A Novel Distributed and Compound Control of an Adaptive Neural Network and PD for Manipulators

Pu Zhao1, Yunfei Zhou1
1 School of Mechanical Science and Engineering, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan 430074, China

Abstract. In this report, a new compound control scheme is proposed for manipulators based on a proportional-differential (PD) controller and a back-propagation neural network. By modifying the dynamic model of manipulators, a second-order function of tracking errors is obtained and approximated by the neural network. A PD-type reference model is introduced to update the weights of the neural network. By using additional PD controllers, rapid response of the control scheme can be realized. Simulation results highlight performance of the controller to compensate the nonlinear terms and its simpleness in the parameter tuning process. To be concluded, the controller is suitable for distributed control and can be used as supplementary of traditional PD controllers.

1. Introduction
Artificial neural networks, as a kind of tools of computational intelligence, have been credited in various applications as powerful tools capable of providing robust controllers for mathematically ill-defined systems that may be subjected to structured and unstructured uncertainties[1]. The universal approximation theorem has been the main driving force behind the increasing popularity of such methods, as it shows that they are theoretically capable of uniformly approximating an continuous real function to any degree of accuracy. This has led to recent advances in the area of intelligent control[2]. Various neural networks have been applied in the control of manipulators, which have led a satisfactory performance[3]. Ponce proposed the design and implementation of a new neural network based on a trigonometric series expansion and applied it to the control of a nonlinear system[4]. Hui suggested an observer-based adaptive controller using time-delay neuro-fuzzy network for manipulators with flexible joints[5]. Chaoui designed an adaptive controller based on two backpropagation neural networks for manipulators with Coulomb friction model and flexible joints[6]. Recently, adaptive fuzzy neural networks are used by Chen to approximate a nonlinear stochastic system with unknown functions[7]. In [8], neural networks are used to approximate nonlinear models of friction and backlash hysteresis. In [9], a trajectory tracking controller, including a neural adaptive compensator, is proposed by Rossomando for a unicycle link mobile robot. Le proposed an adaptive controller with orthogonal neural network for manipulators [10]. In general, the presence of high, particularly unstructured, nonlinearities, such as in the form of Coulomb friction on a manipulator makes it unrealizable to solve the inverse dynamics of the system. Hence, the controllers designed by the researchers adopted desired position as the input of the neural networks, such as Chaoui and Ye[6,11]. However, this leads to unachievable of distributed control of manipulators. In this report, by modifying the dynamic model of manipulators, a second-order function of tracking errors is obtained and approximated by neural networks. Different with existing methods, this method utilize the tracking errors rather than desired trajectory of the manipulators and thus can be seemed as a distributed controller. As shown in the simulation results,
tracking errors and response speed can be tuned easily by using the same tuning method of PD controllers.

2. Dynamics of manipulators

The dynamics of a serial n-link rigid manipulator can be written as

\[
M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) + F(\dot{q}) + T_d(q, \dot{q}, \ddot{q}) = \tau,
\]

where \( q \), \( \dot{q} \) and \( \ddot{q} \) are \( n \times 1 \) vectors of joint angle, angle velocity and angle acceleration, respectively; \( \tau \) is an \( n \times 1 \) vector of input torques; \( M(q) \) is an \( n \times n \) symmetric positive-definite inertia matrix; \( C(q, \dot{q}) \) is an \( n \times n \) matrix of centripetal and Coriolis torques; \( G(q) \) is an \( n \times 1 \) vector of gravitational torques obtained as the gradient of the robot potential energy due to gravity; \( F(\dot{q}) \) is an \( n \times 1 \) vector for the friction torques; \( T_d(q, \dot{q}, \ddot{q}) \) is an \( n \times 1 \) vector of disturbance torques.

The dynamics of a robotic manipulator are characterized by the following property:

**Property 1:** The inertia matrix \( M(q) \), Coriolis matrix \( C(q, \dot{q}) \) and gravity vector \( G(q) \) are upper and lower bounded, i.e., there exist scalars \( \alpha_1(q) \), \( \alpha_2(q) \), \( \alpha_3(q) \) and \( \alpha_4(q) \) such that

\[
\alpha_1(q)I \leq M(q) \leq \alpha_2(q)I, \quad \|C(q, \dot{q})\dot{q}\| \leq \alpha_3(q)\|\dot{q}\|^2 \quad \text{and} \quad \|G(q)\| \leq \alpha_4(q), \quad \text{where} \quad I \quad \text{is the identity matrix.}
\]

Before we proceed further, we introduce the following realistic assumption:

**Assumption 1:** The norm of the unknown disturbance \( T_d \) is upper bounded by a scalar \( \alpha_5 \), i.e.,

\[
\|T_d\| \leq \alpha_5.
\]

3. Control Strategy

The control strategy is based on the design of a reference model based neural network controller and a PD controller. After analysing the characteristics of the two different controllers and dynamics of manipulators, a new compound control scheme is proposed.

3.1. Neural network and reference model

NN-based controllers can overcome the difficult control problems in nonlinear control systems due to the nonlinear approximation capabilities of NNs. Furthermore, NN controllers have been successfully deal with robot control problems mentioned by many researchers. In this report, the neural network proposed in [12] is selected as the network controller for its Lyapunov stability. The relationship between the inputs and outputs of a neural network can be described by

\[
v_j(i)(k) = \sum_{j=1}^{m} \omega_j(i)(k)x_j(i)(k), \quad O_j(i)(k) = \varphi_j(i)(v_j(i)(k)),
\]

where \( m \) and \( n \) being the number of input and hidden nodes, respectively; \( \varphi \) is the sigmoid activation function. The signal \( O(k) \) represents the output of the neural network. The new weight matrices are computed by the rules modified from

\[
\omega_j^{(2)}(k) = \frac{d(k-1) + \eta e(k-1)}{O_j^{(1)}(k-1) + \delta_j}, \quad \omega_j^{(1)}(k) = \frac{1}{mx_j(k) + \delta_2} \varphi^{-1}\left(\frac{d(k-1) + \eta e(k-1)}{O_j^{(1)}(k-1) + \delta_3}\right),
\]

where \( \eta \) is the network’s learning rate. \( \delta_1, \delta_2 \) and \( \delta_3 \) are small values to avoid numerical singularity problems. Based on the results presented in [12], \( \delta_1, \delta_2 \) and \( \delta_3 \) don’t affect the convergence of the neural network when selected small enough. \( d(k-1) \) is the expected output value of the neural network for
the instant \((k-1)\), while \(e(k-1)\) is the output error of the network and satisfies \(d(k-1) = y(k-1) - e(k-1)\).

Theorem 1: The learning algorithm (3) with a learning rate \(\eta\) satisfying \(0 < \eta < 1\) guarantees the bounded stability of the neural network.

Proof: Consider the following candidate Lyapunov function:

\[
V(k) = \frac{1}{\eta} e^2(k), \quad \Delta V(k) = \frac{1}{\eta} e^2(k) - \frac{1}{\eta} e^2(k-1).
\]

Based on the results presented in [6], it’s easy to obtain the following equations

\[
\Delta V(k) = \frac{1}{\eta} (\eta e(k-1) + d(k-1) - d(k)) - \frac{1}{\eta} e^2(k-1)
\]

\[
\leq \eta e^2(k-1) - \frac{1}{\eta} e^2(k-1) + \frac{4}{\eta} (d(k-1) - d(k))^2 \leq \eta \left(1 - \frac{1}{\eta}\right) e^2(k-1) + \frac{4}{\eta} \Omega^2,
\]

where \(\Omega\) denotes the difference between the desired output on instant \(k\) and \(k-1\). Based on the Property 1 and Assumption 1 mentioned in the Section 2, \(\Omega\) is bounded by a specific value. By using (4), the output error \(e(k)\) satisfies the following equation

\[
e^2(k) \leq \eta^2 e^2(k-1) + 4\Omega^2 \leq \eta^2 (\eta^2 e^2(k-2) + 4\Omega^2) + 4\Omega^2 \leq \eta^2 e^2(0) + \frac{1-\eta^2}{1-\eta^2} 4\Omega^2.
\]

As a result, the output error of the neural network converges exponentially to a bounded value when \(0 < \eta < 1\).

Remark: It should be noted that the region becomes smaller for slowly time-varying systems or for fast neural network sampling rates. Besides, this modified proof process also shows that the learning rate affects the region and convergence speed of the NN.

The reference model is designed to calculate the output error of the neural network. Let \(\Delta q = q_d - q\) denote the links’ position with \(q_d\) being the desired motor position vector. A PD-type reference model is selected to calculate the output error and presented as \(e(k) = s = K_{pd}\Delta q + K_{pd}\Delta \dot{q}\). \(K_{pd}\) and \(K_{pd}\) are proportional gain and derivative gain of the reference model. Since the reference model has the same structure with PD controller, the tuning methods of the parameters are the same.

3.2. Compound control scheme

Robotic manipulator dynamics with multi-joints are highly nonlinear, highly coupling and form an uncertain model system. Although the PID controller has a simple form and robustness in broad operating conditions, single PID controllers cannot reach high control precision in uncertain nonlinear control systems. As a result, a compound control method based on NN and PD is introduced for the position-tracking control of manipulators. To decide the number of input variables of the neural network, modifications of (1) are presented as follows:

\[
\tau=M(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)+F(\dot{q})+T_d(q,\dot{q},\ddot{q})
\]

\[
=f(q,\dot{q},\ddot{q})=g(\Delta q, \Delta \dot{q}, \Delta \ddot{q}),
\]

where \(f\) and \(g\) are unknown nonlinear functions. Based on (7), desired output of the neural network can be treated as a second-order nonlinear filter. Besides, to reduce the input of the networks, the coupling factors for each link are also approximated by the networks. As a result, the input variables for joint \(i\) is \([e_i(k), e_i(k-1), e_i(k-2)]\).

Figure 1 shows a block diagram representation of the control system of each joint. In Figure 1, \(q_d\) and \(q\) represent the desired and actual position of the corresponding manipulator joint, respectively. \(u_{PD}\) and \(u_{NN}\) are the output of PD controller and NN controller. In the control process, the PD controllers
reach system stability and help achieve rapid response. Then, the NN controllers can learn in an on-line continuous learning manner to realize the forward compensation control for the dynamic model of the manipulator.

Figure 1. Proposed control scheme.

4. Simulation results and discussions

In order to show the effectiveness of the proposed control system, simulation experiments are carried out. In this section, we consider the first three links of Puma P560. The dynamics of the manipulator can be obtained by Robotics Toolbox for Matlab. The compound controller of the NN and PD and the single PD controller for the three-link model are given to compare their control performance. In the simulation experiments, sampling time is selected as 1ms. As the for the friction, the Coulomb parameter and dynamic friction coefficient are selected as 0.3 and 0.2, respectively. In order to consider the input constraints, 100Nm is selected as the biggest output of the motors. For each link, the NN has 7 hidden neurons and 1 output neuron. The learning parameter of the three NNs and initial weights are all selected as 0.5. The proportional and derivative gains of the three PD controllers are selected as \[200, 1000, 200\] and \[35, 80, 15\], respectively. Besides, the proportional gains and derivative gains of the three reference models are selected as \[20, 100, 20\] and \[3.5, 8, 1.5\] for sample 1, \[60, 300, 60\] and \[10.5, 24, 4.5\] for sample 2, respectively. To avoid the possible instability caused by overquick change of weights of the networks, the maximum value of \(s\) is bounded by 2. The initial conditions are \(q(0) = [0, 0, 0]\) and \(\dot{q}(0) = [0, 0, 0]\). Desired trajectories are selected as \(q_{d1}(t) = q_{d2}(t) = q_{d3}(t) = 0.5(1 - \cos(\pi t))\).

Figure 2 shows the position tracking trajectories and tracking errors of the three joints by using PD controllers. Figure 3 presents the position tracking results of the three joints by using the proposed control scheme. In Figure 4, the output torque of the PD controller and the NN controller of the proposed control scheme is presented. To validate the dynamic response speed of the controller, step signal \(q_{d1}(t) = q_{d2}(t) = q_{d3}(t) = 1\) is applied to the controller and the output of reference models are bounded by \([-100, 100]\), the results are presented in Figure 5.

The above simulation results for the three-link manipulator demonstrate that the single PD control for the three-link manipulator has low trajectory tracking precision and large position-tracking errors. This is because the PD controllers are unable to determine the appropriate PD gains in the case of nonlinear and uncertain controlled plants and can’t compensate static errors. As shown in Figure 3, the proposed control scheme can realize higher trajectory tracking precision and better adaptive control capability due to the powerful capabilities of learning adaptability. Besides, as shown in Figure 5, the step response of the proposed controller is similar to PD controller. This is because the PD controller in the proposed control scheme is then used in rapid dynamic response while NN controller is used to approximate nonlinear parts with lower speed and can be deduced by Figure 4. Besides, as shown in Figure 3, 4 and 5, the trajectory errors become smaller when the gains of reference model are larger. The advantages of the proposed control scheme are that the system model does not need to be identified beforehand and the control gains can be easily tuned.
Figure 2. (a) Tracking results and (b) tracking errors of PD controllers.

Figure 3. (a) Tracking results and (b) tracking errors of the proposed controllers.

Figure 4. Output of NN and PD of proposed controllers: (a) link-1; (b) link-2; (c) link-3.

Figure 5. Step response of PD controllers and the proposed controllers: (a) link-1; (b) link-2; (c) link-3.
5. Conclusions

In this report, a new compound NN and PD control scheme has been proposed. The control scheme uses a PD controller to realize high responses speed and a neural network to approximated unknown nonlinear dynamics. A first order reference model gives the desired dynamics of the error between the desired and actual load positions. Its output is used as supplementary to PD controller and realizes approximation of unknown dynamics. Computer simulation has been carried out on a three-link manipulator with the discontinuous friction model. The results demonstrate the effectiveness of the proposed control scheme and the priorities in the parameter tuning process. Because the tracking errors of each link are the input of the proposed control scheme, distributed control can be realized. The future research work will investigate into the possibility of self-tuning methods of parameters of the proposed scheme.

6. Reference

[1] Chaoui H and Gueaieb W 2008 Type-2 fuzzy logic control of a flexible-joint manipulator Journal of Intelligent and Robotic Systems 51 159-86
[2] Sun C, He W and Hong J 2017 Neural network control of a flexible robotic manipulator using the lumped spring-mass model IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 1863-74
[3] He W, Chen Y and Yin Z 2016 Adaptive neural network control of an uncertain robot with full-state constraints IEEE transactions on cybernetics 46 620-9
[4] Ponce P, Azcue E, Silva J M, Silva J M and Fernandez R 2006 Neural Networks Based on TrigonometricSeries for the Control of a Robot. In: Industrial Technology, 2006. ICIT 2006. IEEE International Conference on: IEEE) pp 2302-7
[5] Hui D, Fuchun S and Zengqi S 2002 Observer-based adaptive controller design of flexible manipulators using time-delay neuro-fuzzy networks Journal of Intelligent and Robotic Systems 34 453-66
[6] Chaoui H, Sicard P and Gueaieb W 2009 ANN-based adaptive control of robotic manipulators with friction and joint elasticity IEEE Transactions on Industrial Electronics 56 3174-87
[7] Chen C P, Liu Y-J and Wen G-X 2014 Fuzzy neural network-based adaptive control for a class of uncertain nonlinear stochastic systems IEEE Transactions on Cybernetics 44 583-93
[8] Huang S and Tan K K 2012 Intelligent friction modeling and compensation using neural network approximations IEEE Transactions on Industrial Electronics 59 3342-9
[9] Rossomando F, Soria C and Carelli R 2012 Neural network-based compensation control of mobile robots with partially known structure IET Control Theory & Applications 6 1851-60
[10] Le Q D and Kang H-J 2018 An Adaptive Controller with An Orthogonal Neural Network and A Third Order Sliding Mode Observer for Robot Manipulators
[11] Ye J 2014 Compound control of a compound cosine function neural network and PD for manipulators International Journal of Control 87 2118-29
[12] Man Z, Wu H R, Liu S and Yu X 2006 A new adaptive backpropagation algorithm based on Lyapunov stability theory for neural networks IEEE Transactions on Neural Networks 17 1580-91

7. Acknowledgments

Research supported by National Natural Science Foundation of China (Project number: 51475177) and State Key Lab of Digital Manufacturing Equipment & Technology, School of Mechanical Science and Engineering, Huazhong University of Science & Technology.