

Improvement of Operator Position Prediction in Teleoperation Systems with Time Delay: Simulation and Experimental Studies on Phantom Omni Devices

Behnam Yazdankhoo^a, Moein Nikpour^a, Borhan Beigzadeh^{*a}, Ali Meghdari^b

^a Biomechatronics and Cognitive Engineering Research Lab, School of Mechanical Engineering, Iran University of Science and Technology, Tehran, Iran.

^b Social & Cognitive Robotics Laboratory, Center of Excellence in Design, Robotics and Automation (CEDRA), Sharif University of Technology, Tehran, Iran.

Received 7 February, 2019

Abstract

An online operator position prediction approach based on artificial neural network for teleoperation systems is proposed in this paper, which predicts future position of operator's hand based on current available data. The neural network gathers inputs for some time at the beginning of the operation, then is trained, and is finally exploited through the rest of the operation. Superiority of the proposed approach can be investigated from two aspects. Firstly, no limiting assumption is required in this approach in contrast with the proposed methods in the literature. Secondly, unknown operator intention can be dealt with in real time if it is not too sudden and unpredictable. Two different scenarios are considered in this paper: in scenario I a simple harmonic motion is both applied and predicted, whereas in scenario II not only the applied motion is more complicated, but also it is different from the motion which is supposed to be predicted. The results of the second scenario show that the designed architecture can be readily extended to a variety sort of situations in which little information exists regarding operator intention. Computer simulations and experiments using Phantom Omni haptic devices further validate the feasibility and performance of the proposed approach, i.e. master and slave robots can move simultaneously with no specific a priori knowledge about operator intention, despite large time delay in the system.

© 2019 Jordan Journal of Mechanical and Industrial Engineering. All rights reserved

Keywords: operator position prediction, neural network, online prediction, teleoperation system, time delay;

1. Introduction

Novel approaches based on predictive control methods have been recently proposed in order to circumvent the time delay in the teleoperation system [1]. From a broader perspective, predictive control methods can be considered as a part of environment-, operator-, and task-adapted (EOT-adapted) controllers [2]. Here we stick to a simpler categorization, and we consider two main predictive approaches: predicting the environment model and predicting the operator motion.

To date, most of the works are concerned with the first approach. This method, which is called *model-mediated teleoperation* [3], has been widely used in designing teleoperation systems in recent years. Sample works regarding this approach can be found in [4, 5]. On the other hand, predicting the operator motion (Figure 1) can also play an important role in improving the quality of teleoperation systems, as Jarrassé et al. [6] proved the influence of human motion prediction on increasing transparency of teleoperation systems. This is, however, a more complicated task since the human operator might

show unpredictable motions [7]. The works which have been done regarding this issue can be categorized into two main approaches: offline and online.

Offline motion prediction is fundamentally based on minimum principles [8], with the most widely used principle being the minimum jerk principle [9], which states that between two predefined points the human hand follows the trajectory whose jerk is the minimum amount. The basic concept of online motion prediction is, however, predicting the human behavior during the operation based on the online data generated by him/her. Linear prediction [10], polynomial or spline predictors [11], double exponential smoothing (DES) [12, 13], autoregressive (AR) method [14] and Markov-chain-based Kalman filter [15] are some samples of online methods. Recent methods based on state observer have also been proposed [16]. Unlike offline methods which are basically derived from physical principles, the common core of online methods is numerical data. This advantage makes online approaches more adaptive to various applications, but at the same time rather more intricate to design.

Each of the above-mentioned methods for online prediction has also some disadvantages. For instance, DES

* Corresponding author e-mail: b_beigzadeh@iust.ac.ir

can be utilized when the data shows a trend, but not seasonality. Triple exponential smoothing (TES) shall be used in this case instead [17]. As another example, AR method models the future state as a linear combination of the current and previous states, which may fail to correctly predict the future states for more complex motions where this dependency is nonlinear. To mitigate these restrictions, a prediction method based on artificial neural network (NN) is proposed in this paper.

Control of teleoperation systems based on NN has recently gained researchers' attentions [18, 19]. Due to the universal approximation property of NNs, any kind of linear or nonlinear relationship between the inputs and outputs can be modeled with a proper NN [20]. It might be interesting to say that it has been contended that even a quadratic regression model is computationally less efficient in comparison with a NN [21]. Also, from prediction task perspective, NNs do not require any assumption regarding the trend or seasonality of the motion beforehand. These properties have made NN a suitable choice for predicting the future states of a system. Nicolau et al., for instance, exploited a NN for prediction of the roll motion of a ship [22]. However, to the best of authors' knowledge, no previous work has utilized NN for predicting operator motion in teleoperation systems.

In this work, we make use of an artificial NN to predict the position of the operator's hand online. The NN is trained for a short time at the beginning of the operation, and then can predict the operator movements thereafter. Two main contributions can be mentioned for the proposed method: Firstly, using the proposed approach, the teleoperation system is able to predict not only the expected operator motions, but also the unexpected ones if they are not unpredictably sudden. This property enables human operator to carry out more than one single task at once, thus there is no need to reinitialize the operation if the task is decided to be altered slightly. Secondly, there is no restricting assumption in the proposed method from applicability point of view, i.e. no specific detailed information about the operator's intended motion is needed beforehand. This feature makes the applicability of the system much wider; from surgery and suturing in medical robotics, which is believed to have major research impact in the next decade [23], to ordinary pick and place tasks in industry. It should also be asserted that the defined motion is set just as a general case; the NN system can predict any other type of motion.

2. System architecture

In this work, both free motion (no environment) case and constant known time delay is assumed, shown in Figure 2. The structure of the NN is also illustrated in Figure 3, which is a feedforward network with three layers. The input layer is comprised of the current and

previous positions of the master robot. If there are m neurons in the input layer and s is assumed to be the sample time, the input to the NN will be $\mathbf{X}(t) := [x(t) \ x(t-s) \ x(t-2s) \ \dots \ x(t-(m-1)s)]^T$. The hyperbolic tangent function is considered as the activation function for the hidden layer.

The output layer always includes one neuron which is the predicted future position of the master robot, namely $\hat{x}(t + \tau)$, where τ denotes the time delay and the accent $\hat{}$ refers to the predicted version of the respective parameter. The activation function for this neuron is the identity function, i.e. $f(x) = x$.

The NN training is carried out based on the backpropagation gradient descent method. According to this method, the weight update in each iteration or epoch follows the relation (2.1),

$$\Delta w_{ij}^{(n)} = -\eta \frac{\partial E}{\partial w_{ij}^{(n)}} \quad (2.1)$$

where $w_{ij}^{(n)}$ represents the connection weight between the i th neuron in the $(n-1)$ th layer and the j th neuron in the n th layer, $n = 2, 3, \dots, N$ with N denoting the total number of layers, and Δ shows the change in the respective weight. Also η is the learning rate, which is assumed to be constant, and E is the objective function to be minimized which is $E = \frac{1}{P} \sum_{p=1}^P (o_p - d_p)^2$, where o and d are the actual and desired outputs of the NN, respectively, and P represents the total number of presented training sets to the network. The update of the biases follows the same rule by simply replacing $w_{ij}^{(n)}$ with $b_j^{(n)}$ in the relation (2.1). More detailed explanation about the concepts and formulation of the gradient descent method can be found in [24].

Generally, there are two approaches to training a NN: batch-mode and incremental-mode [24]. In this work, both methods will be used for training the NN.

As mentioned before, the proposed teleoperation system in this paper is designed based on online prediction scheme. The system performs according to the following procedure:

In the first phase, the operator moves the master robot reciprocally for some time. Knowing a rough estimation of the period of the motion beforehand, which is denoted by T_h , we record the operator's motion for the first T_1 seconds, where T_1 is slightly bigger than T_h . The slave robot receives delayed signals in this phase.

In the second phase, the network is trained in T_2 seconds and the operator are asked to cease the operation. Due to small amount of T_2 , this interruption is expected not to negatively affect the whole operation.

Finally, after $T_1 + T_2$ seconds, the NN is completely ready to be implemented in the system. Although the motion is expected not to be too far from the trained one in this phase, it is not indeed a restricting assumption, as we will show later in section 5 that the NN can predict even untrained motions and there is no need to keep the same motion during the whole operation.

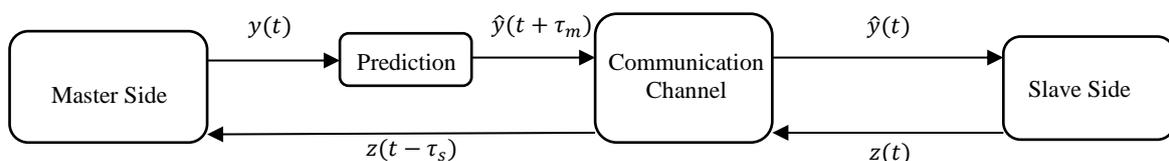


Figure 1. A teleoperation system with master state prediction. y and z are two arbitrary parameters

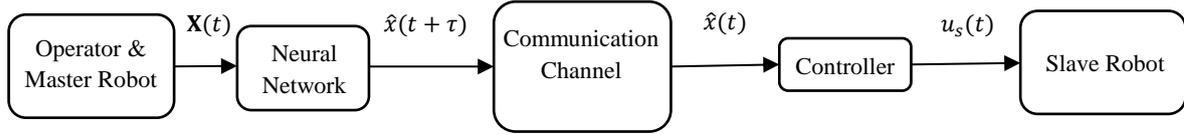


Figure 2. Architecture of a unilateral teleoperation system in which a neural network carries out the prediction task.

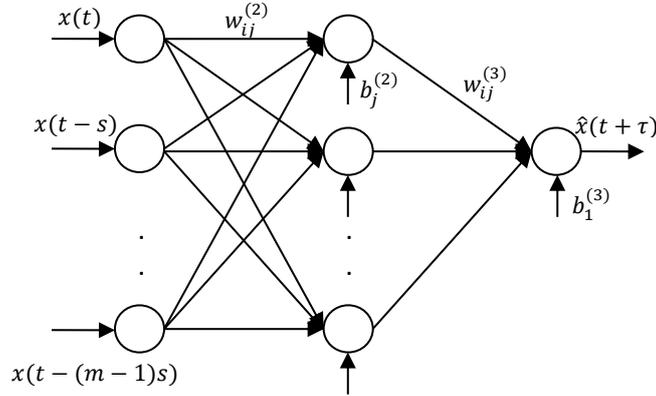


Figure 3. Structure of the proposed neural network.

3. Controller design for the slave side

3.1. Inverse kinematics of Phantom Omni

Phantom Omni haptic device provides three translational and three rotational degrees of freedom. The translational degrees of freedom are resulted from the three angles q_i ($i = 1, 2, 3$) which are depicted in Figure 4. The inverse kinematics relations are expressed as follows [25]

$$\begin{aligned} q_1 &= -\text{atan2}(x_e, z_e + L_4); \quad q_2 = \gamma + \beta; \\ q_3 &= q_2 + \alpha - \frac{\pi}{2} \end{aligned} \quad (3.1)$$

where $\alpha := \cos^{-1}\left(\frac{L_1^2 + L_2^2 - r^2}{2L_1L_2}\right)$, $\beta := \text{atan2}(y_e - L_3, R)$, $\gamma := \cos^{-1}\left(\frac{L_1^2 + r^2 - L_2^2}{2L_1r}\right)$, $R := \sqrt{x_e^2 + (z_e + L_4)^2}$ and $r := \sqrt{x_e^2 + (z_e + L_4)^2 - (y_e - L_3)^2}$. Also $L_1 = L_2 = 133.35 \text{ mm}$, $L_3 = 23.35 \text{ mm}$, and $L_4 = 168.35 \text{ mm}$. For more detailed description and proof, please refer to the main reference [25].

3.2. Controller design

PID controller is adopted for the slave side in this paper. If the actuators are in Cartesian space themselves, the error can be defined straightforwardly as $\tilde{e}^{(x)}(t) := x_m(t - \tau) - x_s(t)$ before and during the NN training, and $e^{(x)}(t) := \hat{x}_m(t) - x_s(t)$ after the NN training. However, if the actuators function in joint space, the error

should be defined as $\tilde{e}_i^{(q)}(t) := q_{i,m}(t - \tau) - q_{i,s}(t)$ before and during the NN training, and $e_i^{(q)}(t) := \hat{q}_{i,m}(t) - q_{i,s}(t)$ after the NN training for each joint ($i = 1, 2, 3$). Hence, the control input for the slave robot is obtained as

$$\begin{aligned} u_s(t) &= k_p^{(x)} e^{(x)}(t) + k_d^{(x)} \dot{e}^{(x)}(t) \\ &\quad + k_i^{(x)} \int e^{(x)}(t) dt \end{aligned} \quad (3.2)$$

in Cartesian space, and

$$\begin{aligned} u_{s,i}(t) &= k_{p,i}^{(q)} e_i^{(q)}(t) + k_{d,i}^{(q)} \dot{e}_i^{(q)}(t) \\ &\quad + k_{i,i}^{(q)} \int e_i^{(q)}(t) dt \end{aligned} \quad (3.3)$$

in joint space, where u_s is the slave control input, k_p , k_d and k_i are PID constant gains, superscripts (x) and (q) denote the Cartesian and joint space respectively, and subscript $i = 1, 2, 3$ represents the joints of Phantom Omni. Note that before and during the NN training, $e(t)$ is replaced by $\tilde{e}(t)$ in (3.2) and (3.3).

We consider a one-degree-of-freedom motion in x -direction of Cartesian coordinates in this paper. So, for obtaining $\hat{q}_{i,m}(t)$ from the inverse kinematic relations we set $\hat{y}_m(t) = y_{des}$ and $\hat{z}_m(t) = z_{des}$, where y_{des} and z_{des} are constant desired positions in y - and z -directions for the end effector on which it is going to stay during the operation, while $\hat{x}_m(t)$ is attained from the NN output (section 2).



Figure 4. Phantom Omni reference XYZ coordinates, joints q_i and their resulting translational DoFs.

4. Stability analysis

There are well-known approaches for analyzing stability of teleoperation systems, such as absolute stability which considers the system as an input/output network [26], passivity which deals with energy generation/dissipation of the system [27], and methods such as Lyapunov analysis [28]. We use the absolute stability method in this paper. For this aim, the two-port network (Figure 5) should first be modeled as

$$\begin{bmatrix} F_h(s) \\ -V_s(s) \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} V_m(s) \\ F_e(s) \end{bmatrix} \quad (4.1)$$

where $V_s(s)$, $V_m(s)$, $F_h(s)$ and $F_e(s)$ are the velocity of the slave and master robots, and the human and environment forces, respectively, in Laplace space. The matrix containing the h-parameters is the hybrid matrix which is defined by the following relations if we assume linear second order dynamics for the desired behavior of the master and slave robots:

$$\begin{cases} h_{11} := \left. \frac{F_h(s)}{V_m(s)} \right|_{F_e=0} = \bar{m}_m s + \bar{b}_m + \frac{\bar{k}_m}{s} \\ h_{12} := \left. \frac{F_h(s)}{F_e(s)} \right|_{V_m=0} = k_1 e^{-\tau_s s} \\ h_{21} := \left. \frac{-V_s(s)}{V_m(s)} \right|_{F_e=0} = -k_2 e^{-\tau_m s} \\ h_{22} := \left. \frac{-V_s(s)}{F_e(s)} \right|_{V_m=0} = \frac{s}{\bar{m}_s s^2 + \bar{b}_s s + \bar{k}_s} \end{cases} \quad (4.2)$$

where \bar{m} , \bar{b} , \bar{k} and τ represent desired mass, desired damping coefficient, desired stiffness and time delay, respectively, and the subscripts m and s denote the master and slave robots, respectively. $k_1 > 0$ and $k_2 > 0$ are also scaling factors for force and position, respectively.

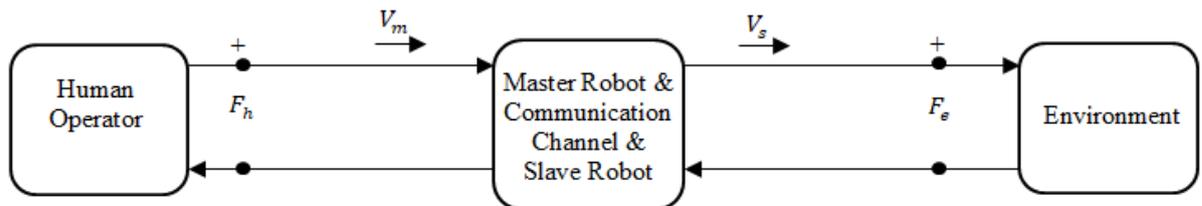


Figure 5. A two-port teleoperation system with classical architecture

Then, according to Llewellyn's stability criteria, the system is absolutely stable if the following conditions are met:

- No poles of h_{11} and h_{22} lie on the right half plane
- If any of poles of h_{11} and h_{22} lie on the imaginary axis, they are simple with real and positive residues
- For all real values of ω :
 - (a): $\text{Re}(h_{11}) \geq 0$
 - (b): $\text{Re}(h_{22}) \geq 0$
 - (c): $2\text{Re}(h_{11})\text{Re}(h_{22}) - \text{Re}(h_{12}h_{21}) - |h_{12}h_{21}| \geq 0$

The above conditions, provided that the operator and environment are passive (which is almost always the case [29]), result in a stable teleoperation system. The first and second conditions in addition to (a) and (b) of the third condition are fulfilled by choosing positive desired impedance parameters [30]. However, satisfying relation (c) of the third condition restricts the choices for the impedance parameters based on the time delay of the system.

Now we want to investigate the stability of the system if we add prediction blocks on master and slave sides. We have ideally $\hat{F}_e(s) = F_e(s)e^{\tau_s s}$ and $\hat{V}_m(s) = V_m(s)e^{\tau_m s}$ for the components which relate the transmitting signals together. By some mathematical manipulation, it is straightforward that $\hat{h}_{11} = h_{11}$ and $\hat{h}_{22} = h_{22}$. Also the other two components can be obtained as $\hat{h}_{12} := \left. \frac{\hat{F}_h(s)}{\hat{F}_e(s)} \right|_{V_m=0} = k_1$ and $\hat{h}_{21} := \left. \frac{-\hat{V}_s(s)}{\hat{V}_m(s)} \right|_{F_e=0} = -k_2$. The first and second conditions and (a) and (b) of the third one have obviously not been changed in comparison with (4.2), since the predictors do not affect the operator-master robot and the environment-slave robot interactions, and thus $\text{Re}(h_{11})\text{Re}(h_{22}) \geq 0$. Therefore, only (c) of the third condition should be investigated in order to prove the stability of the system. That condition is also met, because we have

$$\begin{aligned} 2\text{Re}(\hat{h}_{11})\text{Re}(\hat{h}_{22}) - \text{Re}(\hat{h}_{12}\hat{h}_{21}) - |\hat{h}_{12}\hat{h}_{21}| \\ = 2\text{Re}(h_{11})\text{Re}(h_{22}) + 2k_1k_2 \geq 0 \end{aligned} \quad (4.3)$$

where k_1k_2 is positive due to the definition presented previously.

Finally, provided that the outputs of the prediction block accurately converge to the actual values of their respective input parameters and the designed controller in section 3 is stable, we can conclude from the above-mentioned explanations that the whole proposed teleoperation system in this paper is stable, since it is clearly a special case of the system investigated above.

5. Results

In the following subsections, the simulation and experimental results are presented. The parameters regarding the whole system and the NN are listed in Table 1. To find the proper sample time for training the NN, we ran the simulation test with different sample times, and we arrived at a conclusion that the best one is 0.01 s, as mentioned in Table 1. To provide an example, we compared the sample times of 0.01 s and 0.001 s. It was found that with fixed structures, the NN training time in the latter case was approximately more than ten times the former case, and furthermore, the generality of the latter case results was highly poorer than the former case.

Two scenarios are designed in this section to validate the effectiveness of the proposed approach, the descriptions of which are given below:

Scenario I: In this scenario, a simple input force with an amplitude and frequency is applied by the operator and the same motion continues after the training. This scenario is to investigate whether or not the online trained NN is able to reproduce the results presented to it as input.

Scenario II: In this scenario, a more complex input force with amplitude and frequency is applied by the operator and the same motion continues after the training, but after some time, the amplitude and frequency are altered. This is done in order to investigate the prediction capability of the trained NN in the proposed teleoperation architecture when encountering untrained motions.

Note that in Figures 6 to 9 in this section, the end of T_1 is marked by the first vertical dash-dot line, and the end of T_2 is marked by the second dash-dot line. It should also be stated that all inputs to the NNs are normalized in the interval $[-1, +1]$.

5.1. Simulation results

For simulation, a one-degree-of-freedom teleoperation system in x-direction of Cartesian XYZ system was simulated by means of Simmechanics library (second generation) of MATLAB's Simulink. The dynamics of the master and slave robots were considered as linear second order ones with $M_m = M_s = 0.223$ kg, $B_m = B_s = 17.227$ N.s/m and $K_m = K_s = 6.286$ N/m, where M , B and K denote mass, damping coefficient and stiffness, respectively. This dynamics can represent a Phantom Omni robot [31]. NN training is performed using batch-mode method. A constant time delay τ is also simulated for the system. The slave control input in this case is achieved from (3.2). The system and NN parameters for the two scenarios are summarized in Table 2.

Scenario I: The simulated human force which is applied to the master robot is illustrated in Figure 6(a). The results for this scenario are depicted in Figures 6(b) to 6(d).

As can be seen from Figure 6(b), the slave robot is able to predict the master robot motion and move ahead of it, which results in simultaneous motion of the master and slave robots, as can be seen in Figure 6(c). This leads to a highly transparent, yet stable, teleoperation system in which the time delay is not felt by the operator. A comparison between the pre-training and post-training portions of the operation in Figure 6(c) will further prove this fact. It is noticeable that a code written in MATLAB

environment indeed needs more amount of time to train a NN in comparison with codes written in C or C++, as will be shown later in section 5-2. Therefore, the amount of time T_2 required for NN training is much larger in simulation study than in experimental test in this paper.

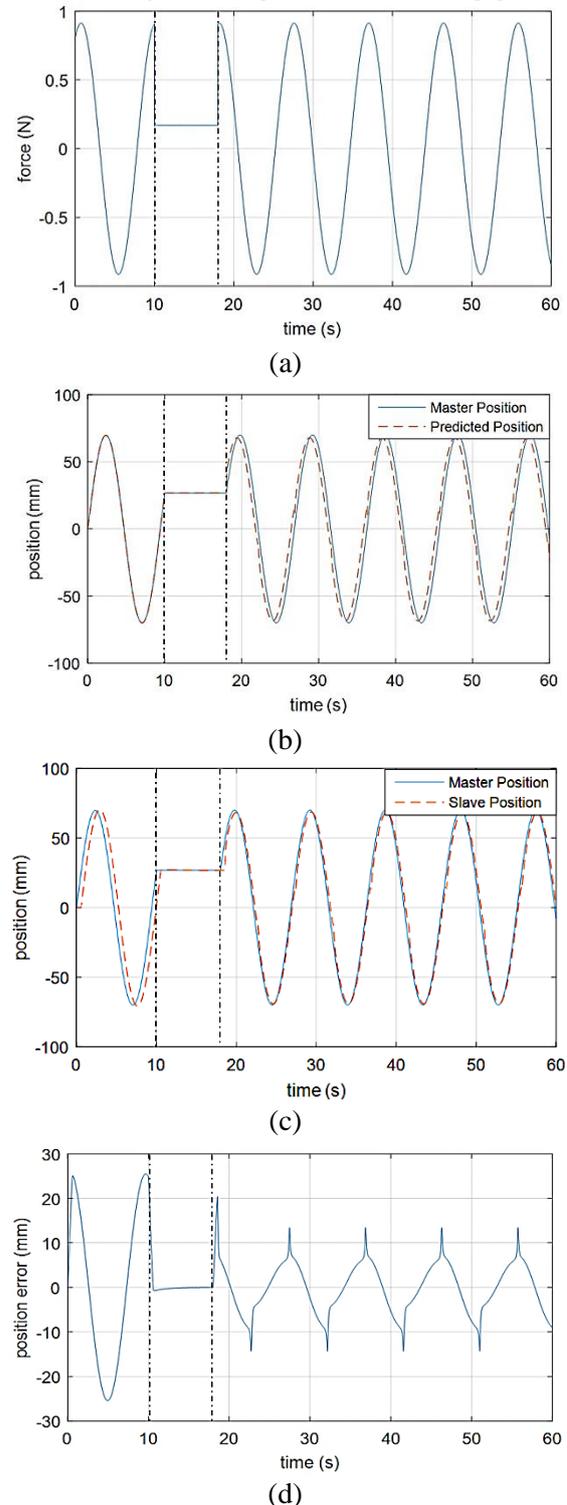


Figure 6. Simulation of Scenario I. (a) Human force. (b) Master position and predicted master position. Note that prediction occurs after T_2 (second dash-dot line). (c) Master and slave positions. (d) Position error. Note that during the whole operation, including T_1+T_2 , the definition for $e^{\wedge(x)}$ is depicted.

Table 1. System and NN parameters which are in common among all simulation and experimental tests.

Parameter	Description	Value	Unit
m	Number of input layer neurons	10	-
s	System sample time	0.01	sec
τ	System time delay	0.5	sec
T_1	Time for gathering NN inputs	10	sec
-	Number of hidden layer neurons for Scenario I	10	-
-	Number of hidden layer neurons for Scenario II	18	-
-	Minimum value of objective function to stop NN training	0.001	-

Table 2. System and NN parameters for both scenarios in simulation study.

Parameter	Description	Value	Unit
η	NN learning rate	0.04	-
T_2	NN training time	8	sec
-	Maximum epochs to stop NN training	4000	-
$k_p^{(x)}$	Proportional PID gain	250	N/m
$k_d^{(x)}$	Derivative PID gain	0	N.s/m
$k_i^{(x)}$	Integral PID gain	155	N/(m.s)

Scenario II: The simulated human force which is applied to the master robot in this scenario is illustrated in Figure 7(a). The results for this scenario are shown in Figures 7(b) to 7(d).

It can be inferred from Figures 7(b) to 7(d) that not only the results of Scenario I are valid for more complex motions, but also using a NN in the system enables it to predict new motions if the amplitude and frequency are not too far from those of the trained motion. In other words, the NN can correctly predict newly intended motions of the operator which had been unknown to the system before their first appearance, particularly in the first phase. As can be seen in Figure 7(b), The NN has performed a good prediction of the master robot position during the whole operation. Therefore, in spite of large amount of time delay in the system, Figures 7(c) and 7(d) show that the slave robot has successfully tracked the master robot position in advance and the system is stable and highly transparent.

5.2. Experimental results

In this section, the end effector of Phantom Omni moves only in x-direction with respect to the reference Cartesian coordinates (Figure 4). Hence, three controllers should be designed for the joints q_1 , q_2 and q_3 .

The online NN training is carried out by incremental-mode method through Fast Artificial Neural Network (FANN) library for C++. A constant time delay τ is considered for the system. The system and NN parameters for these scenarios are summarized in Table 3.

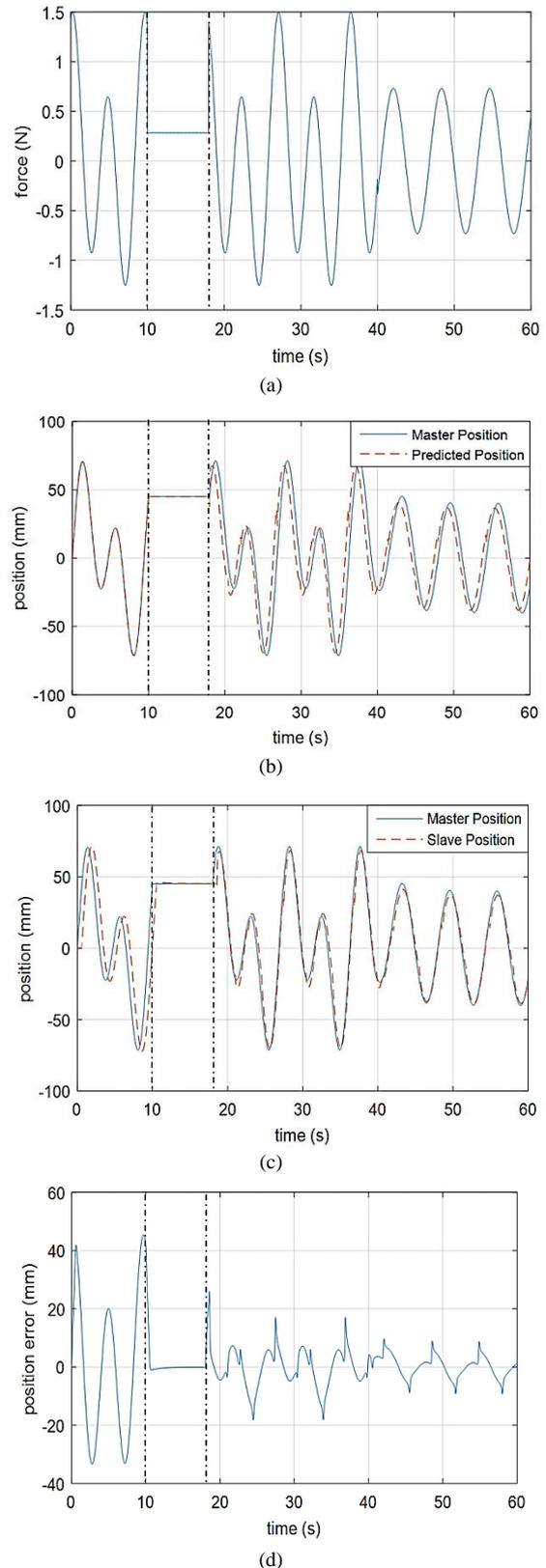


Figure 7. Simulation of Scenario II. (a) Human force. (b) Master position and predicted master position. Note that prediction occurs after T_2 (second dash-dot line). (c) Master and slave positions. (d) Position error. Note that during the whole operation, including T_1+T_2 , the definition for $e^{(x)}$ is depicted.

Table 3. System and NN parameters for both scenarios in experimental test.

Parameter	Description	Value	Unit
η	NN learning rate	0.7	-
T_2	NN training time	4	sec
-	Maximum epochs to stop NN training	2000	-
$k_{p,1}^{(q)}$	Proportional PID gain for q_1	4	N.m/rad
$k_{d,1}^{(q)}$	Derivative PID gain for q_1	0.15	N.m.s/rad
$k_{i,1}^{(q)}$	Integral PID gain for q_1	0.4	N.m/(rad.s)
$k_{p,2}^{(q)}$	Proportional PID gain for q_2	2.2	N.m/rad
$k_{d,2}^{(q)}$	Derivative PID gain for q_2	0.01	N.m.s/rad
$k_{i,2}^{(q)}$	Integral PID gain for q_2	0.06	N.m/(rad.s)
$k_{p,3}^{(q)}$	Proportional PID gain for q_3	2.2	N.m/rad
$k_{d,3}^{(q)}$	Derivative PID gain for q_3	0.01	N.m.s/rad
$k_{i,3}^{(q)}$	Integral PID gain for q_3	0.06	N.m/(rad.s)

Scenario I: The results for this scenario are illustrated in Figures 8(a) to 8(c).

Consistent with the simulation results, Figure 8(a) shows that the motion of the master robot is well predicted by the NN and the slave robot has tracked the master position ahead, which is further demonstrated by Figure 8(b) in which the master and slave robots are shown to be moving simultaneously. High transparency of the system can also be deduced by comparing pre- and post-training intervals in Figures 8(b) and 8(c). Also, to quantitatively prove the stated facts, the root-mean-square error (RMSE) and the normalized root-mean-square error (NRMSE) during the prediction period for this scenario are obtained to be 7.42 mm and 5.57% , respectively. The NRMSE is defined by (5.1), where $\max\{x_m\}$ and $\min\{x_m\}$ denote the maximum and minimum position of the master robot, respectively, during the prediction period.

$$\text{NRMSE} := \frac{\text{RMSE}}{\max\{x_m\} - \min\{x_m\}} \quad (5.1)$$

Meanwhile, as mentioned earlier in section 5-1, the time T_2 in experimental test is seen to be smaller because NN is trained much faster by FANN than in MATLAB environment.

Scenario II: The results for this scenario are depicted in Figures 9(a) to 9(c).

Again, the results in Figures 9(a) to 9(c) show high transparency of the system. Consistent with the results obtained through simulation, Figure 9(a) indicates that the system is capable of predicting both trained and untrained motions of the master robot, despite having a large amount of time delay in the system. Furthermore, online tracking capability of the slave robot can be seen in Figures 9(b) and 9(c), especially by comparing pre- and post-training time intervals. To further prove these facts quantitatively, the RMSE and the NRMSE (as defined by (5.1)) during the prediction period for this scenario are obtained to be 11.71 mm and 7.04% , respectively.

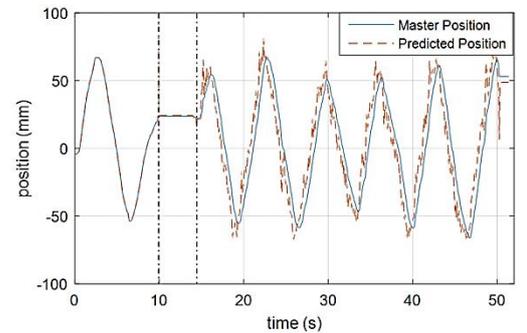
Of course, looking more closely at Figure 9(b), the slave robot is observed not to precisely track the master

robot position at some points, around $t = 28\text{ s}$ for instance. This is because the NN is trained by a small amount of data in comparison with the applied motions after the training. In other words, unlike Scenario I, the slave robot does not have much information about what motion the human operator is going to apply next. Consequently, it may fail to accurately predict the operator motion at sometimes.

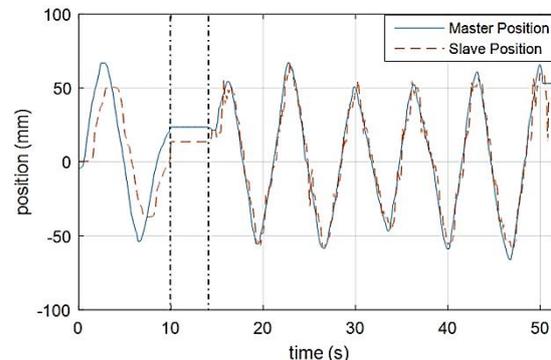
However, figure 9 (b) shows that the variable amplitude and frequency of the applied motion is well predicted despite almost merely one period of input training data to the NN. If more accurate

prediction is desired, obviously more input data should be presented to the NN which, in turn, takes more time and may seem a negative point to the operator.

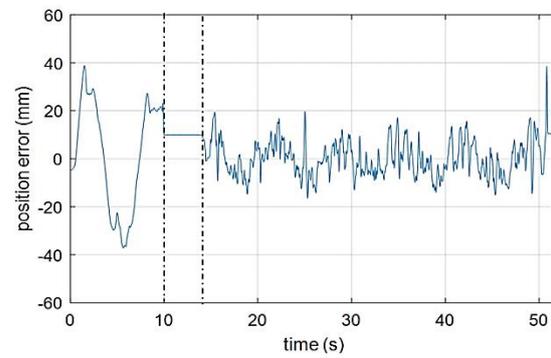
Simulation and experimental results presented in this section also show that both batch-mode and incremental-mode trainings have yielded reliable results.



(a)



(b)



(c)

Figure 8. Experiment of Scenario I. (a) Master position and predicted master position. Note that prediction occurs after T_2 (second dash-dot line). (b) Master and slave positions. (c) Position error. Note that during the whole operation, including T_1+T_2 , the definition for $e^{\wedge(x)}$ is depicted.

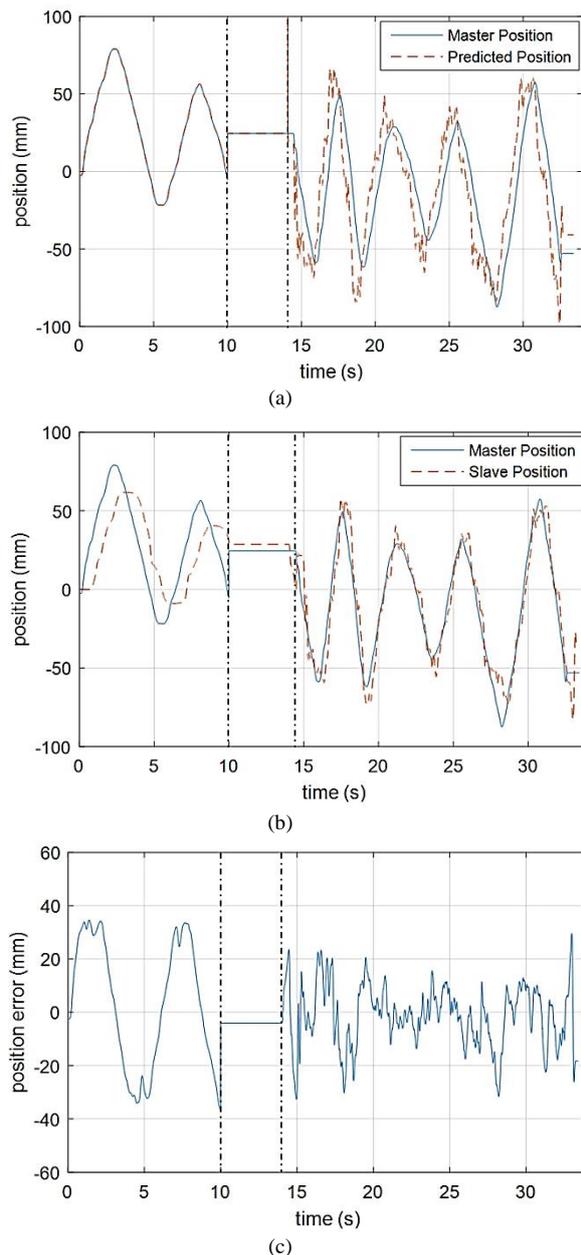


Figure 9. Experiment of Scenario II. (a) Master position and predicted master position. Note that prediction occurs after T2 (second dash-dot line). (b) Master and slave positions. (c) Position error. Note that during the whole operation, including T1+T2, the definition for $e^{\Lambda(s)}$ is depicted.

6. Conclusion and future works

In this paper, a neural network was incorporated on master side in a unilateral teleoperation system in order to predict the future position of the operator's hand online. The discussion on the superiority of the presented method in comparison with previously proposed methods in the literature was provided, too.

The simulation and experimental results indicated that the teleoperation system performs with high transparency, which means the slave robot can well track the master robot motions without delay. Meanwhile, the NN predictor can predict not only the trained motion, but also untrained motions.

Possible future directions include considering stochastic time delay in communication channel, taking environment force into account and reducing the time of motion halt during training phase.

References

- [1] R. Uddin and J. Ryu, "Predictive control approaches for bilateral teleoperation". *Annual Reviews in Control*, Vol. 42, 2016, 82-99.
- [2] C. Passenberg, A. Peer, and M. Buss, "A survey of environment-, operator-, and task-adapted controllers for teleoperation systems". *Mechatronics*, Vol. 20, No. 7, 2010, 787-801.
- [3] X. Xu, B. Cizmeci, C. Schuwerk, and E. Steinbach, "Model-Mediated Teleoperation: Toward Stable and Transparent Teleoperation Systems". *IEEE Access*, Vol. 4, 2016, 425-449.
- [4] B. Yazdankhoo and B. Beigzadeh, "Increasing stability in model-mediated teleoperation approach by reducing model jump effect". *Scientia Iranica*, Vol. 26, Special Issue on: Socio-Cognitive Engineering, 2019, 3-14.
- [5] P. Mitra and G. Niemeyer, "Model-mediated telemanipulation". *The International Journal of Robotics Research*, Vol. 27, No. 2, 2008, 253-262.
- [6] N. Jarrassé, J. Paik, V. Pasqui, and G. Morel, "How can human motion prediction increase transparency?". *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, USA, 2008, 2134-2139.
- [7] R. Uddin, S. Park, and J. Ryu, "A predictive energy-bounding approach for Haptic teleoperation". *Mechatronics*, Vol. 35, 2016, 148-161.
- [8] S. E. Engelbrecht, "Minimum principles in motor control". *Journal of Mathematical Psychology*, Vol. 45, No. 3, 2001, 497-542.
- [9] C. Smith and P. Jensfelt, "A predictor for operator input for time-delayed teleoperation". *Mechatronics*, Vol. 20, No. 7, 2010, 778-786.
- [10] D. Feth, A. Peer, and M. Buss, "Enhancement of multi-user teleoperation systems by prediction of dyadic haptic interaction". *Experimental Robotics*, 2014, 855-869.
- [11] P. Prekopiou, S. G. Tzafestas, and W. S. Harwin, "Towards variable-time-delays-robust telemanipulation through master state prediction". *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Atlanta, USA, 1999, 305-310.
- [12] S. Clarke, G. Schillhuber, M. F. Zaeh, and H. Ulbrich, "Prediction-based methods for teleoperation across delayed networks". *Multimedia Systems*, Vol. 13, No. 4, 2008, 253-261.
- [13] F. Stakem and G. AlRegib, "An adaptive approach to exponential smoothing for CVE state prediction". *2nd International Conference on Immersive Telecommunications*, Berkley, USA, 2009, 141-146.
- [14] N. Sakr, N. D. Georganas, J. Zhao, and X. Shen, "Motion and force prediction in haptic media". *IEEE International Conference on Multimedia and Expo*, Beijing, China, 2007, 2242-2245.
- [15] A. Pentland and A. Liu, "Modeling and prediction of human behavior". *Neural computation*, Vol. 11, No. 1, 1999, 229-242.
- [16] S. Shen, A. Song, T. Li, and H. Li, "Time delay compensation for nonlinear bilateral teleoperation: A motion prediction approach". *Transactions of the Institute of Measurement and Control*, 2019.
- [17] E. S. Gardner Jr, "Exponential smoothing: The state of the art—Part II". *International journal of forecasting*, Vol. 22, No. 4, 2006, 637-666.

- [18] Y. Ji, D. Liu, and Y. Guo, "Adaptive neural network based position tracking control for Dual-master/Single-slave teleoperation system under communication constant time delays". *ISA transactions*, Vol. 93, 2019, 80-92.
- [19] P. M. Kebria, A. Khosravi, S. Nahavandi, D. Wu, and F. Bello, "Adaptive Type-2 Fuzzy Neural-Network Control for Teleoperation Systems with Delay and Uncertainties". *IEEE Transactions on Fuzzy Systems*, 2019.
- [20] C.-C. Hua, Y. Yang, and X. Guan, "Neural network-based adaptive position tracking control for bilateral teleoperation under constant time delay". *Neurocomputing*, Vol. 113, 2013, 204-212.
- [21] J. Nava and V. Kreinovich, "Why A Model Produced by Training a Neural Network Is Often More Computationally Efficient than a Nonlinear Regression Model: A Theoretical Explanation". *Journal of Uncertain Systems*, Vol. 8, No. 3, 2014, 193-204.
- [22] V. Nicolau, V. Palade, D. Aiordachioaie, and C. Miholca, "Neural network prediction of the roll motion of a ship for intelligent course control". *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Vietri sul Mare, Italy, 2007, 284-291.
- [23] G.-Z. Yang et al., "The grand challenges of Science Robotics". *Science Robotics*, Vol. 3, No. 14, 2018, eaar7650.
- [24] K. Mehrotra, C. K. Mohan, and S. Ranka, *Elements of artificial neural networks*. Cambridge: MIT press; 2000.
- [25] T. Sansanayuth, I. Nilkhamhang, and K. Tungpimolrat, "Teleoperation with inverse dynamics control for phantom omni haptic device". *SICE Annual Conference*, Akita, Japan, 2012, 2121-2126.
- [26] F. Llewellyn, "Some fundamental properties of transmission systems". *Proceedings of the IRE*, Vol. 40, No. 3, 1952, 271-283.
- [27] D. Lee and M. W. Spong, "Passive bilateral teleoperation with constant time delay". *IEEE transactions on robotics*, Vol. 22, No. 2, 2006, 269-281.
- [28] X. Liu and X. Dong, "4-Channel Sliding Mode Control of Teleoperation Systems with Disturbances". *Asian Journal of Control*, Vol. 17, No. 4, 2014, 1267-1273.
- [29] H. C. Cho and J. H. Park, "Impedance control with variable damping for bilateral teleoperation under time delay". *JSME International Journal Series C Mechanical Systems, Machine Elements and Manufacturing*, Vol. 48, No. 4, 2005, 695-703.
- [30] L.-G. García-Valdovinos, V. Parra-Vega, and M. A. Arteaga, "Observer-based sliding mode impedance control of bilateral teleoperation under constant unknown time delay". *Robotics and Autonomous Systems*, Vol. 55, No. 8, 2007, 609-617.
- [31] T. Hilliard and Y.-J. Pan, "Stabilization of asymmetric bilateral teleoperation systems for haptic devices with time-varying delays". *American Control Conference*, Washington, DC, USA, 2013, 4538-4543.