IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Seattle, USA, June 2023

# Neural Network Learning of Robot Dynamic Uncertainties and Observer-based External Disturbance Estimation for Impedance Control*

Teng Li[1], Armin Badre[2], Hamid D. Taghirad[3], and Mahdi Tavakoli[1], *Senior Member, IEEE*

***Abstract*— Estimation of dynamic uncertainties is a critical and fundamental problem when designing a control system for a robot. During robot-environment interaction, in addition to the internal dynamic model uncertainties, the external environment-exerted force will also enter the dynamics. For robot impedance control, an exact dynamic model of the robot is needed but usually not available. It has been shown that integrating an impedance controller with a disturbance observer can achieve accurate impedance control. However, it works only for robots in free motion but not robot-environment interaction. Although a disturbance observer is able to accurately estimate the dynamic uncertainties, the estimation is lumped uncertainties that contain all uncertainty sources including both the internal and the external disturbances. Without separating these two parts, the method of combining an impedance controller and an observer will result in the human-applied force being canceled instead of interacting with the robot. To solve this problem in this paper, we propose a framework for learning the internal disturbances and separating the external disturbances by integrating three entities: an impedance controller, a neural network (NN) model, and a disturbance observer. In the framework, the impedance controller provides compliant robot behavior, while the observer captures the lumped uncertainties, and the NN learns to separate the external disturbances. Simulation results of an application scenario with an obstructive virtual fixture demonstrate the effectiveness of the proposed framework.**

## I. Introduction

In the field of robot control, having an accurate dynamic model of a robot is fundamental for ensuring an accurate and stable control [1]. This is especially the case in medical robotic systems where accuracy and safety are overriding concerns. However, accurate dynamic models only exist in theory but not in practice, which means that dynamic uncertainties in a robotic system are inevitable. Common dynamic uncertainties include joint friction, inaccurate center of mass location and link weight, and extra payloads attached to the robot body [1]. Therefore, estimation and compensation of dynamic uncertainties are critical in robot control.

Various observers have been developed to estimate and thus compensate for dynamic uncertainties [2]–[5]. Typical approaches include generalized momentum observer (GMO) [5], extended state observer (ESO) [6], nonlinear disturbance observer (NDOB) [7], disturbance Kalman filter (DKF) method [8], [9], as well as their variations.

GMO, also known as classic first-order momentum observer, is one of the most commonly used observers due to its advantages of being simple and easy to implement [10]. It is often used as a reference when designing new observers [8], [9], [11]. ESO was originally proposed by Han in 1995 [12], [13]. Since then, many of its variations have been developed for different purposes such as collision detection [14] and interaction force estimation [6]. NDOB is designed specifically by considering the nonlinearity of the dynamics of robots which enabled it to have an advantage over the linear ones [7], [15], [16]. DKF is another novel method to estimate the dynamic uncertainties in the control system [8], [17]. The accuracy of its estimation is excellent, but the complexity of implementation could be a limitation for it to be widely used.

All of these observers estimate lumped uncertainties. While the lumped uncertainties do include various components (*e.g.*, joint friction), it is not possible to separate that specific component out of the lumped estimate. Especially in human-robot interaction scenarios, the observer will take the human-exerted force as a part of uncertainties and thus reject it [18]. On the other hand, estimating contact force between human-robot or robot-environment is a critical problem in the field of human-robot interaction and the field of collision detection [5]. Without separating the human-applied force out of the lumped estimate, a disturbance observer will cancel the human-applied force and thus reject human-robot interaction. Finding a solution to this issue is important for robots with impedance control especially in medical robotic applications involving human-robot interaction.

Some methods have been developed to estimate the robot-environment interaction force by involving learning techniques. Hu and Xiong [8] developed a method to estimate external contact force using a semiparametric model and

*This research is supported in part by the Canada Foundation for Innovation (CFI) under grants LOF 28241 and JELF 35916, in part by the Government of Alberta under grants IAE RCP-12-021 and EDT RCP-17-019-SEG, in part by the Government of Alberta's grant to Centre for Autonomous Systems in Strengthening Future Communities (RCP-19-001-MIF), in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grants RTI-2018-00681, RGPIN-2019-04662, and RGPAS-2019-00106.

[1] T. Li and M. Tavakoli are with the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Alberta, Edmonton T6G 1H9, Alberta, Canada. E-mail: {teng4, mahdi.tavakoli}@ualberta.ca

[2] A. Badre is with the Western Upper Limb Facility, Sturgeon Hospital, St. Albert, Alberta, Canada, and the Division of Orthopaedic Surgery, Department of Surgery, Faculty of Medicine & Dentistry, University of Alberta, Edmonton, Alberta, Canada. E-mail: badre@ualberta.ca

[3] H. D. Taghirad is with the Advanced Robotics and Automated Systems (ARAS), Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran. E-mail: taghirad@kntu.ac.ir

DKF. In their method, the semiparametric dynamic model containing a multilayer perceptron (MLP) neural network was used to provide a more accurate dynamic model, while a DKF was used to estimate the contact force between the robot and the environment. Similarly, an integrated framework of neural network (NN) and DKF was developed to estimate external contact force [9]. In the framework, an NN model was used to learn the joint friction, while a DKF observer was used to estimate the contact force. In another work, NN was used to approximate the global friction, while a momentum observer was used to estimate the external force. Additionally, a Kalman filter was employed to filter the measurement noise for a more accurate force estimation.

Sharifi *et al.* [19] employed a nonlinear autoregressive network with exogenous inputs (NARX) to learn and estimate the robot dynamics plus the passive dynamics of a user who wore an exoskeleton. Then the human-exoskeleton interaction force was separated out by subtracting the NARX-learned dynamics from the motor torques in the scenario of active user dynamics. Note that the movement trajectories of the exoskeleton is nearly periodic, which helped with training the NARX model to generate its model predictions.

Inspired by these methods, in this paper we propose a framework for training a NN model to learn the robot dynamic uncertainties so that we can later isolate the external disturbances. Specifically, the framework involves three entities, *i.e.*, an impedance controller, an NDOB, and an NN model, where the impedance controller provides compliance, the NDOB measures dynamic uncertainties plus external disturbance (*i.e.*, the lumped uncertainties), and the NN estimates the dynamic uncertainties. Thus, by subtracting the NN output from the NDOB output, the external disturbance (*e.g.*, human-applied force) can be isolated. A simulation of an application scenario is conducted to evaluate the effectiveness of the framework.

The remaining parts of this paper are organized as follows. Section II describes impedance control, NDOB, and NN model, as well as detailed procedures of the proposed framework. Section III presents simulations, validations, and corresponding results related to implementing the framework. Section IV summarizes and remarks on the conclusions.

## II. Methods

### A. Robot dynamics and impedance control

A general dynamic model for an $n$-degree-of-freedom (DOF) rigid robot [20] can be given by

$$\underbrace{\mathbf{M}(\mathbf{q})}_{\hat{\mathbf{M}}+\Delta\mathbf{M}}\ddot{\mathbf{q}}+\underbrace{\mathbf{S}(\mathbf{q},\dot{\mathbf{q}})}_{\hat{\mathbf{S}}+\Delta\mathbf{S}}\dot{\mathbf{q}}+\underbrace{\mathbf{g}(\mathbf{q})}_{\hat{\mathbf{g}}+\Delta\mathbf{g}}+\tau_{\mathbf{fric}}(\mathbf{q},\dot{\mathbf{q}})=\tau+\underbrace{\tau_{\mathbf{ext}}}_{\mathbf{J^T F_{ext}}} \tag{1}$$

where $\mathbf{M}\in\mathbb{R}^{n\times n}$ denotes the inherent inertia matrix, $\mathbf{S}\in\mathbb{R}^{n\times n}$ denotes a matrix of the Coriolis and centrifugal forces, $\mathbf{g}\in\mathbb{R}^{n}$ represents the gravity vector. $\hat{\mathbf{M}},\hat{\mathbf{S}},\hat{\mathbf{g}}$ represent the user's model estimates, while $\Delta\mathbf{M},\Delta\mathbf{S},\Delta\mathbf{g}$ are the corresponding estimate errors. $\tau_{\mathbf{fric}}\in\mathbb{R}^{n}$ is joint friction, $\tau\in\mathbb{R}^{n}$ is the commanded joint torque vector, $\tau_{\mathbf{ext}}\in\mathbb{R}^{n}$ is the torque caused by external force, $\mathbf{F_{ext}}\in\mathbb{R}^{6}$ is the external force in Cartesian space, and $\mathbf{J}\in\mathbb{R}^{6\times n}$ is the Jacobian matrix.

A desired impedance model [18], [21], [22] for robot-environment contact can be expressed as

$$\mathbf{F_{imp}}=\mathbf{M_m}(\ddot{\mathbf{x}}-\ddot{\mathbf{x}}_\mathbf{d})+(\mathbf{S_x}+\mathbf{D_m})(\dot{\mathbf{x}}-\dot{\mathbf{x}}_\mathbf{d})+\mathbf{K_m}(\mathbf{x}-\mathbf{x_d}) \tag{2}$$

where $\mathbf{M_m},\mathbf{D_m},\mathbf{K_m}$ are user-designed matrices for inertia, damping, and stiffness, respectively. Note that $\mathbf{x_d},\dot{\mathbf{x}}_\mathbf{d},\ddot{\mathbf{x}}_\mathbf{d}$ are the desired position, velocity, and acceleration, respectively in Cartesian space, while $\mathbf{x},\dot{\mathbf{x}},\ddot{\mathbf{x}}$ are the actual ones. $\mathbf{S_x}$ is the Coriolis and centrifugal matrix of the robot in Cartesian space and $\mathbf{S_x}=\mathbf{J^{-T}SJ^{-1}}-\mathbf{M_x}\dot{\mathbf{J}}\mathbf{J^{-1}}$, where $\mathbf{M_x}=\mathbf{J^{-T}MJ^{-1}}$ is the inherent inertia of the robot in Cartesian space [23].

To avoid the need for the measurement of external forces, the designed inertia matrix can be set as $\mathbf{M_m}=\mathbf{M_x}$. Then, to reach (2) as the closed-loop dynamics governing the robot-environment interaction ($\mathbf{F_{ext}}=\mathbf{F_{imp}}$) in an ideal scenario of no model errors and no joint friction, the impedance control law can be given by [18]

$$\begin{aligned}\tau=\mathbf{MJ^{-1}}(\ddot{\mathbf{x}}_\mathbf{d}-\dot{\mathbf{J}}\mathbf{J^{-1}}\dot{\mathbf{x}}_\mathbf{d})+\mathbf{SJ^{-1}}\dot{\mathbf{x}}_\mathbf{d}+\mathbf{g}\\+\mathbf{J^T}[\mathbf{D_m}(\dot{\mathbf{x}}_\mathbf{d}-\dot{\mathbf{x}})+\mathbf{K_m}(\mathbf{x_d}-\mathbf{x})]\end{aligned} \tag{3}$$

Note that when implementing the impedance controller (3) in practice, the estimates $\hat{\mathbf{M}},\hat{\mathbf{S}},\hat{\mathbf{g}}$ will be used since the accurate model of the robot is usually not accessible.

For robot impedance control around a fixed point in space, *i.e.*, set-point regulation, it has $\ddot{\mathbf{x}}_\mathbf{d}=\mathbf{0}$, $\dot{\mathbf{x}}_\mathbf{d}=\mathbf{0}$. Then, the impedance control law (3) can be simplified to (4), which is also known as task-space proportional–derivative (PD) controller with gravity compensation.

$$\tau=\mathbf{J^T}[\mathbf{K_m}(\mathbf{x_d}-\mathbf{x})-\mathbf{D_m}\dot{\mathbf{x}}]+\mathbf{g} \tag{4}$$

### B. Friction model

The joint friction in this work is modeled by the Stribeck model [7], [9] as expressed by (5), which is viewed as the most classical nonlinear expression that can include many characteristics of friction.

$$\tau_{\mathbf{fric}}=\mathbf{F_c}sign(\dot{\mathbf{q}})+(\mathbf{F_s}-\mathbf{F_c})sign(\dot{\mathbf{q}})e^{-|\dot{\mathbf{q}}/\nu_\mathbf{s}|^\sigma}+\mathbf{F_v}\dot{\mathbf{q}} \tag{5}$$

where $\mathbf{F_c}$ is the Coulomb friction, $\mathbf{F_s}$ is the static friction, $\mathbf{F_v}$ is the viscous friction, $\dot{\mathbf{q}}$ is the joint velocity, $\nu_\mathbf{s}$ is the Stribeck parameter, $\sigma$ is the exponent of the Stribeck nonlinearity and $\sigma=2$ is employed in this work.

### C. Disturbance observer

In order to organize all the disturbances together, the dynamic model (1) of a robot can be re-written as

$$\hat{\mathbf{M}}\ddot{\mathbf{q}}+\hat{\mathbf{S}}\dot{\mathbf{q}}+\hat{\mathbf{g}}=\tau+\underbrace{\tau_{\mathbf{ext}}-[\tau_{\mathbf{fric}}+(\Delta\mathbf{M}\ddot{\mathbf{q}}+\Delta\mathbf{S}\dot{\mathbf{q}}+\Delta\mathbf{g})]}_{\tau_{\mathbf{dist}}} \tag{6}$$

where $\tau_{\mathbf{dist}}$ denotes the lumped uncertainties that usually include three main aspects, *i.e.*, (source-1) the model error $(\Delta\mathbf{M}\ddot{\mathbf{q}}+\Delta\mathbf{S}\dot{\mathbf{q}}+\Delta\mathbf{g})$, (source-2) the joint friction $\tau_{\mathbf{fric}}$, and the external disturbances $\tau_{\mathbf{ext}}$ which may involve (source-3) constant disturbance and (source-4) time-varying disturbance. The constant disturbance may be a constant payload attached to the robot end-effector (EE) or body, while time-varying disturbance may be robot-environment interaction forces such as human-applied forces during human-robot interaction. In this paper, internal disturbances refer to the summed uncertainties from source-1,2, while external disturbances refer to the summed uncertainties from source-3,4. Disturbance observer is a commonly used tool to estimate the lumped uncertainties. Note that any other uncertainties beyond the four sources will also be included in the lumped uncertainties that are estimated by the observer.

NDOB is easy to be implemented and has the advantage of estimating the nonlinearities [7]. Therefore, NDOB is employed in this work to estimate the lumped uncertainties. An adapted NDOB design based on [7] is used in this paper which can be expressed as

$$\begin{cases} \mathbf{L} = \mathbf{Y}\hat{\mathbf{M}}^{-1} \\ \mathbf{p} = \mathbf{Y}\dot{\mathbf{q}} \\ \dot{\mathbf{z}} = -\mathbf{L}\mathbf{z} + \mathbf{L}(\hat{\mathbf{S}}\dot{\mathbf{q}} + \hat{\mathbf{g}} - \tau - \mathbf{p}) \\ \tau_{\mathtt{NDOB}} = \mathbf{z} + \mathbf{p} \end{cases} \tag{7}$$

where $\mathbf{L} \in \mathbb{R}^{n\times n}$ is the observer gain matrix, $\mathbf{Y} \in \mathbb{R}^{n\times n}$ is a constant invertible matrix that needs to be designed, $\hat{\mathbf{M}}$ is designed to be a symmetric and positive definite matrix and thus invertible, $\mathbf{z}$ is an auxiliary variable, $\mathbf{p}$ is an auxiliary vector determined from $\mathbf{Y}$, $\tau_{\mathtt{NDOB}}$ is the estimated lumped uncertainties via the NDOB observer, *i.e.*, $\hat{\tau}_{\mathbf{dist}} = \tau_{\mathtt{NDOB}}$.

### D. Neural network

A promising neural network (NN) structure for modeling nonlinear dynamic systems is the nonlinear autoregressive network with exogenous inputs (NARX), which is a recurrent dynamic network with a special feature of feeding its delayed output back as input. This special feature requires the NARX to be trained by using time-series data, which means that the training data set needs to be collected from a long period of one single continuous task, while concatenating datasets collected from a set of separate short-period tasks is usually not acceptable. This is because the learning effect will be distorted by the outliers occurring at the concatenating points. A common feed-forward neural network (fNN) does not have this requirement.

Considering that machine learning usually requires a large amount of data for training, while a single exciting robot trajectory over a long continuous period of time may not be easy to generate. Therefore, a fNN model as shown in Fig. 1 is employed. For the fNN model, the training data can be collected from a set of separate short-period trajectory tracking tasks. Also, one of its variations, cascade-forward neural network (cNN), is used to compare the learning performance. The fNN and cNN have the same structure except that the cNN includes an additional connection from its input layer to each of the following layers.

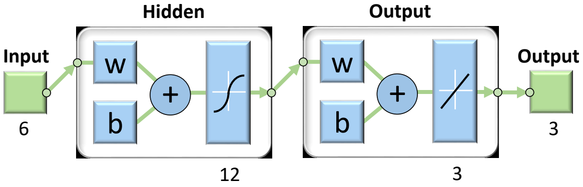


Fig. 1: The structure of feed-forward neural network (fNN) model.

### E. Collecting training and testing data

For training the fNN model (Fig. 1), the robot EE position $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$ will be taken as the inputs, while the estimate of lumped uncertainties from the NDOB will be taken as the learning targets. Since exciting trajectories are often employed for robot dynamic identification [1], the training data will be collected in a set of separate exciting trajectory tracking tasks. In the tasks, the robot is controlled by an impedance controller (3) and NDOB observer (7), a simple addition of the two equations when doing the implementation [18].

First of all, a set of exciting trajectories are generated which include joint position $\mathbf{q}$, velocity $\dot{\mathbf{q}}$, and acceleration $\ddot{\mathbf{q}}$ with a sampling rate of $1,000$ Hz. Each exciting trajectory is generated from a harmonic function with random parameters and lasts 20 seconds, thus every exciting trajectory is unique. In total, 39 exciting trajectories are generated among which 38 of them will be used for training while another for testing.

As mentioned earlier, the lumped uncertainties in this work will come from four sources, which can be categorized as internal disturbances from source-1 (model error) and source-2 (friction), and external disturbances from source-3 (constant payload) and source-4 (time-varying payload). Since this work focuses on simulations, all the uncertainties are fully known and can be precisely controlled.

In source-1, dynamic model error is controlled to be at four different levels, *i.e.*, 0%, 10%, 50%, 100%. The level of model error is tuned by value assignment on matrix $\hat{\mathbf{S}}$ and $\hat{\mathbf{g}}$, while the inertia matrix $\hat{\mathbf{M}}$ is fixed at $\hat{\mathbf{M}} = diag(0.001, 0.001, 0.001)$. For example, model error $10\%$ means $\hat{\mathbf{S}} = 90\%\mathbf{S}$, $\hat{\mathbf{g}} = 90\%\mathbf{g}$, where $\hat{\mathbf{M}}$, $\hat{\mathbf{S}}$, $\hat{\mathbf{g}}$ will be used for the calculation in the impedance controller (3) and NDOB observer (7). Model error $100\%$ means $\hat{\mathbf{S}} = \mathbf{0}$, $\hat{\mathbf{g}} = \mathbf{0}$.

In source-2, joint friction is controlled by (5). In source-3, a constant payload of 22 gram is controlled to be attached to the robot EE or not. In source-4, a time-varying payload is controlled by a set of Fourier series functions given by

$$\begin{cases} F_x = a_1 \sin(\frac{2\pi n_1}{t_1} t) + a_2 \sin(\frac{2\pi n_2}{t_1} t - \phi_2) + a_3 \sin(\frac{2\pi n_3}{t_1} t - \phi_3) \\ F_y = a_1 \cos(\frac{2\pi n_1}{t_1} t) + a_2 \cos(\frac{2\pi n_2}{t_1} t - \phi_2) + a_3 \cos(\frac{2\pi n_3}{t_1} t - \phi_3) \\ F_z = a_1 \cos(\frac{2\pi n_1}{t_1} t) + a_2 \sin(\frac{2\pi n_2}{t_1} t - \phi_2) + a_3 \cos(\frac{2\pi n_3}{t_1} t - \phi_3) \end{cases} \tag{8}$$

where $F_x, F_y, F_z$ are the time-varying payload expressed in Cartesian space, and $t_1 = 10$, $a_1 = 0.1$, $a_2 = 0.15$, $a_3 = 0.05$, $n_1 = 1$, $n_2 = 2$, $n_3 = 3$, $\phi_2 = \frac{1}{2}\pi$, $\phi_3 = \pi$.

In our previous work [18], it has been shown that by integrating an impedance controller and an observer, an accurate impedance control can be achieved in a trajectory tracking task when the actual velocity and acceleration converge to the desired ones. It is noteworthy that the disturbance tracking of the observer works independently from the trajectory tracking of the controller, which means that the observer estimation accuracy is not affected by the trajectory tracking accuracy. In other words, even if the trajectory tracking performance is poor (*e.g.*, due to fully or partially uncompensated disturbances), the observer can still accurately estimate the lumped uncertainties.

The training data are collected by running a trajectory tracking task on each of the 38 exciting trajectories with only internal disturbances (from source-1,2) involved. The testing data are collected by running the trajectory tracking task on a simple figure-eight trajectory as given by (9) and a new exciting trajectory with all of the internal disturbances (from source-1,2) and external disturbances (from source-3,4) involved. In the tasks both of collecting training and testing data, the robot is controlled by an impedance controller (3) and NDOB observer (7), where the former is used to accurately track the trajectory while the latter is used to estimate the lumped uncertainties and compensate them in the controller.

$$\begin{cases} x_d = R\sin(\frac{2\pi}{t_1}t)\cos(\frac{2\pi}{t_1}t) \\ y_d = R\sin(\frac{2\pi}{t_1}t) + R \\ z_d = 0 \end{cases} \tag{9}$$

where $R = 0.02$ m is the amplitude of the figure-eight trajectory, $t_1 = 5$ s is the period for generating a full cycle.

The fNN model is first trained on the training dataset and then tested on the two testing datasets individually. The training goal is that the fNN is expected to learn to estimate only internal disturbances (from source-1,2), thus the external disturbances (from source-3,4) can be separated out by subtracting the fNN prediction outcome from the lumped uncertainties estimated by the NDOB. Note that the NDOB observer can always be implemented to online estimate the lumped uncertainties but not necessarily to do the compensation in the controller. The procedures of the proposed learning framework are illustrated in Fig. 2.

## III. Simulations, Validations, and Results

### A. Robotic system

A 3-DOF PHANToM Premium 1.5A robot (3D Systems, Inc., Cary, NC, USA) is used for simulations in this paper. To build a virtual model of this robot, the kinematic model and dynamic model of the PHANToM robot are reconstructed based on [24]. All the simulations are conducted by using MATLAB/Simulink (version R2020a, MathWorks Inc., Natick, MA, USA), which is running on a computer with a 3.70 GHz Intel(R) Core(TM) i5-9600K CPU and a Windows 10 Education 64-bit operating system. The control rate of the virtual robot is set as $1,000$ Hz, while the sampling rate for acquiring training/testing data is 100 Hz. For all the simulations in this section, the parameter assignments used in the impedance control law, the NDOB, and the fNN model are summarized in Table I.

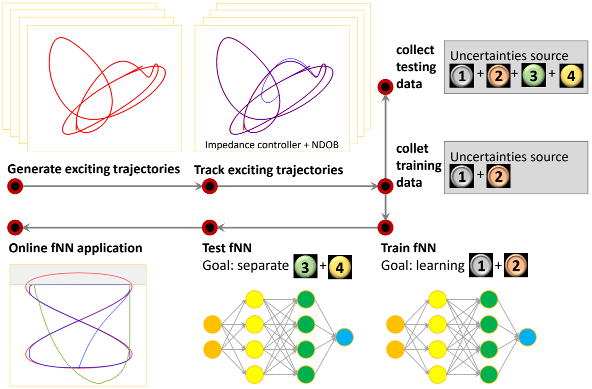


Fig. 2: The procedures of the proposed framework of training fNN to learn internal disturbances and thus separate external disturbances. Uncertainties source-1, model error; source-2, friction; source-3, constant payload; source-4, time-varying payload.

TABLE I: Parameterization for simulations.

| Description | Parameter | Location |
|---|---|---|
| Spring stiffness | $\mathbf{K_m} = \mathbf{7.5I}$ | Eq.(3) |
| Damping | $\mathbf{D_m} = \mathbf{2\sqrt{7.5I}}$ | Eq.(3) |
| Inertia matrix | $\mathbf{\hat{M}} = \mathbf{1.0 \times 10^{-3}} \times \mathbf{I}$ | Eq.(3),(7) |
| Observer gain | $\mathbf{Y} = \mathbf{9.58 \times 10^{-2}} \times \mathbf{I}$ | Eq.(7) |
| Coulomb friction | $\mathbf{F_c} = [0.0049, 0.0031, 0.001]$ | Eq.(5) |
| Static friction | $\mathbf{F_s} = [0.0035, 0.0028, 0.00165]$ | Eq.(5) |
| Viscous friction | $\mathbf{F_v} = [0.06, 0.048, 0.032]$ | Eq.(5) |
| Stribeck parameter | $\nu_\mathbf{s} = [0.00038, 0.0003, 0.00024]$ | Eq.(5) |
| Neurons | 12 | Fig. 1 |
| Input of fNN | $[x_1, x_2, x_3, \dot{x}_1, \dot{x}_2, \dot{x}_3]'$ | Fig. 1 |
| Output of fNN | $\hat{\tau}_\mathbf{dist}$ | Fig. 1 |
| Training function | Bayesian regularization | Fig. 1 |
| Transfer function | Symmetric sigmoid | Fig. 1 |
| Data division | Random | Fig. 1 |
| Data division ratio | $trainRatio = 0.8, valRatio = 0.2$ | Fig. 1 |

Note: $\mathbf{I} \in \mathbb{R}^{3\times3}$ denote identity matrix.

### B. Comparing observers

The disturbance tracking performance among several typical observers, *i.e.*, NDOB [7], GMO [5], DKF [8], [9], and ESO [6], are qualitatively compared when a disturbance of a controlled constant payload (22 gram) along $y$-axis is tracked. As shown in Fig. 3, all of the observers can quickly and accurately estimate the controlled disturbance, although the ESO needs slightly longer time to track the disturbance while the other three observers have comparable disturbance tracking performance among each other.

Considering that NDOB has the advantage of capturing nonlinearities and is easy to implement, NDOB is selected for all simulations in the remaining part of this paper. We also assume that the NDOB can accurately estimate all the lumped uncertainties (*i.e.*, $\tau_\mathbf{dist} = \tau_\texttt{NDOB}$), and its outputs will be taken as the target values during the later training process.

### C. Comparing NN models

The feed-forward neural network (fNN), cascade-forward neural network (cNN), and NARX model are compared on their estimation accuracy. As introduced earlier, the fNN and cNN have the same structure except that the cNN includes

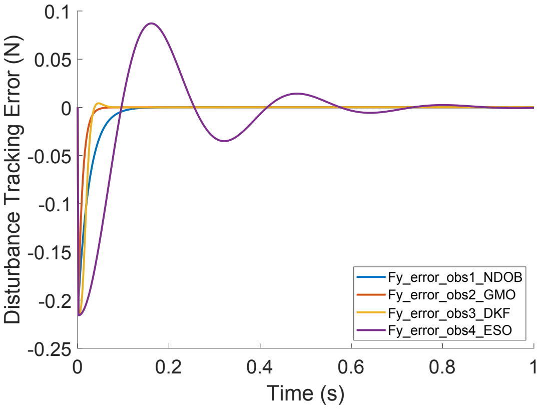
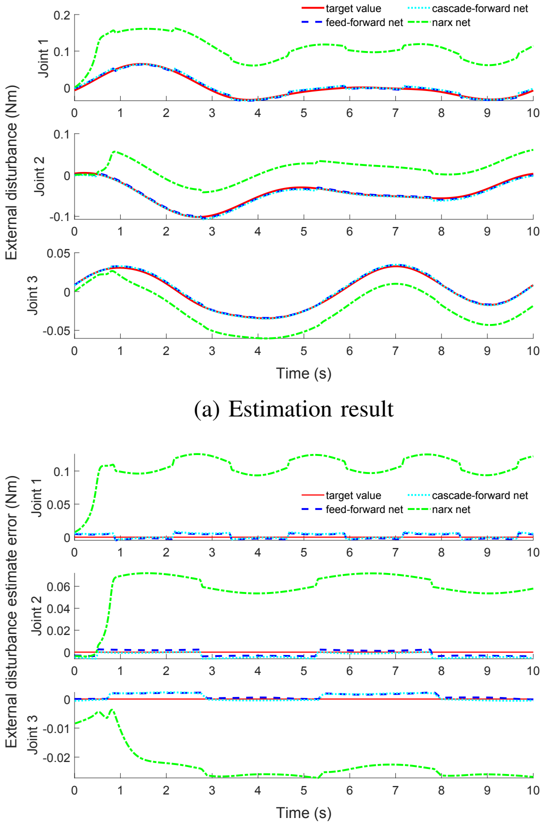


Fig. 3: Comparison on disturbance tracking error of various disturbance observers. The disturbance is a constant payload of 22 gram along the $y$-axis.



(a) Estimation result

(b) Estimation error

Fig. 4: Compare the estimation performance of the feed-forward neural network (fNN), cascade-forward neural network (cNN), and NARX model on external disturbances.

an additional connection from its input layer to each of the following layers. Considering that the learning target in our case is the dynamic model uncertainties rather than the whole dynamic model, only one hidden layer is designed.

For fNN and cNN, as shown in Fig. 1, the input is a 6-by-1 vector of Cartesian position $\mathbf{x}$ and velocity $\dot{\mathbf{x}}$ of the robot EE, and the hidden layer has 12 neurons with Bayesian regularization training function and symmetric sigmoid transfer function. The output layer has 3 neurons with a linear transfer function, and the output is a 3-by-1 vector.

The fNN, cNN, and NARX models are trained separately on data collected from 38 exciting trajectories with each trajectory lasting 20 seconds. The three NN models are trained to learn the uncertainties from source-1,2 by controlling only these two uncertainties sources to exist in the training data. When it comes to the testing process, the three NN models are expected to predict uncertainties only from source-1,2 although uncertainties from all four sources will exist in the testing data. Then, by subtracting the NN prediction output from the NDOB output, the estimation of external disturbances (sum of source-3,4) can be obtained.

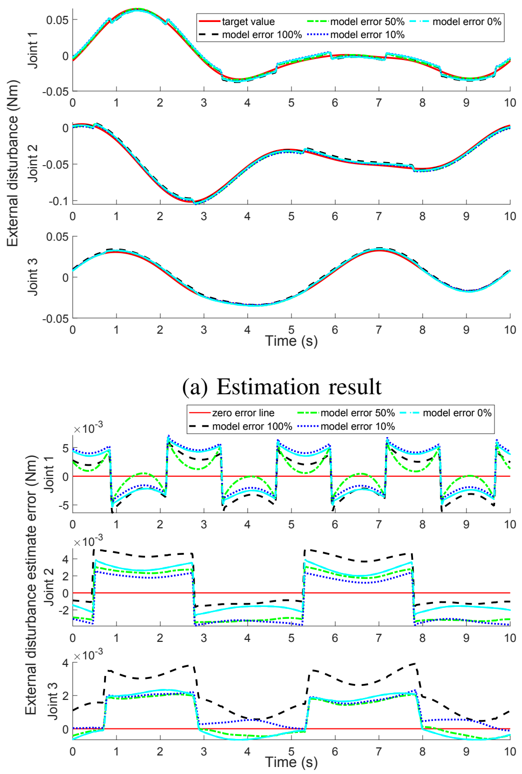
The estimation accuracy of fNN, cNN, and NARX on the external disturbances are compared in Fig. 4. As shown in the figure, there is no significant difference between the fNN and cNN models, but the NARX has a much worse performance. This is because the NARX requires time-series data for its training, and if training data is concatenated from several separate datasets then the training effect will be distorted. Since the fNN model has a common and more compact structure, it is selected for the subsequent simulations.

It is worth mentioning that the learning outcomes of fNN and cNN models are already relatively good when the training data involve 8 exciting trajectories. More exciting trajectories can further improve the learning effect to some extent but at a slow speed, which means that when higher prediction accuracy is not demanding, a relatively small set of training data may meet the requirements.

### D. The effect of dynamic model error

When building a dynamic model of a robot and identifying its dynamic parameters, dynamic model error is inevitable. The effect of dynamic model error (*i.e.*, uncertainties source-1) on the fNN prediction accuracy is explored. More specifically, different levels of model error $(0\%, 10\%, 50\%, 100\%)$ are studied. The level of model error is tuned by value assignment on matrix $\hat{\mathbf{S}}$ and $\hat{\mathbf{g}}$ as introduced earlier.

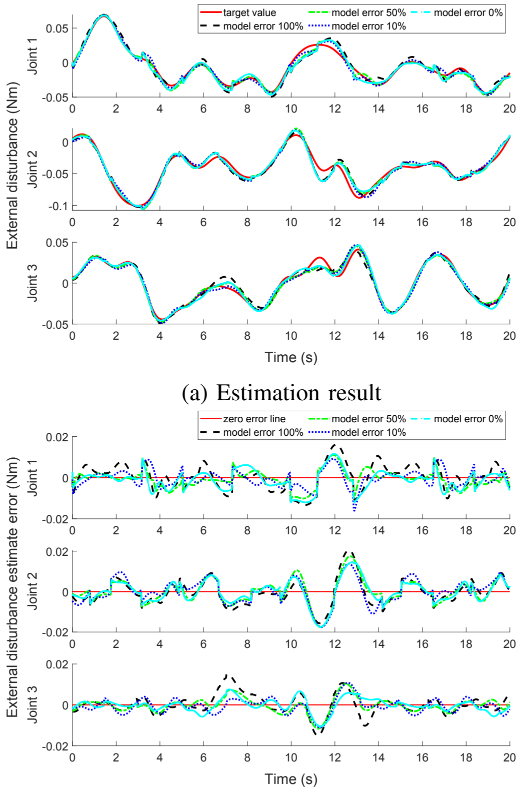
The estimation performance of the fNN model on external disturbances under various levels of model error when tracking a simple figure-eight curve (9) and a complex exciting trajectory curve is shown in Fig. 5a and Fig. 6a, respectively. Correspondingly, the estimation errors are shown in Fig. 5b and Fig. 6b. From the figures, we can see that under all levels of model error except the level of 100%, there is no significant difference between each other. Even with the level of model error 100%, the estimation performance can be acceptable (see the dashed black line in Fig. 5a and Fig. 6a). In general, the effect of model error level on the fNN estimation accuracy is relatively small. The reason could be that for the lightweight robot used in this work, the dynamic model errors are on a relatively small scale. For other heavy industrial robots with dynamic model errors on a large scale, the effect of the model error level needs to be re-evaluated.



(a) Estimation result

(b) Estimation error

Fig. 5: The effect of dynamic model error on fNN estimation performance when tracking a simple figure-eight trajectory.



(a) Estimation result

(b) Estimation error

Fig. 6: The effect of dynamic model error on fNN estimation performance when tracking a complex exciting trajectory.

### *E. Application on virtual fixture*

During a trajectory tracking task, when the robot encounters external disturbances (*e.g.*, an obstacle in the surrounding environment), the NDOB will estimate and compensate for lumped uncertainties including both internal and external disturbances. This may escalate the adverse effect caused by external disturbances. On the other hand, the fNN model can be trained to estimate only internal disturbances, which can make the robot preserve the compliant behavior provided by the impedance controller when external disturbances occur.

In the above sections, the fNN model has been proved to be capable of predicting the total internal disturbances (sum of source-1,2) thus estimating the external disturbances (sum of source-3,4). Here we further explored a potential application of the proposed framework of fNN learning on dynamic model uncertainties.

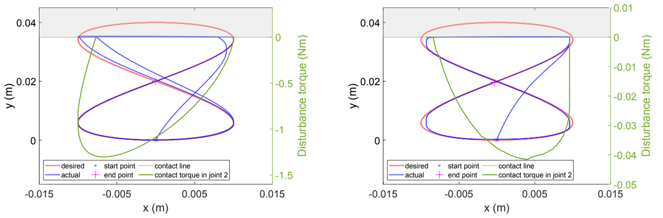
There are two conditions designed in a scenario of a robot encounters a stiff obstacle of virtual fixture (VF) on its way (*i.e.*, external disturbances) when executing a trajectory tracking task. Condition 1 is a scenario with implementing NDOB to estimate and compensate for the lumped uncertainties. Condition 2 is a scenario that is the same as Condition 2 except that the NDOB is replaced with an fNN model that was trained to learn the internal uncertainties at a level of model error $10\%$.

Fig. 7 shows the trajectory tracking task performance in each of the two conditions. In the figure, the gray area represents the obstacle VF which will exert a contact force on the robot EE when they are in contact, while the green line represents the resulting disturbance torque in joint 2 which is obtained by projecting the Cartesian contact force into the robot joint space. Note that the disturbance torques in joint 1 and joint 3 are approximately zero due to the specific task setting and thus ignored.

By comparing Fig. 7a and Fig. 7b, we can see that the contact force in Fig. 7a is extremely large due to the fact that the NDOB escalated the adverse effect of the obstacle VF. Note that if Condition 1 happens in a physical experiment, the robot and/or the obstacle will be badly destroyed once the robot starts to be in contact with the obstacle, thus Condition 1 should be avoided in physical experiments. In Condition 2 as shown in Fig. 7b, the robot is much more compliant with a smaller contact force than that in Condition 1. This is because the fNN in Condition 2, as it was trained in the training process, only estimates the internal uncertainties excluding the external ones caused by the VF. This verified the effectiveness of the trained fNN model.

### *F. Limitations*

Compared to prior work [8], [9], an advantage of the proposed learning framework is that, the fNN model is not restricted to learning only friction model or only model error



(a) Condition 1

(b) Condition 2

Fig. 7: Simulation result of an application scenario on an obstacle of the virtual fixture. Condition 1, NDOB only; Condition 2, fNN only.

which makes it more robust. In fact, it can learn all the uncertainties that exist in the training data, thus separating the later-coming uncertainties in the testing/application scenario. For example, if the fNN model is trained to learn uncertainties from source-1,2,3, then it is able to separate the uncertainties of source-4 when all four sources of uncertainties exist in the application scenario.

A limitation is that the trained fNN in this paper works in trajectory tracking tasks but not set-point regulation tasks. This is because the fNN is trained by data collected from exciting trajectory tracking tasks. Therefore, the fNN model will be capable of working for both trajectory tracking tasks (*i.e.*, tasks with non-zero velocity) and setpoint regulation tasks (*i.e.*, tasks with zero velocity) if it is trained by data from both. This will be evaluated in future work.

## IV. Conclusions

In this paper, a framework for learning robot dynamic model uncertainties and separating external disturbances by integrating impedance controller, NDOB observer, and feedforward neural network (fNN) model was presented. By accurately controlling each of the four uncertainty sources in the simulations, the results show that the fNN can accurately learn the internal disturbances (source-1,2) thus separating the external disturbances (source-3,4). A further simulation on an application scenario, where an external obstacle of virtual fixture (VF) is involved, demonstrated the effectiveness of the presented learning framework. The fNN model in the framework is robust to learn all uncertainties that exist in the training process, then estimate all other newly added uncertainties in the testing process or application scenarios.

In the present work, the fNN model was only validated in scenarios of robots in free motion and obstructive motion. In future work, we will expand it to set-point regulation tasks which will enable human-robot interaction by separating the interaction force. Also, physical experiments will be conducted to evaluate the effectiveness of the proposed framework in real application scenarios.

## References

[1] C. Gaz *et al.*, "Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4147–4154, 2019.

[2] A. Radke and Z. Gao, "A survey of state and disturbance observers for practitioners," in *2006 American Control Conference*. IEEE, 2006, pp. 5183–5188.

[3] W.-H. Chen *et al.*, "Disturbance-observer-based control and related methods—an overview," *IEEE Transactions on industrial electronics*, vol. 63, no. 2, pp. 1083–1095, 2016.

[4] A. Mohammadi, H. J. Marquez, and M. Tavakoli, "Nonlinear disturbance observers: Design and applications to Euler-Lagrange systems," *IEEE Control Systems Magazine*, vol. 37, no. 4, pp. 50–72, 2017.

[5] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot collisions: A survey on detection, isolation, and identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, 2017.

[6] G. Sebastian *et al.*, "Interaction force estimation using extended state observers: An application to impedance-based assistive and rehabilitation robotics," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1156–1161, 2019.

[7] A. Mohammadi *et al.*, "Nonlinear disturbance observer design for robotic manipulators," *Control Engineering Practice*, vol. 21, no. 3, pp. 253–267, 2013.

[8] J. Hu and R. Xiong, "Contact force estimation for robot manipulator using semiparametric model and disturbance Kalman filter," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 4, pp. 3365–3375, 2018.

[9] S. Liu, L. Wang, and X. V. Wang, "Sensorless force estimation for industrial robots using disturbance observer and neural learning of friction approximation," *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102168, 2021.

[10] T. Zhang and X. Liang, "Disturbance observer-based robot end constant contact force-tracking control," *Complexity*, vol. 2019, 2019.

[11] G. Garofalo *et al.*, "Sliding mode momentum observers for estimation of external torques and joint acceleration," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6117–6123.

[12] J. Han, "A class of extended state observers for uncertain systems," *Control and decision*, vol. 10, no. 1, pp. 85–88, 1995, [In Chinese].

[13] J. Han, "From PID to active disturbance rejection control," *IEEE transactions on Industrial Electronics*, vol. 56, no. 3, pp. 900–906, 2009.

[14] T. Ren *et al.*, "Collision detection and identification for robot manipulators based on extended state observer," *Control Engineering Practice*, vol. 79, pp. 144–153, 2018.

[15] W.-H. Chen *et al.*, "A nonlinear disturbance observer for robotic manipulators," *IEEE Transactions on industrial Electronics*, vol. 47, no. 4, pp. 932–938, 2000.

[16] S. Yousefizadeh and T. Bak, "Nonlinear disturbance observer for external force estimation in a cooperative robot," in *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, pp. 220–226.

[17] S. Liu, L. Wang, and X. V. Wang, "Sensorless force estimation for industrial robots using disturbance observer and neural learning of friction approximation," *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102168, 2021.

[18] T. Li *et al.*, "Integrating impedance control and nonlinear disturbance observer for robot-assisted arthroscope control in elbow arthroscopic surgery," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 11 172–11 179.

[19] M. Sharifi *et al.*, "Adaptive CPG-based gait planning with learning-based torque estimation and control for exoskeletons," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8261–8268, 2021.

[20] J. Fong, H. Rouhani, and M. Tavakoli, "A therapist-taught robotic system for assistance during gait therapy targeting foot drop," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 407–413, 2019.

[21] S. Bruno *et al.*, "Robotics: modelling, planning and control," 2010.

[22] P. Song, Y. Yu, and X. Zhang, "A tutorial survey and comparison of impedance control on robotic manipulation," *Robotica*, vol. 37, no. 5, pp. 801–836, 2019.

[23] A. Torabi *et al.*, "Application of a redundant haptic interface in enhancing soft-tissue stiffness discrimination," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1037–1044, 2019.

[24] M. C. Çavuşoğlu, D. Feygin, and F. Tendick, "A critical study of the mechanical and electrical properties of the phantom haptic interface and improvements for highperformance control," *Presence: Teleoperators & Virtual Environments*, vol. 11, no. 6, pp. 555–568, 2002.