Development of Neural Network-based Explicit Force Control with Disturbance Observer

Toshiaki Okano*✉, Takahiro Nozaki**,***, Senior Member
Toshiyuki Murakami**, Fellow

(Manuscript received May 12, 2020, revised Oct. 19, 2020)
J-STAGE Advance published date: Dec. 4, 2020

Human support robots are in high demand in certain countries owing to the circumstances of declining working population due to the lower birth rate and the occurrence of an aged society; moreover, these robots are expected to operate in open environments where they interact with other objects. The development of force control is expected in such an environment, and the performance enhancement of force control has been extensively studied. In particular, this study discusses force control with a disturbance observer and applies a neural network (NN) into its controller; the NN works as both the feedback and feedforward components. The contribution of this study is to show the development method of force control using disturbance observer and a NN, which enhances the performance of force control from the perspective of both feedback and feedforward components. The structure of the controller and composition of the NN were selected through simulation results; moreover, the compensator based on NN was designed in a frequency range higher than the cutoff frequency of the observer with a small number of hidden layers. Moreover, this study discusses a training method of weights in real time. Simulations and experiments were performed for showing the effectiveness of the proposal.

Keywords: disturbance observer, force control, neural network

1. Introduction

Human support robots are in high demand in certain countries owing to the circumstances of declining working population due to the lower birth rate and the occurrence of an aged society; moreover, these robots are expected to operate in open environments where they interact with other objects. The development of force control is expected in such an environment, and the performance enhancement of force control has been extensively studied. In particular, this study considers explicit force control with a disturbance observer (DOB) and reaction force observer (RFOB) instead of as a DOB in the acceleration dimension. The DOB and RFOB can enhance the robustness and estimate the external force without using force sensors.

A proportional-integral-derivative (PID) controller is often utilized for controlling certain physical variables; moreover, a PID controller can be applied for explicit force control. Certain studies estimated the derivative of force, which is jerk, and a proportional-derivative gain was applied by designing the state estimator. The derivative gain improves the performance in the high-frequency range; however, the information in the dimension of the jerk is sometimes difficult to estimate and control owing to the noise effect. The DOB is designed in the dimension of force as a force-based DOB (FDOB) instead of as a DOB in the acceleration dimension or based on integral gain. The transient response was improved because the inertial force is also compensated in addition to other disturbances; however, the system easily becomes unstable depending on the environmental impedance. Therefore, a damper or velocity feedback is often combined to stabilize the system in the FDOB; however, the transient response is degraded. First- or high-order sliding mode control was combined with force control to enhance the performance while guaranteeing stability. However, chattering or a complicated parameter design is sometimes a big obstacle in implementing sliding mode control into force control. The explicit force control with DOB and RFOB has been extensively studied; however, the design of a controller and selection of appropriate gains are sometimes challenging because they are affected by the condition of motion. For instance, the appropriate gains can easily vary according to the environmental impedance. At the same time, certain disturbances such as model uncertainties or friction also affect the performance, and they easily vary according to the motion. Therefore, both feedback and feedforward components should be considered to enhance the performance of force control, which is in contrast to the methods used in conventional studies.

The target of this study is to combine a neural network (NN), which has both feedback and feedforward components, into the explicit force control composed of proportional-integral (PI) gain and DOB for enhancing the performance of force control. The implementation of an NN into motor
control operations, such as position control, hybrid position and force control, and force control, has been extensively studied, and the parameters of the compensator are iteratively updated based on the error of the control variable. For instance, a system of two feedforward neural networks was proposed to learn the nonlinearities of the position-controlled flexible arm associated with an inverse dynamics controller; moreover, an approach for a two-link flexible manipulator was also discussed. Hybrid position and force control was developed using feedback error learning (FEL). Both the position and force controllers in the subspace were constructed using NNs and each controller learned based on the free and contact motions. The NN or FEL was also applied to force control. Fuzzy logic and NN compensators were compared in a milling control operation, in which the cutting force was in the order of hundreds. Nonlinear viscoelastic environmental conditions were considered as a part of the system and approximated using an NN feedforward term and the interactive motion was considered based on its approximated model.

NNs were also developed to compensate for model uncertainties or to be added as an additional gain in these studies. The DOB is also known as a strong tool to compensate for model uncertainties of the system. To enhance the performance of force control, a combination method of DOB and NN must be discussed by avoiding the interference between the works of these two methods.

This study proposed a combination of NN and explicit force control with DOB. The structure of the controller and composition of the NN were selected through simulation results, and the compensator based on NN was designed to operate in a frequency range higher than the cutoff frequency of the observer with a small number of hidden layers. Moreover, this study discusses a training method of weights in real time. Simulations and experiments were performed in the study for showing the effectiveness of the proposal.

The rest of this study consists of five sections. In Section 2, the explicit force control is briefly discussed. In Section 3, the proposed NN-based explicit force control is discussed. In Section 4, the simulation is explained. In Section 5, the outline and results of the experiments are discussed. This study is concluded in Section 6.

2. Explicit Force Control

This study considers explicit force control with DOB and RFOB, and its block diagram is shown in Fig. 1. The environmental impedance is computed as follows:

\[ f^{\text{ext}} = Z(s)x^{\text{res}} = (sd_c + k_c)x^{\text{res}}. \]  

The nomenclature used in this study is listed in Table 1.

The transfer function of its control is computed as follows, where the derivation of its transfer function is explained in the appendix:

\[ f^{\text{cmd}} = \frac{L(s)}{1 + L(s)}, \]  

\[ L(s) = C_f(s) \frac{M_n G_f(s)}{K_n G_f(s)} \]  

\[ G_f(s) = K_n G_f(s) Z(s) - [1 - G_f(s)] g_f K_n M_{f0} G_f(s), \]

\[ G_f(s) = \frac{K_n}{K_{n,f}} \]  

\[ Z(s) = M s^2 + Z_c(s). \]

The position response is measured using an encoder. Derivative control is often implemented in the controller, but its dimension becomes a jerk in the case of force control and is heavily affected by noise. Therefore, we considered the PI controller instead of the PID controller in \( C_f(s) \). The first-order low-pass filter is utilized as \( \frac{1}{s + g_p} \) in the DOB and RFOB, respectively. The velocity response is estimated through the pseudo derivative of the position response by \( G_v(s) \).

The block diagram of operations from the occurrence of external and disturbance forces to the force response is shown in Fig. 2 and computed using the following equation:

\[ \frac{M s^2 x^{\text{res}}}{f^{\text{ext}} + f^{\text{dist}}} = -\frac{s^2}{s^2 + g_d G_v(s) \frac{K_n}{K_{n,f} M_{f0}}}. \]  

If the nominal parameters are properly identified and the pseudo derivative can be treated as a simple derivative, then Eq. (3) can be recomputed as follows:

\[ \frac{M s^2 x^{\text{res}}}{f^{\text{ext}} + f^{\text{dist}}} = -\frac{s}{s + g_d}. \]

Eqs. (3) and (4) show the filtering effect by DOB.
The motion equation of explicit force control with DOB is computed as follows:

\[ M_s^2 x_{\text{res}} = C_f(s) \frac{M_n}{K_{\text{nn}}} f_{\text{cmd}} - f_{\text{ext}} - \left[1 - G_d(s)](f_{\text{ext}} + f_{\text{dis}}) \right]. \]

Here, \( G_d(s) \) is equivalent to Eq. (3) or (4). The estimated external force is computed as follows:

\[ f_{\text{ext}} = f_{\text{cmd}} - \left[1 - G_d(s)] \frac{K_{\text{nn}}}{C_f(s) M_n} (f_{\text{ext}} + f_{\text{dis}}) \right]. \]

As the gain of the force controller becomes larger, the second and third terms on the right-hand side of Eq. (6) become smaller. Therefore, an appropriate control gain setting or a combination of another controller can improve the performance of its control.

The Bode diagram of explicit force control and the filtering effect by DOB with the parameters listed in Table 2 are shown in Fig. 3. DOB ideally compensates for disturbances that are lower than its cutoff frequency; however, the parameter setup of the nominal values or nonlinear terms affects the performance. Moreover, the inertial force is not compensated because the DOB is designed in the dimension of acceleration, and the transient response of explicit force control with DOB must be improved. This study aimed to improve the performance of force control from the perspective of both feedback and feedforward components without significantly limiting the condition of motion.

3. Proposed Neural Network (NN)-based Explicit Force Control with Disturbance Observer

3.1 Outline  The NN is implemented into the explicit
Table 3. Candidate structures

|                | DOB        | NN (all frequencies) | NN (high frequency) |
|----------------|------------|----------------------|---------------------|
| Conventional   | ✓          | –                    | –                   |
| Structure 1    | –          | ✓                    | –                   |
| Structure 2    | ✓          | ✓                    | –                   |
| Structure 3    | ✓          | –                    | ✓                   |

Fig. 4. Block diagram of explicit force control with neural network (NN) (all frequency)

Fig. 5. Block diagram of explicit force control with NN (high frequency)

force control with DOB and RFOB. This study presents three candidate structures, as listed in Table 3, to select the method for combining DOB and NN. The tick mark (✓) specifies that the compensator of the structure is implemented, while the dashed mark (–) specifies that the compensator is not implemented. NN (all frequencies) implies that the compensator works in all frequency ranges, while NN (high frequency) implies that the compensator works in frequency ranges higher than the cutoff frequency of the DOB. The block diagrams of Structures 2 and 3 are shown in Figs. 4 and 5, where \( C_f(s) \) and \( M_n \) denote the values generated by DOB in the case of NN (all frequencies) and NN (high frequency), respectively.

The block parts labeled Motor+Env. and DOB in Fig. 5 can be equivalently transformed as illustrated in Fig. 6. The compensation value generated by the NN (high frequency) is substituted from a similar point to the external and disturbance forces, as illustrated in Fig. 6.

The motion equations of Structures 2 and 3 are computed using Eqs. (9) and (10), respectively:

\[
M_2 x_{\text{res}} = C_f(s) \frac{M_n}{K_{\text{in}}} (f_{\text{cmd}} - f_{\text{ext}}) - (1 - G_d(s)) f_{\text{ext}} + f_{\text{dis}} + f_{\text{comp}}^n, \quad \ldots \quad (9)
\]

\[
M_2 x_{\text{res}} = C_f(s) \frac{M_n}{K_{\text{in}}} (f_{\text{cmd}} - f_{\text{ext}}) - (1 - G_d(s)) f_{\text{ext}} f_{\text{comp}}^n, \quad \ldots \quad (10)
\]

In Structure 3 or Eq. (10), the compensation value generated by NN (high frequency) also becomes the input of the transfer function as well as \( f_{\text{ext}} + f_{\text{dis}} \) in Fig. 2. Therefore, it aims at compensating the external and disturbance forces at a value higher than the cutoff frequency of the DOB as the feedforward compensator.

3.2 Structure of NN

The concept of recurrent NNs (RNNs) is useful when the time-sequence data is treated. NN nodes are connected not only feedforward direction but also the recurrent direction in the RNN. An example of network structure is illustrated in Fig. 7. Two units in the input layer, two units in the hidden layer with recurrent network, two units in the hidden 2 layer with recurrent network, and one unit in the output layer are shown in this example. Each layer has certain units, which are represented by circles.
leaky rectified linear unit (LReLU) is utilized as the activation function of all units. The number of units in each hidden layer is considered to be the same, and the outputs from the hidden 1, hidden 2, and output layers are computed according to Eqs. (11), (12), and (13), respectively.

\[
    o'_{h1} = \sum_i w_{bi,l} z_i^l + \sum_H w_{h1,h1} z_{h1}^{h1} + b_{h1},
\]

\[
    z'_{h1} = \text{LReLU}(o'_{h1}), \quad \cdots \cdots \cdots \cdots \cdots \cdot (11)
\]

\[
    o'_{h2} = \sum_H w_{h2,h1} z_{h1}^{h1} + \sum_H w_{h2,h2} z_{h2}^{h2} + b_{h2},
\]

\[
    z'_{h2} = \text{LReLU}(o'_{h2}), \quad \cdots \cdots \cdots \cdots \cdots \cdot (12)
\]

\[
    o'_o = \sum_H w_{o,h2} z_{h2}^{h2} + b_{o},\quad \cdots \cdots \cdots \cdots \cdots \cdot (13)
\]

The subscript of weight represents the connected layers. For instance, \(w_{bi,l}\) specifies the weight from the input layer to the hidden 1 layer, and \(w_{h2,h1}\) specifies the weight from the hidden 1 layer to the hidden 2 layer. \(w_{h1,h1}\) indicates the recurrent network and specifies the weight from the hidden 1 layer in the previous data to the hidden 1 layer in the current data.

The error between the command and response becomes an error function of all units. The number of units in each hidden layer is generally expected to be as large as possible to enhance the representation ability of variables. The learning was performed until the NN-based approach in all situations, where PI gain with DOB was considered as the conventional approach in Lines 9–10. Once the error function of the NN-based approach becomes smaller than that of the conventional approach, the position response takes a value smaller than one. The learning was performed until the NN-based approach achieves better performance than the conventional approach in all situations, where PI gain with DOB was considered as the conventional approach. The weights were updated at each sampling time if the error function of the NN-based approach was larger than the conventional approach in Lines 9–10. Once the error function of the NN-based approach became smaller than that of the conventional approach, the continuous number of trial successes was increased and the situation was updated in Lines 15–16. The error function was computed after one training was completed as follows:

\[
    J = \frac{1}{2}(c^{\text{cmd}} - f^{\text{ext}})^2, \quad \cdots \cdots \cdots \cdots \cdots \cdot (15)
\]

The objective of training is to minimize the cost function, and the estimated external force is expected to follow the command by iterating the process of gradient descent. The objective is satisfied even if the motor is accelerated in the negative direction owing to the inertial force, and the position response is added as a constraint (\(x^{\text{res}} \geq 0\)).

### 3.4 Learning Protocol

The network is trained on a simulation, and the trained network is directly utilized in the actual experimental setup. The environmental stiffness, environmental viscosity, and coefficient of command amplitude were changed to enhance the versatility of the system.

The learning algorithm is as presented subsequently.

```plaintext
1: while c < c' do
2:  if Violate constraints then
3:      Initialization
4:  else
5:      Initialization except weights
6:  end if
7:  while t \leq T do
8:    Motor Control
9:    if c = 0 then
10:      Update weights based on Eq. (14)
11:  end if
12:  t += ST
13: end while
14: Compute \(E_{\text{nn}}\) and \(E_{\text{conv}}\)
15: if \(E_{\text{nn}} < (\alpha E_{\text{conv}})\) then
16:    c += 1
17:  Update situation
18: end else
19:  c = 0
20: end if
21: end while
```

Here, the amount of improvement \(\alpha\) takes a value smaller than one. The learning was performed until the NN-based approach achieved better performance than the conventional approach in all situations, where PI gain with DOB was considered as the conventional approach. The weights were updated at each sampling time if the error function of the NN-based approach was larger than the conventional approach in Lines 9–10. Once the error function of the NN-based approach became smaller than that of the conventional approach, the continuous number of trial successes was increased and the situation was updated in Lines 15–16. The error function was computed after one training was completed as follows:
The error function shows the absolute mean square error of one training.

**3.5 Learning Conditions** The learning code was written in C language; moreover, the learning process was performed 10 times for each candidate structure and available composition of the NN. The learning was terminated after one hundred thousand iterations if the learning was not completed by the algorithm within its number of iterations. The learning was performed again in the actual experimental setup because certain elements of the experiment, such as contact, could not be described perfectly in the simulation. Therefore, a structure of the NN that requires fewer learning iterations is expected to be revealed and selected through the learning process.

The parameters that were setup for the learning process are listed in Table 5. The remaining parameters are similar to those listed in Table 2. While considering the controller gain setting, the proportional gain was initially tuned in the order of 0.1 by minimizing the error function similar to the limit sensitivity method. Then, the integral gain was tuned in the order of 1 by minimizing the error function. The tuned proportional and integral gains, which were designed for the conventional method, were applied to all candidates structures for a fair comparison. The learning rate was randomly chosen at the initialization step. The input of the NN was normalized by assuming the range of each variable. For instance, the assumed range of force command was $\pm 10$ and the input to NN was 0, 0.5, or 1 if the force command was $-10$ N, 0 N, or 10 N, respectively.

The parameters were set in different situations are listed in Table 6. One thousand different types of situations (twenty types of environmental stiffness values, ten types of environmental viscosity values, and five types of amplitudes) were considered. The command of motion was designed as shown in Fig. 8, where $A$ represents the coefficient of the command amplitude. The motion includes both the step and ramp motions, and these amplitudes and inclinations are varied based on the coefficient of the command amplitude.

**3.6 Learning Results** The learning results based on the different candidate structures with composition candidate 2, as listed in Table 4, are shown in Fig. 9, where $\delta^{str,1}$, $\delta^{str,2}$, $\delta^{str,3}$, and $\delta^{fan,2}$ denote the values of Structures 1, 2, and 3, and composition candidate 2 of the NN, respectively. The learning was not terminated within one hundred thousand iterations in the case of Structures 1 and 2, as shown in Fig. 9(b). The error functions gradually became smaller, and the situation was updated at approximately 12000 and 40000 iterations in Structures 1 and 2. However, once the situation was updated, where the error functions of conventional approaches were updated, the learned weight was not useful for the updated situation. DOB is a significantly powerful method for compensating the disturbance, and it is considerably difficult to achieve the work of DOB only by (R)NN with
a limited number of layers. The enlarged view of Structure 3 is shown in Fig. 9(c). It can be observed that the amount of error functions of the proposed method became smaller than that of the conventional method even if the situation was updated. Structure 3 enables the avoidance of the interference between DOB and NN, and the learning of the NN could be appropriately performed. The compensation by NN improves the performance. Therefore, the implementation of the NN for explicit force control with DOB improves the performance; however, it should be designed without interrupting the work of DOB.

The learning results of Structure 3 with five available compositions of NN are shown in Fig. 10. Compositions 3, 4, and 5 could not complete the learning within one thousand iterations in certain trials. Conversely, compositions 1 and 2 terminated the learning within the limited number of iterations in all 10 trials. In particular, composition 2 required fewer iterations from the average of 10 trials in each composition. In summary, this study implemented Structure 3 with composition 2.

3.7 Fault Tolerance Fault tolerance is also designed for the case when the learning of the NN is not performed appropriately. Because NN may not operate properly if it is utilized in a situation that it has not encountered previously, this study cannot guarantee that the command designed in Fig. 8 includes all types of motions. The force references of the conventional method \( f_{\text{cmd}} = C_1(s) \cdot f_{\text{ref}}^\text{prop} + f_{\text{dist}}^\text{conv} \) and proposed methods \( f_{\text{ref}}^\text{prop} \) are computed.

The following four cases are the desired generation of force references when the force command takes a value larger than zero:

1. \( f_{\text{cmd}} = f_{\text{cmd}} \land (f_{\text{ref}}^\text{prop} \leq f_{\text{ref}}^\text{prop},-1) \),
2. \( f_{\text{cmd}} = f_{\text{cmd}} \land (f_{\text{ref}}^\text{prop} > f_{\text{ref}}^\text{prop},-1) \),
3. \( f_{\text{cmd}} > f_{\text{cmd}} \land (f_{\text{ref}}^\text{prop} < f_{\text{ref}}^\text{prop},-1) \),
4. \( f_{\text{cmd}} < f_{\text{cmd}} \land (f_{\text{ref}}^\text{prop} > f_{\text{ref}}^\text{prop},-1) \).

The force reference of the proposed method is utilized if the above-mentioned four cases are satisfied, while the force reference of the conventional method is utilized if these cases are violated.

The switching algorithm also contributes to the safety of the experiment. The stability of the control system cannot be assured because the transfer function of the NN cannot be precisely described. The NN works as the feedforward component; moreover, the setting constraints to the NN compensation value or cutting off NN enables the avoidance of unacceptable motion.

4. Simulation

4.1 Outline Simulations were conducted to observe the performance of the proposed NN-based explicit force control. The objectives of the simulations are as follows: 1. to compare the proposed method with the conventional method; 2. to show the performance for a wide frequency range of disturbances. In the first simulation, the leaning of NN was performed and the proposed method was compared with the conventional method (PI + DOB). In the second simulation, the sweep sine disturbance was intentionally added to check the compensation performance of the NN. The motor was assumed to contact both hard and soft objects, and the environmental impedances were set as listed in Table 7. The rest of the parameters were set to the values listed in Table 2.

4.2 Comparison between Proposed and Conventional Methods The motor was simulated to contact hard and soft objects for a time ranging from 0 to 5 s and 5 to 10 s. The force and position responses and root mean square error (RMSE) of the force responses are shown in Fig. 11 and listed in Table 8. The proposed method achieved faster command tracking and a smaller RMSE. In particular, the force response oscillated faster in the proposed method when the step commands were provided.

4.3 Performance against Wide Frequency Range of Disturbance The motor was simulated to contact hard and soft objects from 0 to 5 s and 5 to 10 s with a wide frequency range of disturbances. A sweep was intentionally added as the disturbance, and the amplitude of disturbance was ±0.1 N. The frequency was gradually increased and decreased from 10 to 1000 rad/s in steps of 0.1 s. The command of force control was constantly set as 1 N to evaluate the performance against the disturbance. The input variables of the NN were the same as those illustrated in Fig. 5; further, the learning was performed as discussed in Section 3. However, the constant value of 1 N was set as the command of motion, and the coefficient of command amplitude was not changed in this simulation. The proposed method was compared not only with the conventional approach but also with the conventional approach with high control gains. The proportional and integral gains in the case of high control gains were set as 1.6 kg\(^{-1}\) and 190 kgm\(^{-1}\), respectively. These gains were designed based on the error function. The force result, position result, and added disturbance and estimated result are shown in Fig. 12. The simulation results show that the enhancement of control gains marginally improved the response although its amount of improvement was limited. Conversely, the output of NN almost estimated the added disturbance \( f_{\text{dist}} \), and the NN compensated the added disturbance. NN has the capability to compensate for a wide frequency range.

| Description                        | Amount        |
|------------------------------------|---------------|
| Environmental stiffness of hard object | 25000 N/m     |
| Environmental viscosity of hard object | 100 N/s/m     |
| Environmental stiffness of soft object | 2000 N/m      |
| Environmental viscosity of soft object | 15.4 N/s/m    |
of disturbances by the feedforward method if the learning is properly performed.

5. Experiments

5.1 Outline of Experiments Experiments were conducted to confirm the effectiveness of the proposed method. Four types of experiments were performed: frequency, contact, disturbance, and fault tolerance experiments. The conventional method was composed of PI gain + DOB, while the proposed method was composed of PI gain + DOB + NN (high frequency) with composition 2.

5.1.1 Frequency Experiment In the frequency experiment, the sweep sine was utilized as the input, and the frequency response of the conventional and proposed methods were measured by a contact with a screw box. The magnitude and phase of the conventional and proposed methods were experimentally measured and compared to determine the performance of the controller in the frequency domain.

5.1.2 Contact Experiment In the contact experiment, the motor contacted the screw box (hard object) and sponge (soft object) using the conventional and proposed controllers to compare the performance of the controllers. In addition to these two controllers, the feedback-based NN (NN:FB) and the feedforward-based NN (NN:FF) were also considered to observe the effects of the feedback and feedforward components.

5.1.3 Disturbance Experiment In the disturbance experiment, unexpected disturbances were deliberately added to the system to verify whether the proposed system could handle the disturbances. The motor contacted the screw box, similar to the contact experiment, and the step- and lamp-shaped disturbances were added in the negative direction from 2.5 to 2.8 s and from 4.5 to 4.8 s, respectively.

5.1.4 Fault Tolerance Experiment In the fault tolerance experiment, the amounts of certain weights were deliberately changed from the arbitrary value to zero at 4.5 s to check whether the undesirable control motion could be avoided by the four cases defined in Section 3.7. The force reference of the conventional method was utilized subsequently.

5.2 Experimental Setup The linear motor shown in Fig. 13 was utilized. The weights were learned in the simulation and applied in the experimental setup. Learning was also performed in the experimental setup again to adapt to the real environment, which could not be modeled in the simulation environment. Because the gain setup, which minimized the
Neural Network-based Force Control with Disturbance Observer (Toshiaki Okano et al.)

The training of the NN was performed using six items and the testing or experiment was conducted using two items, as shown in Fig. 14, and listed in Table 9. The learning result of the experimental setup is shown in Fig. 15. Because the error function of the proposed method became smaller when compared to the product of the error function of the conventional method and the amount of improvement from the 42nd to 47th iterations, the proposed method that was composed of the same weights could contact six training objects with a smaller number of error functions. In the contact experiment, NN:FB and NN:FF were also considered. These NNs were also initially trained by the simulation; then, the training was performed in the actual experimental setup. The structures of each NN were the same as that in the proposed method; however, the number of input variables was different. Only the error was considered as the input variable in NN:FB, while the remaining three variables were considered as the input in NN:FF.

5.3 Experimental Results

5.3.1 Frequency Experiment

The results of magnitude and phase are shown in Fig. 16. The magnitude and phase in the low-frequency range (approximately 0.01 rad/s to 100 rad/s) were almost the same between the conventional and proposed methods. The decay of the magnitude or the lag in the phase began at a higher frequency in the proposed method. The proportional and integral gains were tuned based on the error function; however, a combination of NNs can be considered to further improve the control performance.

5.3.2 Contact Experiment

The force and position responses of the contact with the screw box and sponge are shown in Figs. 17 and 18, respectively. The result of the proposed methods followed the force step command among other methods pertaining to the contact to the screw box and sponge. NN:FB and NN:FF also achieved better performance than the conventional approach. In particular, the force response of NN:FF was better than that of NN:FB.
Neural Network-based Force Control with Disturbance Observer  
(Toshiaki Okano et al.)

the contact experiment with the sponge. All methods appropriately tracked the lamp command from 5 to 9.5 s. The RMSEs of the force response are listed in Table 10. The proposed method achieved the smallest RMSEs in the contact with both the screw box and sponge. Both the feedback and feedforward components contributed to the performance improvement of force control with the DOB.

5.3.3 Disturbance Experiment The force and position responses are shown in Fig. 19 and the RMSEs of the force response are listed in Table 11. The proposed method could tackle unexpected disturbances as well as the conventional method.

5.3.4 Fault Tolerance Experiment The force and position responses are shown in Fig. 20 and the RMSEs of the force response are listed in Table 12. Even if the method of reference generation was suddenly changed, the force controller continued to follow the force command.

Table 10. RMSEs of the contact experiment

| Contact object | Method         | RMSE [N × 10^{-2}] |
|----------------|----------------|---------------------|
| Screw box      | Conventional   | 1.85                |
| Screw box      | Proposed (NN:FB) | 1.74               |
| Screw box      | Proposed (NN:FF) | 1.68               |
| Screw box      | Proposed (NN:FB+FF) | 1.59              |
| Sponge         | Conventional   | 2.63                |
| Sponge         | Proposed (NN:FB) | 2.45               |
| Sponge         | Proposed (NN:FF) | 2.32               |
| Sponge         | Proposed (NN:FB+FF) | 2.23              |

Table 11. RMSEs of disturbance experiment

| Method          | RMSE [N × 10^{-2}] |
|-----------------|---------------------|
| Conventional    | 2.73                |
| Proposed        | 2.31                |
Neural Network-based Force Control with Disturbance Observer (Toshiaki Okano et al.)

6. Conclusion

This study proposed an NN-based explicit force control method with DOB and RFOB. The compensator of the NN was designed by comparing certain candidate structures and available compositions. Because the DOB is already a strong method, the NN was designed in a frequency range higher than the cutoff frequency of the DOB (Structure 3). The weights of the NN were updated at each sampling time based on the cost or error function in both the simulation and actual setups. The simulation and four experiments, i.e., frequency, contact, disturbance, and fault tolerance, were performed to demonstrate the effectiveness of the proposed method. In future works, the versatility of the system should be considered. For instance, this study considered simple steps or ramp commands. However, the adaptability to various command shapes is necessary if the proposal is applied to bilateral control, in which the operator manipulates the motor using various command shapes.

Appendix

1. Transfer Function of Force Control

The block diagram and transfer function of Motor+Env. are shown in app. Fig. 1 and computed as follows:

\[ G_3(s) = \frac{K_G(s)}{M_s^2 + Z(s)} = \frac{K_G(s)}{Z(s)} \]  \hspace{1em} (A1)

The block diagram of an open loop is as shown in app. Fig. 2. The variable \( \theta \) is calculated using Eq. (A2) and recomputed to Eq. (A3), as shown below:

\[ \theta = \frac{C_I(s)M_a}{K_{in}} + G_d(s)\theta - \frac{G_3(s)}{K_{in}}[1 - G_d(s)]q_dM_n\theta, \]  \hspace{1em} (A2)

\[ \theta = \frac{C_I(s)M_a}{K_{in}} + G_d(s)\theta - \frac{G_3(s)}{K_{in}}[1 - G_d(s)]q_dM_n\theta \]  \hspace{1em} (A3)

References

(1) K. Ohnishi, M. Shibata, and T. Murakami: “Motion control for advanced mechatronics”, IEEE/ASME Transactions On Mechatronics, Vol.1, No.1, pp.56–67 (1996)

(2) T. Murakami, F. Yu, and K. Ohnishi: “Torque sensorless control in multidegree-of-freedom manipulator”, IEEE Transactions on Industrial Electronics, Vol.40, No.2, pp.259–265 (1993)

(3) Y. Nagatsu and S. Katsura: “High-order disturbance estimation using Kalman filter for precise reaction-torque control”, Advanced Motion Control, 2004, 8th IEEE International Workshop On, pp.79–84 (2016)

(4) S. Sakaano, T. Sato, and K. Ohnishi: “Force-based disturbance observer for dynamic force control and a position/force hybrid controller”, IEEE Transactions on Electrical and Electronic Engineering, Vol.8, No.5, pp.505–514 (2013)

(5) L.G. Garcia-Valdivinos, V. Parra-Vega, and M.A. Arteaga: “Higher-order sliding mode impedance bilateral teleoperation with robust state estimation under constant unknown time delay”, Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp.1293–1298 (2005)

(6) H. Gomi and M. Kawato: “Neural network control for a closed-loop system using feedback-error-learning”, Neural Networks, Vol.6, No.7, pp.933–946 (1993)

(7) M. Teshnehlab and K. Watanabe: “Neural network controller with flexible structure based on feedback-error-learning approach”, Journal of Intelligent and Robotic Systems, Vol.15, No.4, pp.367–387 (1996)

(8) E. Kayacan: “Sliding mode control for systems with mismatched time-varying uncertainties via a self-learning disturbance observer”, Transactions of the Institute of Measurement and Control, Vol.41, No.5, pp.2039–2052 (2019)

(9) J. Nakanishi and S. Schaal: “Feedback error learning and nonlinear adaptive control”, Neural Networks, Vol.17, No.10, pp.1453–1465 (2004)

(10) T. Fukuda and T. Shibata: “Theory and applications of neural networks for industrial control systems”, IEEE Transactions on Industrial Electronics, Vol.39, No.6, pp.472–489 (1992)

(11) Z. Su and K. Khorasani: “A neural-network-based controller for a single-link flexible manipulator using the inverse dynamics approach”, IEEE Transactions on Industrial Electronics, Vol.48, No.6, pp.1074–1086 (2001)

(12) H. Gao, W. He, C. Zhou, and C. Sun: “Neural network control of a two-link flexible robotic manipulator using assumed mode method”, IEEE Transactions on Industrial Informatics, Vol.15, No.2, pp.755–765 (2018)

(13) T. Tsuji, K. Ito, and P.G. Morasso: “Neural network learning of robot arm impedance in operational space”, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol.26, No.2, pp.280–298 (1996)

(14) X. Yao, Y. Zhang, B. Li, Z. Zhang, and X. Shen: “Machining force control with intelligent compensation”, The International Journal of Advanced Manufacturing Technology, Vol.69, No.5–8, pp.1701–1715 (2013)

(15) S. Bhasin, K. Dupree, P.M. Patre, and W. E Dixon: “Neural network control of a robot interacting with an uncertain viscoelastic environment”, IEEE Transactions on Control Systems Technology, Vol.19, No.4, pp.947–955 (2010)

(16) J. Mazumdar and R.G. Harley: “Recurrent neural networks trained with back-propagation through time algorithm to estimate nonlinear load harmonic current”, IEEE Transactions on Industrial Electronics, Vol.55, No.9, pp.3483–3491 (2008)
The transfer function from $\theta$ to the output is computed as follows:

$$\frac{y}{\theta} = K_{t,n}G_{r}(s) - G_{3}(s)[1 - G_{r}(s)]g_{d}M_{n}.$$  \hfill (A4)

Therefore, the transfer function from input to output is computed as follows:

$$\frac{y}{u} = C_{f}(s)\frac{M_{n}}{K_{t,n}}K_{t,n}G_{r}(s)\frac{Z(s)}{G_{3}(s)}\frac{[1 - G_{r}(s)]g_{d}M_{n}}{[1 - G_{d}(s)][Z(s) + K_{t,n}g_{d}M_{n}]}.$$  \hfill (A5)

Eq. (A5) is recomputed by substituting Eq. (A1) to obtain the following equation:

$$\frac{y}{u} = C_{f}(s)\frac{M_{n}}{K_{t,n}}K_{t,n}G_{r}(s)\frac{Z(s)}{G_{3}(s)}\frac{[1 - G_{r}(s)]g_{d}M_{n}}{[1 - G_{d}(s)][Z(s) + K_{t,n}g_{d}M_{n}]}.$$  \hfill (A6)

---

**Toshiaki Okano** (Student Member) received a B.E. degree in system design engineering and an M.E. degree in integrated design engineering from Keio University, Yokohama, Japan, in 2015 and 2018, respectively, where he is currently working toward a Ph.D. degree. His research interests include motion control, robotics, and haptics.

**Takahiro Nozaki** (Senior Member) received his B.E., M.E., and Ph.D. degrees from Keio University, Yokohama, Japan, in 2010, 2012, and 2014, respectively. In 2014, he joined Yokohama National University, Yokohama, Japan, as a Research Associate. In 2015, he joined the Keio University, where he is currently an Assistant Professor. He is also with Massachusetts Institute of Technology, Massachusetts, United States, as a Visiting Scientist. He was selected as a winner of the IEEE Industrial Electronics Society Under 35 Innovators Contest in 2019.

**Toshiyuki Murakami** (Fellow) received his B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1988, 1990, and 1993, respectively. In 1993, he joined the Department of Electrical Engineering, Keio University, where he is currently a professor with the Department of System Design Engineering. From 1999 to 2000, he was a Visiting Researcher at the Institute for Power Electronics and Electrical Drives, Aachen University of Technology, Aachen, Germany. His research interests include robotics, intelligent vehicles, mobile robots, and motion control.