stage I and stage III.](image)

### 3.3. Modeling of control object

In order to analyse the experimental object, the flexible joint is equated to a linear torsion spring and the motor is considered as a torque power source. The two-inertia dynamics of the control object is modeled as follows [15]:

\[
 T_e = J_m \ddot{\theta}_m + B_m \dot{\theta}_m + \frac{K_s}{R_g} \left( \frac{\theta_m}{R_g} - \theta_l \right) + f_m \tag{12}
\]

\[
 K_s \left( \frac{\theta_m}{R_g} - \theta_l \right) = J_l \ddot{\theta}_l + B_l \dot{\theta}_l + f_l \tag{13}
\]

where \( T_e \) is the motor-side output torque, \( J \) denotes inertia, \( B \) means damping, \( f \) means disturbance term, subscript "m" means motor-side variable, "l" means load-side variable; superscript "" denotes differentiation. \( k_s \) denotes joint stiffness, and \( R_g (R_g > 1) \) means transmission ratio.
Figure 8. Block diagram of intelligent control.

Usually, three-loop PID control is used for two-inertia control object. In order to improve the position tracking performance at the end of the load, an intelligent controller is considered in this paper based on the cooperative learning theory. The control structure is shown in figure 8, which the basis for the design of the controller in the experimental stage. The shaded part is the traditional control scheme, which is regarded as the new control object $G_p$.

4. Experiment

The experimental object of this paper is a wire rope driven robot joint, as shown in figure 9. Maxon's RE series rare metal brushed motors were used, and 15-bit absolute encoders were mounted on the motors. The drive communicates with the host PC at a frequency of 4000 Hz. In addition, an acceleration sensor is installed at the end of the load for testing the vibration data at the end.

4.1. Experimental design

The experiments are mainly divided into two parts: 1) position tracking experiments; 2) robustness experiments. The performance of four controllers is compared, including: PID controller, RBF controller, MPC controller and COL controller.

4.2. Position tracking experiments

The position tracking experiments with sinusoidal signals were carried out on the actual flexible joint platform under no-load conditions. The position tracking curves and position tracking error accumulation curves of the four methods under sinusoidal signals are shown in figure 10 and figure 11. From the figure, the tracking performance of the PID controller and the RBF neural network controller
Figure 10. Position tracking curve of sinusoidal signal.

Figure 11. Cumulative tracking error of sinusoidal signal.

is bad, and the intelligent controller based on cooperative learning theory is better than the first two, but slightly worse than the MPC controller.

The error data for tracking sinusoidal signals for different controllers are summarized in Table 1. In terms of absolute error, PID controller (0.092 rad) > RBF controller (0.081 rad) > COL controller (0.062 rad) > MPC controller (0.041 rad). In terms of cumulative error, the PID controller (51.361 rad) > RBF controller (46.014 rad) > COL controller (31.715 rad) > MPC controller (13.514 rad). It can be seen that both the maximum absolute error and the cumulative error are smaller for the cooperative learning-based intelligent controller than the PID controller and the RBF neural network controller, but larger than the MPC controller. The cumulative error produced by the intelligent controller designed based on collaborative learning theory decreased by 38.25% and 31.08% compared to the PID controller and RBF neural network controller, respectively.
Table 1. Error analysis table for sine signals.

| Controller | Cumulative error (rad) | Error standard Deviation(rad) | Maximum absolute Error(rad) |
|------------|------------------------|------------------------------|----------------------------|
| PID        | 51.361                 | 0.057                        | 0.092                      |
| RBF        | 46.014                 | 0.051                        | 0.081                      |
| COL        | 31.715                 | 0.035                        | 0.062                      |
| MPC        | 13.514                 | 0.011                        | 0.041                      |

4.3. Robustness experiments

In this section, the robustness of the four controllers is verified by varying the joint stiffness and load inertia with constant controller parameters. To facilitate the robustness experiments, a flexible linkage is added to the end of the linkage and is equated to the joint flexibility.

4.3.1. Robustness experiments with variable stiffness: The equivalent stiffness of the joint is changed by varying the linkage length and the end position information is obtained by integrating the accelerometer. The four controllers were tested using the same experimental scheme. The maximum overshoot curves and error variance curves of the four controllers are given in figure 12(a) and figure 12(b), respectively. It can be seen that the end-of-load overshoot and error variance decrease with increasing joint stiffness under different control schemes. In the range of stiffness from $2.2 N \cdot m/\text{rad}$ to $3 N \cdot m/\text{rad}$, the overshoot and error variance of COL and MPC are largely unchanged, while they are significantly changed in the PID and RBF control schemes. This shows that the robustness of the COL intelligent controller proposed in this paper for joint stiffness changes is slightly worse than MPC, but better than RBF and PID controllers.

4.3.2. Variable inertia robustness experiment: Varying the end load inertia, the position data at the end of the load is measured with an accelerometer, and the inertia varies from $1.69 \times 10^{-3} K g \cdot m^2$ to $2.75 \times 10^{-3} K g \cdot m^2$ incrementally. The maximum overshoot curves and error variance curves for the four controllers are given in figure 13(a) and figure 13(b), respectively. The comparison shows that the overshoot and error variance increase for all four controllers with increasing inertia; the COL intelligent control is more robust than PID and RBF and slightly less robust than MPC control.

(a) Comparison of overshoot. (b) Comparison of error variance.

Figure 12. Variable stiffness experiments.
5. Conclusion
This paper focuses on the problem that the accurate dynamics of flexible joints is derived difficultly, resulting in low tracking accuracy and poor control stability during the motion of the robot. Solutions and experimental results are summarized. A model-free intelligent control method is proposed, and intelligent controller for flexible joint is trained through three stages of cooperative learning theory. Four control schemes, COL, PID, RBF, and MPC, are compared by simulation and experiment, and it shows the effectiveness of cooperative learning theory and intelligent controller. The intelligent controller trained based on cooperative learning theory can be used for the control of flexible jointed robots, and its control performance is not much different from MPC controller, but better than traditional PID and RBF controller, which can maintain good tracking accuracy and robustness.

References
[1] Yabuki, A., Ohishi, K., Miyazaki, T., Yokokura, Y., Kanmachi, T. and Ando, I. (2020) Vibration Suppression Control Using Resonance Frequency Damping Disturbance Observer for Robot Servo System. IEEJ Journal of Industry Applications, vol. 10.
[2] Assanimoghaddam, M. (2017) Algebraic Estimation and Control of Single-Link Flexible Joint Robots.
[3] Spong, M. W. (1987) Modeling and Control of Elastic Joint Robots. Journal of Dynamic Systems, Measurement, and Control, vol. 109, no. 4, pp. 310-318.
[4] Chaoui, H., Sicard, P. (2012) Adaptive neural network control of flexible-joint robotic manipulators with friction and disturbance. pp. 2644-2649.
[5] Morimoto, S., Takeda, Y. (1996) Two-degrees-of-freedom speed control of resonant mechanical system based on $H_\infty$ control theory. Electrical Engineering in Japan, vol. 117, no. 1, pp. 112-121.
[6] Symens, W., van Brussel, H., Swevers, J. (2004) Gain-scheduling control of machine tools with varying structural flexibility. CIRP Annals, vol. 53, no. 1, pp. 321-324.
[7] Kumar, A. and Sharma, R. (2018) Linguistic Lyapunov reinforcement learning control for robotic manipulators. Neurocomputing, vol. 272, pp. 84-95.
[8] Farias, G., Garcia, G., Montenegro, G., Fabregas, E. (2020) Position control of a mobile robot using reinforcement learning. IFAC-PapersOnLine, vol. 53, no. 2, pp. 17393-17398.
[9] Manschitz, S., Kober, J., Gienger, M., Peters, J. (2014) Learning to sequence movement primitives from demonstrations. in 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 14-18 Sept, pp. 4414-4421
[10] Cui, Y., Isele, D., Niekum, S., Fujimura, K. (2019) Uncertainty-Aware Data Aggregation for Deep Imitation Learning.
[11] Sutton, R. S., Barto, A. G. (1998) Reinforcement Learning: An Introduction. IEEE Transactions on Neural Networks, vol. 9, no. 5, p. 1054.
[12] Ho, J., Gupta, J. K., Ermon, S. (2016) Model-Free Imitation Learning with Policy Optimization. arXiv e-prints, p. arXiv:1605.08478.
[13] Wu, Y., Mansimov, E., Liao, S., Grosse, R., Ba, J. (2017) Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation. 08/17.
[14] Xie, L., Wang, S., Rosa, S., Markham, A., Trigoni, N. (2018) Learning with Training Wheels: Speeding up Training with a Simple Controller for Deep Reinforcement Learning. in 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 6276-6283.
[15] Saito, E., Katsura, S. (2012) Vibration Control of Two-Mass Resonant System Based on Wave Compensator. IEEJ Transactions on Industry Applications, vol. 132, pp. 473-479.