Neural Network-Based Learning Impedance Control for a Robot*

Nan-Feng XIAO** and Isao TODO**

In this paper, a neural network-based learning approach for robot impedance control is presented to accomplish a contact task. Firstly, a discrete time impedance control algorithm is obtained to control the contact task of the robot. Secondly, on-line learning algorithms based on a new evaluation function are developed for the neural networks which adjust the inertia, damping and stiffness parameters of the robot in order to adapt it to the unknown contact environment. Thirdly, experiments are carried out and the effectiveness of the present approach is verified by pressing a spring using a 6 degrees of freedom robot. The adaptiveness, stability and flexibility of the present approach are also confirmed by the experimental results.

Key Words: Mechatronics and Robotics, Measurement and Control, Robot, Sensor, Digital Control, Impedance Control, Neural Networks, On-Line Learning, Force Control, Contact Tasks

1. Introduction

Nowadays, robots are used for operations which can be handled by simple position control strategies. If a robot autonomously works in an unknown contact environment, learning ability of the robot is indispensable. When a robot grasps a rigid tool to perform a contact task, due to unknown contact environment, nonlinear robotic dynamics and various disturbances, it is difficult for the robot to accomplish the contact task in the absence of learning ability.

Many control methods1)-10) have been so far developed for controlling the position and contact force of a robot. Despite the diversities of those methods, it is possible to classify them into the method of impedance control11)-17) and the method of hybrid position/force control18)-20). In the impedance control methods11)-17), the classic iterative learning method21) and the modern connectionist learning techniques22)-27) are used to control the contact force of a robot. In the hybrid control methods28)-38), the visual servoing control is used to make a robot follow the desired position and force trajectories.

In the iterative learning method22), the inertia, damping and stiffness parameters (or the impedance parameters) of the robot are specified to be constants, which may result in poor adaptiveness in the unknown contact environment. The connectionist learning algorithms23)-27) are verified only by the simulation.

In this paper, a neural network-based impedance control learning approach is presented for a robot to perform a contact task. Firstly, a discrete time impedance control law is obtained to control the contact task and the free motion of the robot. Secondly, a new evaluation function-based on-line learning algorithms are developed for the neural networks which are used to adjust the inertia, damping and stiffness parameters of the robot in order to adapt it to the unknown contact environment. Thirdly, the effectiveness of the present approach is demonstrated by spring pressing experiments on the control of a 6 degrees of freedom (6 DOF) robot which grasps a rigid tool.

By adjusting the impedance parameters of the robot to adapt it to the unknown contact environment, the robot can accurately follow the given force reference, and the present impedance control system can stably work under the unknown contact environment, the nonlinear robotic dynamics and disturbances. The
experimental results also show that the present approach has high adaptiveness, stability and flexibility.

2. Model and Control of Contact Tasks

Generally, contact tasks may be classified into wiping, painting, deburring and pressing operations on a contact plane which may be modeled physically by mass-damper-spring systems. In the operations stated above, it is important to control the contact force of a robot in the normal direction of the contact plane.

Figure 1(a) shows a pressing operation by a robot on a contact plate with a spring. The pressing operation may be divided into a planning stage and an execution stage. At the planning stage, the free motion trajectory of the robot from a reference location A to a contact location C, and the contact force trajectories of the robot from the contact location C to a goal location G need to be specified and represented accurately in the workspace frame $\Sigma_o$ according to the three dimensional position/orientation vectors of the featured points on the robot and the contact plate with a spring. At the execution stage, the robot needs to be controlled precisely, so that it can accurately follow the contact force trajectories specified at the planning stage. In Fig. 1(a), the contact plate with a spring is depicted on the left-hand side for the pressing operation, so that the pressing operation can be clearly distinguished from the free motion.

3. Control Problem, Model and Law

3.1 A contact control problem

By considering the contact force and the constraint on the robot in Fig. 1(a), the motion equation of the robot with 6 DOF is given in the robot joint space by

$$\ddot{\theta}(t)+\mathbf{J}'(\theta(t))\mathbf{p}(t)+\mathbf{V}(\theta(t),\dot{\theta}(t))=\mathbf{f}(t),$$

where $\theta(t) \in \mathbb{R}^{6\times1}$ is a joint angle vector, and $t \in [0, t_f]$ is the contact time. $\mathbf{M}(\theta(t)) \in \mathbb{R}^{6\times6}$ is a symmetric and positive definite inertia matrix. $\mathbf{V}(\theta(t), \dot{\theta}(t)) \in \mathbb{R}^{6\times1}$ contains the centrifugal, Coriolis and gravity terms. $\mathbf{f}(t) \in \mathbb{R}^{6\times1}$ is a torque vector for the robot joint control. $\mathbf{J}(\theta(t)) \in \mathbb{R}^{6\times6}$ is a nonsingular Jacobian matrix. $\mathbf{p}(t) \in \mathbb{R}^{6\times1}$ is a position/orientation vector which is defined in $\Sigma_o$. Vector $\mathbf{p}(t)$ corresponds to the featured point on the tool shown in Fig. 1(a). It is known from the forward kinematics of the robot that

$$\mathbf{p}(t)=\hat{\Lambda}[\theta(t)] \quad t \in [0, t_f],$$

where $\hat{\Lambda}[\theta(t)] \in \mathbb{R}^{6\times1}$ is the forward kinematics of the robot. Combining Eq. (1) with Eq. (2) and considering the derivatives of $\mathbf{p}(t)$, Eq. (1) can be expressed in $\Sigma_o$ as

$$\ddot{\mathbf{p}}(t)+\mathbf{V}(\mathbf{p}(t), \dot{\mathbf{p}}(t))=\mathbf{f}(t) \quad t \in [0, t_f].$$

where

$$\ddot{\mathbf{p}}(t)=\mathbf{J}(\theta(t))\dot{\mathbf{p}}(t),$$

$$\mathbf{V}(\mathbf{p}(t), \dot{\mathbf{p}}(t))=\mathbf{J}(\theta(t))\mathbf{V}(\theta(t), \dot{\theta}(t)),$$

$$\mathbf{J}(\theta(t))\mathbf{J}^{-1}(\theta(t))\mathbf{f}(t),$$

$$\mathbf{f}(t)=\mathbf{J}(\theta(t))\dot{\mathbf{p}}(t),$$

If we consider the relationship between $\mathbf{f}(t)$ and the impedance parameters of the contact plate with a spring, we have

$$\mathbf{f}(t)=\phi[M_s, D_s, K_s, \mathbf{p}(t), \dot{\mathbf{p}}(t), \ddot{\mathbf{p}}(t), \mathbf{p}_s(t)] \quad t \in [0, t_f],$$

where $M_s, D_s$ and $K_s \in \mathbb{R}^{6\times6}$ are the diagonal inertia, damping and stiffness matrices of the contact plate.
with a spring, respectively, and they are assumed to be unknown. \( p_i(t) \in \mathbb{R}^{6 \times 1} \) is the position/orientation vector of the contact point on the contact plate with a spring which is in a static state. Generally, \( \phi \) can be regarded as a nonlinear vector function.

It is seen from Eqs. (3.2) to (3.5) and (4) that the control of the contact force of the robot is a highly nonlinear problem. It has been shown\(^{11}\) that impedance control is an effective method which can preserve stability of the motion and the contact force of a robot. Therefore, the impedance control is adopted in this paper to control the contact force of the robot in contacting tasks.

3.2 An impedance control model

The impedance control can be defined by\(^{23,92)12}\)

\[
M \dot{\mathbf{p}}(t) - \mathbf{K}(\mathbf{p}(t) - \mathbf{p}_e(t)) + D \dot{\mathbf{p}}(t) = \mathbf{f}(t) \quad t \in [0, t_1],
\]

where \( M, D \) and \( K \in \mathbb{R}^{6 \times 6} \) are symmetric and positive definite matrices which specify the inertia, damping and stiffness of the robot, respectively. \( \mathbf{f}(t) \in \mathbb{R}^{6 \times 1} \) is a force/torque vector defined in Eq. (1). \( \mathbf{p}(t), \dot{\mathbf{p}}(t) \) and \( \mathbf{p}_e(t) \in \mathbb{R}^{6 \times 1} \) are a reference acceleration vector, a reference velocity vector and a reference position vector of \( \mathbf{p}(t) \), respectively, and they are defined in \( \Sigma_o, \mathbf{f}(t) \in \mathbb{R}^{6 \times 1} \) is a reference force/torque vector of \( \mathbf{f}(t) \).

According to Hogan's impedance control theory\(^{41}\), the impedance control should adjust the mechanical impedance of the robot to make \( \mathbf{f}(t), \mathbf{p}(t), \dot{\mathbf{p}}(t) \) and \( \ddot{\mathbf{p}}(t) \) satisfy the behavior specified by Eq. (5). Therefore, the objective of this paper is to adjust \( M, D \) and \( K \) in the unknown contact environment in order to converge the impedance error to a minimum. The impedance error \( E(t) \) is defined by

\[
E(t) = \mathbf{e}(t)^T W(t) \mathbf{e}(t),
\]

\[
\mathbf{e}(t) = \mathbf{p}_e(t) - \mathbf{p}(t) + \mathbf{M}^{-1} \mathbf{f}(t) - \mathbf{f}(t) - \mathbf{M}^{-1} \dot{\mathbf{p}}(t) + \mathbf{M}^{-1} \mathbf{D}[\mathbf{p}_e(t) - \mathbf{p}(t)]
\]

\[
\in \mathbb{R}^{6 \times 1}, \quad t \in [0, t_1],
\]

where \( W \in \mathbb{R}^{6 \times 6} \) is a diagonal weighting matrix.

3.3 An impedance control law

Let us define

\[
\Delta \mathbf{p}(k+1) = \mathbf{p}(k+1) - \mathbf{p}(k), \quad \Delta \mathbf{f}(k+1) = \mathbf{f}(k+1) - \mathbf{f}(k).
\]

Combining Eq. (7) with Eq. (5) yields

\[
M \Delta \dot{\mathbf{p}}(k+1) + D \Delta \dot{\mathbf{p}}(k+1) + K \Delta \mathbf{p}(k+1) = \Delta \mathbf{f}(k+1)
\]

\[
te \in [0, t_1].
\]

When the sampling period for the robot joint control \( T \) is very minute, it is reasonable to substitute \( \Delta \mathbf{p}(k+1) = (\Delta \mathbf{p}(k+1) - \Delta \mathbf{p}(k))/T \) for \( \Delta \mathbf{p}(k) \) in Eq. (8) at time \( t = kT \). Therefore, we get the discrete-time expression of Eq. (8) as

\[
\Delta \mathbf{p}(k+1) = \Delta \mathbf{p}(k) + T \mathbf{M}^{-1} [\Delta \mathbf{f}(k) - D \Delta \mathbf{p}(k) - K \Delta \mathbf{p}(k)]
\]

\[
\Delta \mathbf{f}(k+1) = \Delta \mathbf{f}(k).
\]

(9)

where \( N_r \) is an integer given by \( N_r = t_r / T \).

Defining

\[
\dot{\mathbf{p}}(k+1) = \mathbf{p}(k+1), \quad K_M = M^{-1}, \quad K_M = M^{-1} D \quad \text{and}
\]

\[
K_{\text{ex}} = M^{-1} K,
\]

we obtain from Eq. (9)

\[
\dot{\mathbf{p}}(k+1) = T \dot{\mathbf{p}}(k) + \dot{\mathbf{f}}(k) + T [K_M \Delta \mathbf{p}(k) + K_{\text{ex}} \Delta \mathbf{f}(k)]
\]

\[
t = 0, 1, \ldots, N_r - 1.
\]

(10)

Equation (10) can be regarded as a discrete-time impedance control law.

Furthermore, if the reference inputs for the robot joint control are joint angles, \( \dot{\mathbf{p}}(k+1) \) does not match with the robot joint angles. By computing the derivative of the forward kinematics defined by Eq. (2) and substituting \( \dot{\mathbf{p}}(k) \) for \( \dot{\mathbf{p}}(k) \), we have

\[
\dot{\mathbf{p}}(k+1) = J(\dot{\theta}(k)) \dot{\theta}(k)
\]

\[
t = 0, 1, \ldots, N_r - 1.
\]

(11)

where \( J(\dot{\theta}(k)) \in \mathbb{R}^{6 \times 6} \) is a reference angular velocity vector for the robot joint control at time \( t = kT \). \( J(\dot{\theta}(k)) \in \mathbb{R}^{6 \times 6} \) is the Jacobian matrix at time \( t = kT \) defined in Eq. (3). By substituting \( \dot{\theta}(k) = \dot{\theta}(k)(k+1) - \theta(0)(k) / T \) for \( \dot{\theta}(k) \) in Eq. (11), we get

\[
\dot{\mathbf{p}}(k+1) = \dot{\mathbf{p}}(k) + T J^{-1}(\dot{\theta}(k)) \dot{\mathbf{f}}(k).
\]

(12)

\[
\theta(0)(k) = \theta(0) \quad (k = 0, 1, \ldots, N_r - 1).
\]

4. Reference Inputs for the Impedance Control

It is seen from Eq. (10) that \( \dot{\mathbf{p}}(k), \dot{\mathbf{p}}(k) \) and \( \dot{\mathbf{p}}(k) \) need to be specified. For the free motion, \( \dot{\mathbf{p}}(k) \) can be set to zero, because \( \dot{\mathbf{p}}(k), \dot{\mathbf{p}}(k) \) and \( \dot{\mathbf{p}}(k) \) have no concern with \( \dot{\mathbf{f}}(k) \). For the contact motion, \( \dot{\mathbf{p}}(k) \) is determined by the contact task, moreover, \( \dot{\mathbf{p}}(k), \dot{\mathbf{p}}(k) \) and \( \dot{\mathbf{p}}(k) \) are dependent on \( \dot{\mathbf{f}}(k) \).

4.1 Reference inputs for free motion

It is assumed in Fig. 1(b) that the position vectors \( \dot{\mathbf{p}}(k) \in \mathbb{R}^{6 \times 1} \) are known, and that the position vector \( \dot{\mathbf{p}}_C(\in \mathbb{R}^{6 \times 1}) \) corresponds to the contact point \( C \) on the contact plane with a spring shown in Fig. 1(a). When the rigid tool contacts the contact point \( C \), a measured value can be detected by the force/torque sensor installed on the gripper. Therefore, \( \dot{\mathbf{p}}_C \) can be conveniently determined by detecting whether the force/torque sensor has a value and computing the forward kinematics of the robot.

When the robot moves from the reference location \( A \) to the contact location \( C \), the position vectors
where \( p_{k+1} \in \mathbb{R}^{3+i} \) (\( i=1,2,3,4 \)) can be computed using \( q_p \) and \( \omega \) and the relation \( q_p = \omega p \). By the equivalent angle axis method, the orientation vectors \( q_{pA} \) and \( q_{pc} \) are computed with respect to \( \sum \) and to \( \Sigma \) can be computed by \( q_{pA} \) and \( q_{pc} \) for \( i=1,2,3,4 \), respectively. Therefore, we can define position/orientation vectors \( p_{m} = \{ q_{pA}, q_{pc} \} \in \mathbb{R}^{3+i} \) and \( p_{m} = \{ p_{ic}, \omega c \} \in \mathbb{R}^{3+i} \).

If we let \( \hat{p}(k) \) be defined in Eq. (5) move from \( p_{m} \) to \( p_{ic} \) through a 5 order polynomials trajectory \( p_{ic}(k) \in \mathbb{R}^{3+i} \) with six constraints:

\[
\begin{align*}
\hat{p}_{x}(0) &= 0, \quad \hat{p}_{x}(1) = 0, \quad \hat{p}_{x}(2) = 0, \quad \hat{p}_{x}(3) = 0, \quad \hat{p}_{x}(4) = 0, \quad \hat{p}_{x}(5) = 0, \\
\hat{p}_{y}(0) &= 0, \quad \hat{p}_{y}(1) = 0, \quad \hat{p}_{y}(2) = 0, \quad \hat{p}_{y}(3) = 0, \quad \hat{p}_{y}(4) = 0, \quad \hat{p}_{y}(5) = 0, \\
\hat{p}_{z}(0) &= 0, \quad \hat{p}_{z}(1) = 0, \quad \hat{p}_{z}(2) = 0, \quad \hat{p}_{z}(3) = 0, \quad \hat{p}_{z}(4) = 0, \quad \hat{p}_{z}(5) = 0.
\end{align*}
\]

(13.a)

we obtain the reference input for the free motion

\[
\begin{align*}
p_{ic}(k) &= a_{0} + a_{1}k + a_{2}k^2 + a_{3}k^3 + a_{4}k^4 + a_{5}k^5 \\
&= \begin{cases} \\
0 & k = 0 \\
1 & k = 1 \\
2 & k = 2 \\
3 & k = 3 \\
4 & k = 4 \\
5 & k = 5
\end{cases}, \\
&= \begin{cases} \\
0 & k = 0 \\
1 & k = 1 \\
2 & k = 2 \\
3 & k = 3 \\
4 & k = 4 \\
5 & k = 5
\end{cases}.
\end{align*}
\]

(13.b)

where \( t_{i} \) is free motion time, \( N_{i} \) is an integer given by \( N_{i} = t_{i}/T \), and \( a_{i} \in \mathbb{R}^{3+i} \) for \( i = 0, 1, 2, ..., 5 \) are real number vectors given by

\[
\begin{align*}
a_{0} &= p_{mA}, \quad a_{1} = 0, \quad a_{2} = 0, \quad a_{3} = \frac{10}{t_{1}^3}(p_{ic} - p_{mA}), \\
a_{4} &= -\frac{15}{t_{1}^4}(p_{ic} - p_{mA}), \quad a_{5} = \frac{6}{t_{1}^5}(p_{ic} - p_{mA}).
\end{align*}
\]

(13.c)

4.2 Reference inputs for contact motion

In Fig. 1 (c), let \( \theta_{pA} \in \mathbb{R}^{3+i} \) be a position vector, and \( \theta_{pA} \in \mathbb{R}^{3+i} \) be an orientation vector. At time \( t = kT \), we define a position/orientation vector \( \theta_{pA}(k) = \{ \theta_{pA}, \omega_{A} \} \in \mathbb{R}^{3+i} \) which corresponds to the position and orientation of the robot at a goal position \( G \). If \( \theta_{A}(k) \) is exerted on the robot at the goal position \( G \), \( \theta_{pA}(k) \) can be specified by the motion equation

\[
\begin{align*}
M \ddot{p}_{ic}(k) + D \dot{p}_{ic}(k) + K[p_{ic}(k) - p_{ic}] &= \theta_{A}(k) \\
p_{ic}(N_{i+1}) &= p_{ic}(N_{i}) \quad (N_{i} \leq k \leq N_{i+1} - 1)
\end{align*}
\]

(14)

where \( M \), \( D \), and \( K \) are defined in Eq. (5), and they are empirically determined. \( t_{r} \) is contact motion time, and \( N_{i} \) is an integer given by \( N_{i} = t_{r}/T \). By solving \( p_{ic}(k) \) from Eq. (14) and letting \( p_{ic}(k) = \theta_{A}(k) \), we obtain the reference input \( p_{ic}(k) \) for the contact motion.

For the free motion and the contact motion, \( \dot{p}_{ic}(k) \) and \( \ddot{p}_{ic}(k) \) can be obtained by computing \( \dot{p}_{ic}(k) = \{ \dot{p}_{ic}(k+1) - \dot{p}_{ic}(k) \}/T \) and \( \ddot{p}_{ic}(k) = \{ \ddot{p}_{ic}(k+1) - \ddot{p}_{ic}(k) \}/T \).

5. Learning Impedance Control

5.1 Learning impedance controllers

The impedance control law defined by Eq. (10) requires \( K_{u}, K_{md} \) and \( K_{mk} \). If we use the empirical \( M \), \( D \), and \( K \) (which can be calculated from \( K_{u}, K_{md} \) and \( K_{mk} \) specified in Eq. (14)) during the contact control, it is difficult to make \( f(k), \dot{p}(t), \ddot{p}(t) \) and \( \dddot{p}(t) \) satisfy the behavior specified by Eq. (5) due to the unknown \( M_{s}, D_{s} \) and \( K_{s} \) defined in Eq. (4). Therefore, it is desirable that \( K_{u}, K_{md} \) and \( K_{mk} \) affect \( f(k), \dot{p}(t), \ddot{p}(t) \) and \( \dddot{p}(t) \). Moreover, \( f(k), \dot{p}(t), \ddot{p}(t) \) and \( \dddot{p}(t) \) also affect \( f(k), \dot{p}(t), \ddot{p}(t) \) and \( \dddot{p}(t) \). Therefore, the adjustment of \( K_{u}, K_{md} \) and \( K_{mk} \) has nonlinear relations with \( f(k), \dot{p}(t), \ddot{p}(t), \dddot{p}(t) \).

Three similar multilayer neural networks \( NN_{i}, NN_{e} \) and \( NN_{d} \) shown in Fig. 2 are introduced to learn on-line the above nonlinear relations. In Fig. 2, the signals flow from left to right. Let \( M, P, U \) and \( N \) be the number of the neurons of the input layer A, hidden layer B, C and output layer D, respectively. For \( g = 1, 2, ..., M \), \( I = 1, 2, ..., P \), \( f = 1, 2, ..., N \) and \( n = 1, 2, ..., N \), the neural networks \( NN_{i}, NN_{e} \) and \( NN_{d} \) can be represented by the following relations:

\[
\begin{align*}
x_{i} &= \sum_{j=1}^{M} w_{ij}^{mN} \cdot y_{j}, \quad x_{i} = \sum_{j=1}^{P} w_{ij}^{mN} \cdot y_{j}, \quad x_{i} = \sum_{j=1}^{N} w_{ij}^{mN} \cdot y_{j}, \\
y_{j} &= x_{i'}, \quad y_{j'} = f(x_{i'}), \quad y_{j'} = f(x_{i'}), \quad y_{j'} = f(x_{i'}), \\
(15.a)
\end{align*}
\]

where \( x_{i} \) and \( y_{j} \) are the input and output of \( m \)th neuron of the layer \( R = \text{g}, l, j, i ; R = \text{A}, \text{B}, \text{C}, \text{D} \), respectively. \( w_{ij}^{mN}, w_{ij}^{mN} \) and \( w_{ij}^{mN} \) are the weights between \( y_{j} \) and \( x_{i} \), \( y_{j'} \) and \( x_{i} \), \( y_{j'} \) and \( x_{i} \), respectively. The sigmoid function \( f(x) \) is given by

\[
f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}.
\]

(15.b)

Figure 3(a) shows a block diagram of the present impedance control system. In Fig. 3(a), \( I \) is a 6 \( \times \) 6 unity matrix, \( f(k) \) is given in advance, and the off-line computation of \( \hat{p}(k), \ddot{p}(k) \) and \( \dddot{p}(k) \) is shown in Fig. 3(b). The input vectors of \( NN_{i}, NN_{e} \) and \( NN_{d} \) are determined by the nonlinear relations mentioned above. When the robot contacts the contact plate with a spring, \( NN_{i}, NN_{e} \) and \( NN_{d} \) begin to learn and to change their weights and biases. The diagonal matrices \( \Delta K_{u}, \Delta K_{md} \) and \( \Delta K_{mk} \in \mathbb{R}^{3+i} \) are outputs of

---

**Fig. 2** Neural network structure \((NN_{i}, NN_{e} \text{ and } NN_{d})\)
NNr, NNc and NNp, respectively. Their diagonal elements are used to adjust the diagonal elements of the matrices \( K_{Kr}, K_{Kn} \) and \( K_{Kn} \), respectively.

According to Fig. 3(a), Eq.(10) can be rewritten by

\[
\hat{p}_v(k+1) = T\hat{p}_v(k) + \hat{p}_v(k) + T\left[ u_{sd}(k) + u_{sd}(k) \right] \quad (N_1 \leq k \leq N_1 + N_2 - 1),
\]

where \( u_{sd}(k) = K_{sd}J\hat{p}_v(k), \quad K_{sd} = K_{sd} + U_s \Delta K_{sd}, \)
\[ u_{sd}(k) = K_{sd}J\hat{p}_v(k), \quad K_{sd} = K_{sd} + U_s \Delta K_{sd}, \]
\[ u_{sd}(k) = K_{sd}J\hat{p}_v(k), \quad K_{sd} = K_{sd} + U_s \Delta K_{sd}. \]

and \( U_s, U_c \) and \( U_r \) are scalar constants. The initial values \( K_{sd}, K_{sn} \) and \( K_{sn} \) for \( K_{Kr}, K_{Kn} \) and \( K_{Kn} \) are empirically determined. After sufficient learning, the obtained matrices \( K_{Kr}, K_{Kn} \) and \( K_{Kn} \) are regarded as the desirable impedance parameters for the impedance control of the robot.

5.2 On-line learning algorithms

In this paper, \( \delta \) rule-based learning algorithms are used to train the weights and the biases of \( NN_r, NN_c \) and \( NN_p \). We use the impedance error \( E(t) \) defined in Eq.(6) as an evaluation function for the learning algorithms. At time \( t = kT \), we can get the discrete time expression of Eq.(6) as

\[
E(k) = e(k)W_e(k), \quad e(k) = \Delta \hat{p} - \hat{p}_v(k) - u_{sd}(k), \quad (N_1 \leq k \leq N_1 + N_2 - 1),
\]

where \( \Delta \hat{p}(k) = \hat{p}_v(k) - \hat{p}(k) \), and \( \hat{p}(k) = \hat{p}(k) - \hat{p}(k-1)/T \).

Using Eq.(17), \( K_{Kr}, K_{Kn} \) and \( K_{Kn} \) can be adjusted by on-line learning of \( NN_r, NN_c \) and \( NN_p \).

According to the \( \delta \) rule, let \( w_{sp}^{ij} \) in \( NN_p \) change as

\[
\Delta w_{sp}^{ij} = -\eta_p \frac{\partial E}{\partial w_{sp}^{ij}} \quad (i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, U),
\]

where \( \Delta w_{sp}^{ij} \) is the increment of \( w_{sp}^{ij} \). \( \eta_p \) is the learning rate of \( NN_p \). Similarly, we have \( \eta_c \) for \( NN_c \) and \( NN_p \), respectively.

Let \( e_i, u_{sd}(k), \hat{p}_v(k), p(k) \) and \( \Delta K_{sd} \) be the \( i \)th elements of \( e(k), u_{sd}(k), \hat{p}_v(k), p(k) \) and \( \Delta K_{sd} \), respectively, and \( w_{sp} \) be the \( i \)th diagonal element of \( W \).
Table 1: Diagonal ith element of \( \mathbf{K}_p, \mathbf{K}_f \) and \( \mathbf{K}_o \)

<table>
<thead>
<tr>
<th>( i )</th>
<th>( \mathbf{K}_p ) V/( \text{rad} )</th>
<th>( \mathbf{K}_f ) V/( \text{rad} )</th>
<th>( \mathbf{K}_o ) V/( \text{rad} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25 473</td>
<td>0.000 039</td>
<td>0.023 5</td>
</tr>
<tr>
<td>2</td>
<td>8 748</td>
<td>0.000 114</td>
<td>0.029 6</td>
</tr>
<tr>
<td>3</td>
<td>11 759</td>
<td>0.000 085</td>
<td>0.023 5</td>
</tr>
<tr>
<td>4</td>
<td>228</td>
<td>0.004 380</td>
<td>0.015 7</td>
</tr>
<tr>
<td>5</td>
<td>2 664</td>
<td>0.000 250</td>
<td>0.011 2</td>
</tr>
<tr>
<td>6</td>
<td>795</td>
<td>0.001 260</td>
<td>0.010 7</td>
</tr>
</tbody>
</table>

We can obtain from Eqs. (17) and (16) \( c \)

\[
\frac{\partial E}{\partial \mathbf{x}^p} = \frac{\partial E}{\partial \mathbf{c}_i} + \frac{\partial E}{\partial \mathbf{H}_\text{MKI}} \frac{\partial \mathbf{H}_\text{MKI}}{\partial \mathbf{x}^p} + \frac{\partial E}{\partial \mathbf{y}^p} \frac{\partial \mathbf{y}^p}{\partial \mathbf{x}^p} + \frac{\partial E}{\partial \mathbf{x}^p} = \mathbf{0}, \quad i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, U, \quad (19)
\]

where

\[
\frac{\partial E}{\partial \mathbf{c}_i} = 2 w_{i,j}, \quad \frac{\partial E}{\partial \mathbf{H}_\text{MKI}} = 1, \quad \frac{\partial E}{\partial \mathbf{y}^p} = f'(x^p), \quad \frac{\partial \mathbf{y}^p}{\partial \mathbf{x}^p} = y_f.
\]

The weights \( \mathbf{w}_i^p \) and \( \mathbf{w}_j^p \) in \( \mathbf{NN}_p \) are computed by the back-propagation algorithm, and the detailed description is omitted. The same learning algorithm is adopted to compute the weights of \( \mathbf{NN}_e \) and \( \mathbf{NN}_v \).

In Fig. 3(a), the PID controller \( \mathbf{G}_e(z) \) is given by

\[
\mathbf{G}_e(z) = \mathbf{K}_p + \frac{\mathbf{K}_f}{1 - z^{-1}} + \mathbf{K}_o(1 - z^{-1}) \quad (20)
\]

where \( \mathbf{K}_p, \mathbf{K}_f \), and \( \mathbf{K}_o \) are diagonal matrices which are empirically determined, and their diagonal elements are given in Table 1.

Table 2: Parameters used in the experiments

<table>
<thead>
<tr>
<th>( \mathbf{NN}_f )</th>
<th>( \mathbf{NN}_e )</th>
<th>( \mathbf{NN}_p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M=36 )</td>
<td>( M=36 )</td>
<td>( M=36 )</td>
</tr>
<tr>
<td>( p=75 )</td>
<td>( p=50 )</td>
<td>( p=50 )</td>
</tr>
<tr>
<td>( U=45 )</td>
<td>( U=40 )</td>
<td>( U=40 )</td>
</tr>
<tr>
<td>( N=6 )</td>
<td>( N=6 )</td>
<td>( N=6 )</td>
</tr>
<tr>
<td>( U_f=0.055 )</td>
<td>( U_c=0.1 )</td>
<td>( U_p=2.5 )</td>
</tr>
<tr>
<td>( \eta_f=0.9 )</td>
<td>( \eta_c=0.9 )</td>
<td>( \eta_p=0.9 )</td>
</tr>
<tr>
<td>( T=7 \text{ ms} )</td>
<td>( t_{f1}=5 \text{ s} )</td>
<td>( t_{f2}=5 \text{ s} )</td>
</tr>
</tbody>
</table>

6. Experiments

6.1 Experimental setup and parameters

To evaluate the validity of the proposed approach, experiments are carried out using a 6 DOF robot, Motoman K3S (Yasukawa, Co.). Figure 4 gives the experimental setup. A PC-9821Xv20 personal computer is used as the robot joint controller. The gripper with 2 parallel fingers (mu RH707 Gripper, Takano Co.) grasps the rigid tool to perform the pressing operation shown in Fig. 1(a). A force/torque sensor (F/T 5/50 HSS, BI-Autotec Co.) fixed between the 6th joint of the robot and the gripper measures the contact force/torque of the robot. The parameters used in the experiments are given in Table 2, and the weighting matrix \( \mathbf{W} \) is set to be an unity matrix \( \mathbf{I} \). The initial values for \( \mathbf{K}_s, \mathbf{K}_m, \mathbf{K}_{	ext{MK}} \) and \( \mathbf{K}_{	ext{MK}} \) are empirically determined as \( \mathbf{K}_s = \text{diag}(0.1 \text{N}^{-1} \cdot \text{m/s}^{-1}, 0.1 \text{N}^{-1} \cdot \text{m/s}^{2}, 0.1 \text{N}^{-1} \cdot \text{rad/s}^{-1}, 0.1 \text{N}^{-1} \cdot \text{rad/s}^{2}, 1 \text{N}^{-1} \cdot \text{rad} / \text{s}^{-1}, 1 \text{N}^{-1} \cdot \text{rad} / \text{s}^{2}) \), \( \mathbf{K}_m = \text{diag}(100 100 100 100 100 100) \text{s}^{-1}, \mathbf{K}_{	ext{MK}} = \text{diag}(75 75 75 75 75 75) \text{s}^{-2} \), respectively.

In the experiments shown in Fig. 1(a), the robot moves from the reference location \( A \) to the contact location \( C \), and presses the contact plate with a spring from the contact location \( C \) to the goal location \( G \). This process is regarded as one trial.

6.2 Experimental results and analyses

Figure 5 shows the reference inputs \( f_{	ext{ref}}(t), p_{	ext{ref}}(t) \) and the responses \( f_{	ext{d}}(t), p_{	ext{d}}(t) \) which are the elements in the \( z \) axis direction of \( f_{	ext{d}}(t), p_{	ext{d}}(t) \), and \( p(t) \) for the contact motion when \( f_{	ext{ref}}(t)=5 \text{ N} \) and \( t_{f1}=5 \text{ s} \).

Firstly, \( \mathbf{K}_s, \mathbf{K}_m, \mathbf{K}_{	ext{MK}} \) and \( \mathbf{K}_{	ext{MK}} \) are used to compute \( \mathbf{p}_e(k) \) from \( f_{	ext{d}}(k) \) by Eq. (14) for the impedance control. After the five trials using \( \mathbf{K}_s, \mathbf{K}_m, \mathbf{K}_{	ext{MK}} \) as the initial values, the new \( \mathbf{K}_s, \mathbf{K}_m, \mathbf{K}_{	ext{MK}} \) are obtained. Then, the obtained \( \mathbf{K}_s, \mathbf{K}_m, \mathbf{K}_{	ext{MK}} \) are used to compute \( \mathbf{p}_e(k), \mathbf{p}_d(k), \mathbf{p}_{	ext{d}}(k) \) from \( f_{	ext{d}}(k) \) using Fig. 3(b) for the impedance control. Secondly, using the renewed \( \mathbf{K}_s, \mathbf{K}_m, \mathbf{K}_{	ext{MK}} \) and \( \mathbf{K}_{	ext{MK}} \) as the initial values, the new \( \mathbf{K}_s, \mathbf{K}_m, \mathbf{K}_{	ext{MK}} \) are obtained after another five trials, by which \( \mathbf{p}_e(k), \mathbf{p}_d(k), \mathbf{p}_{	ext{d}}(k) \) can be computed.
Fig. 5  Time responses for step force input

Fig. 6  Time responses for sinusoidal force input

from $f_d(k)$ using Fig. 3(b) for the impedance control.

Figures 5(a)–(f) show the responses after the 1st trial, 10th trial and 20th trial. It is shown in Figs. 5(c)–(f) that after twenty trials (total 14,300 iteration), the error between $f_d(t)$ and $f_{ad}(t)$ is approximately equal to 0, and $p_d(t)$ can accurately follow $p_{ad}(t)$, and the diagonal elements of the $K_M$, $K_{MD}$ and $K_{MK}$ are obtained and are given in Table 3. In Fig. 5(g), the learning error $E$ is given by

$$E = \frac{1}{N_{tr}} \sum_{k=1}^{N_{tr}} E(k),$$

where $N_{tr}=715$.

When $f_{ad}(t)=5\sin(0.6t)\ N$ and $t_{f}=5\ s$, the pressing experiments are carried out again, and $p_d(k)$, $\dot{p}_d(k)$ are renewed by the above method every five trials. Figures 6(a)–(f) show the responses after the 1st trial, 10th trial and 20th trial. After twenty trials (total 14,300 iteration), the error between $f_d(t)$ and $f_{ad}(t)$ is approximately equal to 0, $p_d(t)$ can accurately follow $p_{ad}(t)$, and the third diagonal elements of the obtained $K_M$, $K_{MD}$ and $K_{MK}$ are approximately equal to the values shown in Table 3.

In the experiments, the third diagonal element of $M_S$ contains the inertia of the gripper, the rigid tool and the contact plate, and the third diagonal element of $K_S$ includes the stiffness of the spring. Comparing the diagonal elements of $K''_M$, $K''_{MD}$ and $K''_{MK}$ and those of the $K_M$, $K_{MD}$, $K_{MK}$ in Table 3, it can be seen that $NN_y$, $NN_x$, $NN_\phi$ can adjust the third diagonal elements of the impedance parameters $K_M$, $K_{MD}$, $K_{MK}$ and that the third diagonal elements of the $K_M$, $K_{MK}$ are approxi-
Table 3 Diagonal $i$th element of $K_M$, $K_{MD}$ and $K_{MK}$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$K_M$ (N·m/s)</th>
<th>$K_{MD}$ (s$^{-1}$)</th>
<th>$K_{MK}$ (s$^{-2}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>100</td>
<td>75</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>100</td>
<td>75</td>
</tr>
<tr>
<td>3</td>
<td>0.1075</td>
<td>104.75</td>
<td>77.28</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>100</td>
<td>75</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>100</td>
<td>75</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>100</td>
<td>75</td>
</tr>
</tbody>
</table>

approximately equal to those of $M^{-1}, M^{-1}K_M$. The measurements of $M, K_M$ also confirm the above fact.

7. Conclusions

The following conclusions have been obtained:

(1) The present evaluation function and the online learning algorithms can adjust the impedance parameters of a robot to match the impedance parameters of the unknown contact environment.

(2) By adjusting the impedance parameters of the robot to adapt it to the unknown contact environment, the robot can accurately follow the given force reference, and the present impedance control system can stably work.

(3) The proposed approach is used to control both the contact motion and the free motion of the robot, it is unnecessary to switch between the control modes as the motion conditions change. Therefore, the proposed approach has high flexibility.

(4) The experimental results show that the learning processes of the neural networks are convergent by using the present evaluation function and learning algorithms.

References


