Int. J. Fuzzy Syst.
https://doi.org/10.1007/s40815-024-01921-x

Adaptive Position/Force Controller Design Using Fuzzy Neural Network and Stiffness Estimation for Robot Manipulator

Bo-Ru Tseng[1] · **Jun-Yi Jiang**[2] · **Ching-Hung Lee**[1,3]

Received: 24 May 2024 / Revised: 17 October 2024 / Accepted: 11 November 2024

Abstract This paper proposes an adaptive hybrid position/force control approach using fuzzy neural networks (FNNs) for a robot manipulator with joint friction compensation. The dynamics model and system uncertainties are estimated by FNNs. For force tracking control, an adaptive impedance controller is employed with an online stiffness estimator, wherein the stiffness of the contacted environment is estimated using a gradient descent algorithm. The adaptive update laws of the FNNs and the stability of the controller are obtained using the Lyapunov stability theorem. Finally, the proposed adaptive hybrid controller is implemented on the AR605, a 6-axis articulated robot manipulator manufactured by the Industrial Technology Research Institute (ITRI). The effectiveness and robustness of the proposed control strategies are verified by the simulation and experimental results.

Keywords Robot manipulator · Adaptive control · Force control · Fuzzy neural network

✉ Ching-Hung Lee
chl@nycu.edu.tw

Bo-Ru Tseng
b.r.tseng@gmail.com

[1] Institute of Electrical and Control Engineering, National Yang Ming Chiao Tung University, Daxue Rd., Hsinchu City 300, Taiwan

[2] Department of Mechanical Engineering, National Chung Hsing University, Xingda Rd., Taichung City 402, Taiwan

[3] Department of Electrical Engineering, Chun Yuan Christian University, Zhongbei Rd., Taoyuan City 320, Taiwan

1 Introduction

Recently, several intelligent controllers have been proposed for robot manipulators, since the dynamics model is a highly nonlinear system [1–12]. In general, the dynamics model of robot manipulators are not exactly known and hard to be modeled appropriately. To address these issues, many researches have made efforts to investigate the robot control learning problems to develop advanced adaptive control strategies. Due to the prominent approximation capability of the neural network (NN) and fuzzy system, they have been utilized for adaptive control systems to estimate the unknown dynamics properties [12]. Literature [5,11–15] incorporated the NN learning methods into fuzzy logic systems for developing adaptive robot controllers to address the problems of lacking automatic learning capabilities in the conventional fuzzy systems. A fuzzy neural network (FNN) based sliding mode control scheme has been proposed to reduce the need for detailed system information and to address chattering control efforts [8]. A radial basis function neural network (RBFNN) has been adopted to online learn the inverse dynamics of the robot manipulator in [9,16], owing to its fast learning speed for the approximated problems and superior robustness to the disturbances. The literature [10] proposed an adaptive bias radial basis function neural network (BRBFNN) control scheme designed to effectively mitigate the influence of dynamic biases, which may arise from uncertainties in gravitational effects or external payloads. While adaptive radial basis function neural network (RBFNN) methods demonstrate outstanding function approximation capabilities, rapid learning speed, and effective handling of uncertainties in dynamic models, they do present a significant drawback. To enhance approximation performance, the number of basis nodes in each input channel must be increased, which may lead to polynomial growth in the dimensionality of the

Published online: 18 December 2024

RBFNNs, consequently raising computational costs. A novel adaptive controller design integrates a broad learning system (BLS) [17] with FNNs and NNs has been proposed in [11,12], where the control performances are achieved via the online incremental network structure update laws. However, this approach also increases the complexity of the controller scheme and raises computational costs. The FNNs has the ability to adaptively compensate the inherent uncertainties in system models and has a lighter computation loads by using fewer adjustable parameters. In this paper, a simple FNN-based adaptive robot controller are proposed to achieve good tracking performances with robustness and high computational efficiency. The FNNs are employed to constructed the dynamics model estimators of the robot manipulator. The controller is designed in joint space. When the robot manipulator engages in contact tasks, the environment generate a contact force 'that must be considered as disturbance and suppressed. Otherwise, it can lead to increased interaction force, resulting in saturation, instability, and potential physical failure [18]. To manage external force control, force/position control and impedance control methods are introduced [15,18–27]. Hybrid force/position control focuses on regulating the position and the end-effector forces in both tangential and perpendicular directions of the contact surface [18,27]. It controls the end-effector to follow the desired trajectory in the normal direction, while in the tangential direction, it addresses the force control problem. In addition, impedance control adjusts the position of the end-effector and senses the contact force, so that a second-order mass-spring-damper system meets the target impedance equation [5,15,24,27]. Consequently, this study applies the hybrid position/force control scheme to develop our adaptive controller, specifically for industrial machining applications, such as polishing and grinding. To realize the hybrid control scheme, it may face the challenge of control mode switching between free-space motion and contact scenarios, necessitating the definition of switching rules. Poor switching events could significantly lead the robot system unstable. To address these challenges, control strategies for switched systems are proposed in [28,29]. The control of switched systems will be a crucial issue for this study in the future. Therefore, this study will set some restrictions on application scenarios to bypass this problem. This paper develops an adaptive position/force controller for an n-link robot manipulator, addressing system uncertainties and unknown environmental stiffness. FNN-based estimators are used to estimate system model matrices and mitigate uncertainties, ensuring controller stability via the Lyapunov stability theorem. Additionally, the paper explores joint friction effects on position tracking, using FNNs to estimate friction characteristics and conduct robustness analysis against friction variations, compared to conventional methods [30]. For force control with unknown environmental stiffness, an update law is derived using the gradient method. The adaptive controller is applied to the AR605 6-axis robot manipulator developed by the Industrial Technology Research Institute (ITRI), with effectiveness verified through experimental results.

The organization of this paper is as follows: Sect. 2 presents the preliminaries, including the FNN and the dynamics system of the robot manipulator. The proposed adaptive hybrid position/force controller is introduced in Sect. 3. The corresponding simulation and experimental results are discussed in Sect. 4. Finally, the conclusion is provided.

2 System Descriptions

2.1 Robot Manipulator Dynamics

We consider the dynamics of n-link robot manipulator modeled using the Euler–Lagrange method, as follows [31]:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{\tau}_f(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} - \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{F}_{ext} \quad (1)$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ denote the joint positions, angular velocities, and angular accelerations, respectively. $\boldsymbol{\tau} \in \mathbb{R}^n$ is the joint torques, $\boldsymbol{F}_{ext} \in \mathbb{R}^m$ represents the external force applied to the end-effector, quantities m and n are the dimension of Cartesian-space and joint-space, respectively, with the condition that $m \leq n$, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the inertia matrix, which is symmetric bounded and positive definite, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} \in \mathbb{R}^n$ represents the centrifugal and Coriolis torques, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ is the gravitational torques, $\boldsymbol{\tau}_f \in \mathbb{R}^n$ is the friction torques of each joint, and $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the geometric Jacobian matrix expressed in Cartesian task-space.

In this paper, an adaptive position control law is designed based on the dynamics model (1). Therefore, the system is developed according to the following assumptions and properties [31,32]:

Assumption 1 The external forces are bounded by nominal allowance exerted forces determined by the robot specification, i.e. $|\boldsymbol{F}_{ext}| < \Delta_f$.

Assumption 2 The dynamic system of the robot manipulator in (1) has unknown uncertainties and are bounded, as follows:

$$\boldsymbol{M} = \bar{\boldsymbol{M}} + \boldsymbol{\epsilon}_M \quad (2a)$$

$$\boldsymbol{C} = \bar{\boldsymbol{C}} + \boldsymbol{\epsilon}_C \quad (2b)$$

$$\boldsymbol{g} = \bar{\boldsymbol{g}} + \boldsymbol{\epsilon}_g \quad (2c)$$

$$\boldsymbol{\tau}_f = \bar{\boldsymbol{\tau}}_f + \boldsymbol{\epsilon}_f, \quad (2d)$$

where $\bar{\boldsymbol{M}}$, $\bar{\boldsymbol{C}}$, $\bar{\boldsymbol{g}}$ and $\bar{\boldsymbol{\tau}}_f$ represent the nominal part, and $\boldsymbol{\epsilon}_M \in \mathbb{R}^{6\times 6}$, $\boldsymbol{\epsilon}_C \in \mathbb{R}^6$, $\boldsymbol{\epsilon}_g \in \mathbb{R}^6$, and $\boldsymbol{\epsilon}_f \in \mathbb{R}^6$ denote the model

uncertainties, regarded as the modeling error. Furthermore, the lumped modeling errors have an maximum boundary δ^*, such that,

$$\boldsymbol{\epsilon}_l = \boldsymbol{\epsilon}_M \ddot{\boldsymbol{q}} + \boldsymbol{\epsilon}_C \dot{\boldsymbol{q}} + \boldsymbol{\epsilon}_g + \boldsymbol{\epsilon}_f, \tag{3}$$

with

$$|\boldsymbol{\epsilon}_l| \leq \delta^*, \tag{4}$$

where $\boldsymbol{\epsilon}_l$ is the bounded lumped modeling error.

Property 1 The matrix $\boldsymbol{M}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is skew-symmetric such that satisfies the property as follows:

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}^T(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{5}$$

In most practical cases, the nominal dynamics are not exactly known [9,12,15,19,33]. To address this, we employ an FNN model to design a dynamic estimator tasked with estimating the dynamics model and compensating for system uncertainties through an online learning mechanism. This approach is then used to implement an adaptive control scheme for position tracking.

2.2 Fuzzy Neural Networks Modeling

As described above, the unknown dynamic models are estimated by FNNs. The FNNs are composed of four layers: input nodes, membership nodes, rule nodes, and output nodes, as illustrated in the schematic structure in Fig. 1. Layer 1 transmits the input variables x_n to the next layer as linguistic values. Layer 2 operates the fuzzification inference, mapping the linguistic values into fuzzy sets by a grade of membership as follows:

$$O_{ij}^{(2)}(t) = \exp\left[-\frac{(x_n(t) - m_{ij})^2}{\sigma_{ij}^2}\right] \tag{6}$$

where the notation $\boldsymbol{O}^{(i)}$ denotes the i-th layer output vector of the FNN, and m_{ij} and σ_{ij} represent the center and width of the Gaussian functions, respectively. The rule nodes layer is adopted as a fully connected network structure, such that:

$$O_k^{(3)}(t) = \prod O_{ij}^{(2)}(t), \tag{7}$$

The output nodes conduct the defuzzification operations.

$$y = O^{(4)}(t) = \boldsymbol{w}^T \boldsymbol{\psi}^k = \frac{\sum_{k=1}^{R} w_k O_k^{(3)}(t)}{\sum_{k=1}^{R} O_k^{(3)}(t)} \tag{8}$$

where $\boldsymbol{w} = [w_1\, w_2 \ldots\, w_R]^T$ is the weighting vector of the membership grade of rules, ψ^k is the k-th fuzzy rule output, and $R = ij$ is the total number of rules.

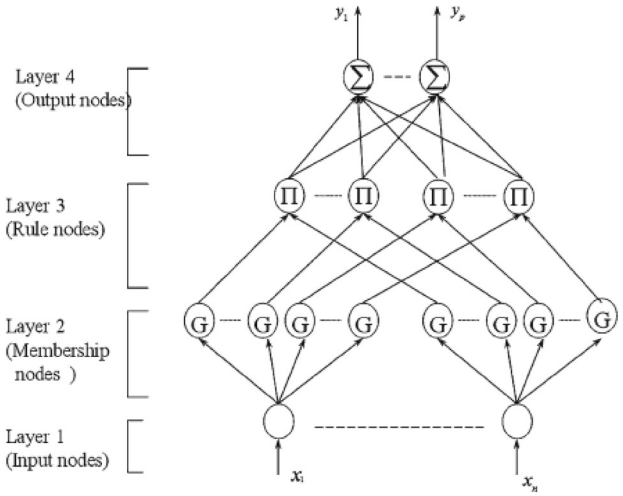

Fig. 1 Schematic of FNN structure [9]

In this paper, FNNs are adopted to estimate the unknown dynamic matrices, $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{g}$, and $\boldsymbol{\tau}_f$. Consequently, the dynamic models can be represented as a combination of the optimally approximated models, computed by the FNNs, and the bounded approximation errors, as

$$\boldsymbol{M} = \boldsymbol{w}_M^{*T} \boldsymbol{\psi}_M + \boldsymbol{\epsilon}_M \tag{9a}$$

$$\boldsymbol{C} = \boldsymbol{w}_C^{*T} \boldsymbol{\psi}_C + \boldsymbol{\epsilon}_C \tag{9b}$$

$$\boldsymbol{g} = \boldsymbol{w}_g^{*T} \boldsymbol{\psi}_g + \boldsymbol{\epsilon}_g \tag{9c}$$

$$\boldsymbol{\tau}_f = \boldsymbol{w}_f^{*T} \boldsymbol{\psi}_f + \boldsymbol{\epsilon}_f, \tag{9d}$$

where $\boldsymbol{w^*}_M \in \mathbb{R}^{p^2 R}$, $\boldsymbol{w^*}_C \in \mathbb{R}^{p^2 R}$, $\boldsymbol{w^*}_g \in \mathbb{R}^{pR}$, and $\boldsymbol{w^*}_f \in \mathbb{R}^{pR}$ are the optimal weighting matrices.

2.3 Position-Based Impedance Control

In robotics, impedance control is widely used to adjust the interaction response between the robot manipulator and its contact environments. In contact scenario, the position of end-effector $\boldsymbol{X}$ satisfies the following target impedance equation:

$$\boldsymbol{M_m}(\ddot{X} - \ddot{X}_d) + \boldsymbol{B_m}(\dot{X} - \dot{X}_d) + \boldsymbol{K_m}(X - X_d) = -\boldsymbol{F_{ext}}, \tag{10}$$

where $\boldsymbol{M_m}$, $\boldsymbol{B_m}$, and $\boldsymbol{K_m}$ represent the desired robot inertia, damping, and stiffness, respectively, and can be designed as customized control gains. Since inertia and damping parameters only affect the transient response, and assuming the system only considers the steady-state response, Eq. (10) can be simplified as follows:

$$\boldsymbol{K_m}(\boldsymbol{X - X_d}) = -\boldsymbol{F}_{ext}, \tag{11}$$

Considering only the steady-state response, the interaction between the robot manipulator and the environment can be modeled as a linear spring form:

$$\boldsymbol{F}_{ext} = \boldsymbol{K_e}(\boldsymbol{X} - \boldsymbol{X_e}) \tag{12}$$

where $\boldsymbol{K_e}$ represents the stiffness coefficient of the interacting environment, and $\boldsymbol{X_e}$ denotes the position of the contact environment surface. In machining applications such as polishing and grinding, the tool-center-point (TCP) of the robot manipulator must be compliant with workpiece surface to achieve the desired contact force. The position-based impedance control strategies had been implemented in [34,35]. According to (12), if an ideal robot position $\boldsymbol{X}^*$ exists that allows the end-effector and the contact environment to satisfy a desired interactive force $\boldsymbol{F}_d$, the relationship can be expressed as:

$$\boldsymbol{F}_d \cong \boldsymbol{K_e}(\boldsymbol{X^*} - \boldsymbol{X_e}) \tag{13}$$

where $\boldsymbol{X}^*$ is the ideal trajectory such that $\boldsymbol{F}_{ext} = \boldsymbol{F}_d$. By substituting (13) into (11) and (12), an ideal position command can be derived as follows,

$$\boldsymbol{X}_d^* = \boldsymbol{K}_m^{-1}(\boldsymbol{K}_m - \boldsymbol{K}_e)\boldsymbol{X}^* - (\boldsymbol{K}_m + \boldsymbol{K}_e)^{-1}\boldsymbol{K}_e\boldsymbol{X}_e \tag{14}$$

Once the precise location of the environment, $\boldsymbol{X}_e$, and the exact value of environment stiffness, $\boldsymbol{K}_e$, are acquired, the ideal position command can be obtained based on (14) to achieve the desired contact force in the steady-state.

In this study, the proposed force control method is primarily intended for specific applications such as polishing and grinding. Therefore, we assume that the actual environment position $\boldsymbol{X}_e$ represents the surface of the workpiece and is obtained from the CAD-CAM model. Consequently, this study develops an adaptive estimation strategy for environmental stiffness based on the force tracking error over time, as discussed in Sect. 3.2.

3 Adaptive Position/Force Controller Design

This paper proposes an adaptive hybrid position/force controller for a n-link serial-link robot manipulator. In this section, the hybrid controller design is divided into two phases: (1) position tracking control and (2) force tracking control. The proposed hybrid controller scheme is shown as Fig. 2. The aim of the hybrid control scheme is to ensure that the end-effector follows the reference trajectory while simultaneously achieving the desired contact forces. The stability of the closed-loop system is guaranteed, and the update laws of the controller are derived using the Lyapunov stability theorem.

3.1 FNN-Based Position Controller

In this subsection, we first investigate the behavior of the robot manipulator motion in free-space, i.e. $\boldsymbol{F}_{ext} = 0$. For position tracking, the controllers are designed to generate the torque commands for each joint. Given the inherent uncertainty in the dynamics model parameters, and the fact that the dynamics matrices M, C, g, and τ_f are generally unknown [10,12,19], the FNNs are tasked with estimating them. Firstly, we define $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q_d}$ and $\boldsymbol{s} = \dot{\boldsymbol{e}} + \boldsymbol{\Lambda e}$, where $\boldsymbol{\Lambda} = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$ with $\lambda_i > 0$ for $i = 1, 2, \ldots, n$. Subsequently, the dynamics model (1) can be rewritten as follows:

$$\hat{\boldsymbol{M}}\dot{\boldsymbol{s}} + \hat{\boldsymbol{C}}\boldsymbol{s} + \hat{\boldsymbol{g}} + \hat{\boldsymbol{\tau}}_f + \hat{\boldsymbol{M}}\ddot{\boldsymbol{q}}_d - \hat{\boldsymbol{M}}\boldsymbol{\Lambda}\dot{\boldsymbol{e}} + \hat{\boldsymbol{C}}\dot{\boldsymbol{q}}_d - \hat{\boldsymbol{C}}\boldsymbol{\Lambda e} = \boldsymbol{\tau} \tag{15}$$

where $\hat{\boldsymbol{M}} = \hat{\boldsymbol{w}}_M^T\boldsymbol{\psi}_M$, $\hat{\boldsymbol{C}} = \hat{\boldsymbol{w}}_C^T\boldsymbol{\psi}_C$, $\hat{\boldsymbol{g}} = \hat{\boldsymbol{w}}_g^T\boldsymbol{\psi}_g$, and $\hat{\boldsymbol{\tau}}_f = \hat{\boldsymbol{w}}_f^T\boldsymbol{\psi}_f$ are the estimations of $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{g}$, and $\boldsymbol{\tau}_f$ by FNNs, where $\psi_{(\cdot)}$ denotes the firing strength of the corresponding rule. Hence, we define the following theorem according to the Lyapunov stability theorem [15].

Theorem 1 *Consider the system model of the robot manipulator (1) with uncertainties that are not exactly known. The adaptive torque controller can be designed as follows:*

$$\boldsymbol{\tau} = \hat{\boldsymbol{M}}\ddot{\boldsymbol{q}}_d + \hat{\boldsymbol{g}} + \hat{\boldsymbol{\tau}}_f + \boldsymbol{\tau}_r - \hat{\boldsymbol{M}}\boldsymbol{\Lambda}\dot{\boldsymbol{e}} + \hat{\boldsymbol{C}}\dot{\boldsymbol{q}}_d - \hat{\boldsymbol{C}}\boldsymbol{\Lambda e} - \boldsymbol{K}_d\boldsymbol{s} \tag{16}$$

$$\boldsymbol{\tau}_r = -\hat{\delta}[\mathrm{sgn}(s_1), \ldots, \mathrm{sgn}(s_n)]^T \tag{17}$$

where $\boldsymbol{\tau}_r$ is the robust controller, which tasks with compensating the estimation errors of the dynamics matrices in (9), and $\boldsymbol{K}_d \in \mathbb{R}^{n\times n}$ is a positive definite diagonal matrix, $\hat{\delta}$ represents the estimation of approximation error boundary. In addition, the update laws for the FNNs and the robust controller are chosen as follows:

$$\dot{\hat{\boldsymbol{w}}}_M = -\boldsymbol{\eta}_M^{-1}[\boldsymbol{\psi}_M(\ddot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\dot{\boldsymbol{e}})\boldsymbol{s}^T] \tag{18a}$$

$$\dot{\hat{\boldsymbol{w}}}_C = -\boldsymbol{\eta}_C^{-1}[\boldsymbol{\psi}_M(\dot{\boldsymbol{q}}_d - \boldsymbol{\Lambda e})\boldsymbol{s}^T] \tag{18b}$$

$$\dot{\hat{\boldsymbol{w}}}_f = -\boldsymbol{\eta}_f^{-1}(\boldsymbol{\psi}_f\boldsymbol{s}^T) \tag{18c}$$

$$\dot{\hat{\boldsymbol{w}}}_g = -\boldsymbol{\eta}_g^{-1}(\boldsymbol{\psi}_g\boldsymbol{s}^T) \tag{18d}$$

$$\dot{\hat{\delta}} = -\Gamma_\delta \|s\| \tag{18e}$$

where $\boldsymbol{\eta}_M$, $\boldsymbol{\eta}_C$, $\boldsymbol{\eta}_g$, and $\boldsymbol{\eta}_f$ are diagonal positive definite matrices, and Γ_δ is a positive adaptive parameter rate. The asymptotic convergence of the tracking error is guaranteed via the Lyapunov theorem. This means that the state $\boldsymbol{q}$ will follow the desired $\boldsymbol{q_d}$, and consequently, the trajectory $\boldsymbol{x}$ will follow $\boldsymbol{x}_d$ as t approaches infinity.

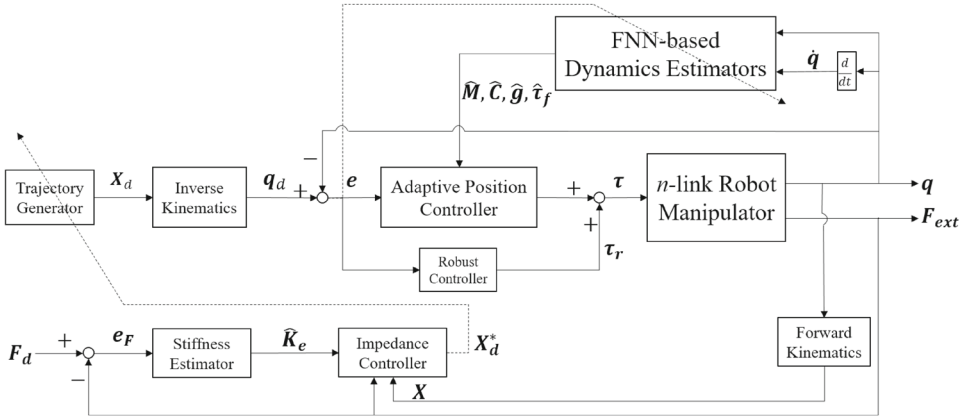

Fig. 2 Schematic of the hybrid position/force controller

Proof of Theorem 1 Firstly, by substituting the controllers (16) into the dynamics model (1), we obtain a closed-loop system model as follows:

$$\begin{aligned} \boldsymbol{M}\dot{\boldsymbol{s}} + \boldsymbol{C}\boldsymbol{s} = &(\hat{\boldsymbol{M}} - \boldsymbol{M})(\ddot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\dot{\boldsymbol{e}}) \\ &+ (\hat{\boldsymbol{C}} - \boldsymbol{C})(\dot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\boldsymbol{e}) \\ &+ (\hat{\boldsymbol{g}} - \boldsymbol{g}) + (\hat{\boldsymbol{\tau}}_f - \boldsymbol{\tau}_f) \\ &+ \boldsymbol{\tau}_r - \boldsymbol{K}_d\boldsymbol{s} \end{aligned} \tag{19}$$

substitute (17) and (18) into (19)

$$\begin{aligned} \boldsymbol{M}\dot{\boldsymbol{s}} + \boldsymbol{C}\boldsymbol{s} + \boldsymbol{K}_d\boldsymbol{s} = &\tilde{\boldsymbol{w}}_M^T\boldsymbol{\psi}_M(\ddot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\dot{\boldsymbol{e}}) + \tilde{\boldsymbol{w}}_C^T\boldsymbol{\psi}_C(\dot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\boldsymbol{e}) \\ &+ \tilde{\boldsymbol{w}}_g^T\boldsymbol{\psi}_g + \tilde{\boldsymbol{w}}_f^T\boldsymbol{\psi}_f + \boldsymbol{\epsilon}_l + \boldsymbol{\tau}_r \end{aligned} \tag{20}$$

where $\tilde{w}_{(\cdot)} = \hat{w}_{(\cdot)} - w_{(\cdot)}^*$. Subsequently, define the Lyapunov candidate function as follows:

$$\begin{aligned} V = &\frac{1}{2}\boldsymbol{s}^T\boldsymbol{M}\boldsymbol{s} + \frac{1}{2\Gamma_\delta}\tilde{\delta}^2 + \frac{1}{2}Tr(\tilde{\boldsymbol{w}}_M^T\boldsymbol{\eta}_M\tilde{\boldsymbol{w}}_M \\ &+ \tilde{\boldsymbol{w}}_C^T\boldsymbol{\eta}_C\tilde{\boldsymbol{w}}_C + \tilde{\boldsymbol{w}}_g^T\boldsymbol{\eta}_g\tilde{\boldsymbol{w}}_g + \tilde{\boldsymbol{w}}_f^T\boldsymbol{\eta}_f\tilde{\boldsymbol{w}}_f) \end{aligned} \tag{21}$$

where $Tr(\cdot)$ denotes the trace operation, $\tilde{\delta} = \hat{\delta} - \delta^*$. Hence, the Lyapunov candidate function can be expressed as follows:

$$\begin{aligned} \dot{V} = &\boldsymbol{s}^T(-\boldsymbol{C}\boldsymbol{s} - \boldsymbol{K}_d\boldsymbol{s} + \tilde{\boldsymbol{w}}_M^T\boldsymbol{\psi}_M(\ddot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\dot{\boldsymbol{e}}) \\ &+ \tilde{\boldsymbol{w}}_C^T\boldsymbol{\psi}_C(\dot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\boldsymbol{e}) + \tilde{\boldsymbol{w}}_g\boldsymbol{\psi}_g + \tilde{\boldsymbol{w}}_{\tau_f}\boldsymbol{\psi}_f) \\ &+ \frac{1}{\Gamma_\delta}\tilde{\delta}\dot{\tilde{\delta}} + \boldsymbol{s}^T\boldsymbol{\epsilon}_l + \boldsymbol{s}^T\boldsymbol{\tau}_{robust} + \frac{1}{2}\boldsymbol{s}^T\dot{\boldsymbol{M}}\boldsymbol{s} \\ &+ Tr(\tilde{\boldsymbol{w}}_M^T\boldsymbol{\eta}_M\dot{\tilde{\boldsymbol{w}}}_M^T + \tilde{\boldsymbol{w}}_C^T\boldsymbol{\eta}_C\dot{\tilde{\boldsymbol{w}}}_C^T + \tilde{\boldsymbol{w}}_g^T\boldsymbol{\eta}_g\dot{\tilde{\boldsymbol{w}}}_g^T \\ &+ \tilde{\boldsymbol{w}}_f^T\boldsymbol{\eta}_f\dot{\tilde{\boldsymbol{w}}}_f^T). \end{aligned} \tag{22}$$

since $\boldsymbol{s}^T[\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})]\boldsymbol{s} = 0$, such that,

$$\begin{aligned} \dot{V} = &-\boldsymbol{s}^T\boldsymbol{K}_d\boldsymbol{s} + Tr(\tilde{\boldsymbol{M}}_M^T[\boldsymbol{\psi}_M(\ddot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\dot{\boldsymbol{e}})\boldsymbol{s}^T \\ &+ \boldsymbol{\eta}_M + \dot{\hat{\boldsymbol{w}}}_M] + \tilde{\boldsymbol{w}}_C^T[\boldsymbol{\psi}_C(\dot{\boldsymbol{q}}_d - \boldsymbol{\Lambda}\boldsymbol{e})\boldsymbol{s}^T \\ &+ \boldsymbol{\eta}_C + \dot{\hat{\boldsymbol{w}}}_C] + \tilde{\boldsymbol{w}}_g^T[\boldsymbol{\psi}_g\boldsymbol{s}^T + \boldsymbol{\eta}_g\dot{\hat{\boldsymbol{w}}}_g] \\ &+ \tilde{\boldsymbol{w}}_f^T[\boldsymbol{\psi}_f\boldsymbol{s}^T + \boldsymbol{\eta}_f\dot{\hat{\boldsymbol{w}}}_f]) \\ &+ \boldsymbol{s}^T\boldsymbol{\epsilon}_l + \boldsymbol{s}^T\boldsymbol{\tau}_r + \frac{1}{\Gamma_\delta}\tilde{\delta}\dot{\tilde{\delta}}. \end{aligned} \tag{23}$$

Since we select the update laws as (18) and the lumped estimation error is bounded by δ^*, we can have the following inequality:

$$\begin{aligned} \dot{V} &= -\boldsymbol{s}^T\boldsymbol{K}_d\boldsymbol{s} + \boldsymbol{s}^T\boldsymbol{\epsilon}_l + \boldsymbol{s}^T\boldsymbol{\tau}_{robust} + \frac{1}{\Gamma_\delta}\tilde{\delta}\dot{\tilde{\delta}} \\ &\le -\boldsymbol{s}^T\boldsymbol{K}_d\boldsymbol{s} + |\boldsymbol{s}||\boldsymbol{\epsilon}_l| - \hat{\delta}|\boldsymbol{s}| + \frac{1}{\Gamma_\delta}\tilde{\delta}\dot{\tilde{\delta}} \\ &\le -\boldsymbol{s}^T\boldsymbol{K}_d\boldsymbol{s} + |\boldsymbol{s}||\boldsymbol{\epsilon}_l| - \hat{\delta}|\boldsymbol{s}| + (\hat{\delta} - \delta^*)|\boldsymbol{s}| \\ &\le -\boldsymbol{s}^T\boldsymbol{K}_d\boldsymbol{s} - |\boldsymbol{s}|(\delta^* - |\boldsymbol{\epsilon}_l|) \le 0. \end{aligned} \tag{24}$$

□

Subsequently, the stability of the closed-loop system is guaranteed by choosing a positive definite matrix $\boldsymbol{K}_d$ via the Lyapunov stability theorem. According to Theorem 1, the FNN-based position controller (16), with the update laws of (18), ensures that the robot follows the desired trajectory. This means that the tracking error $\boldsymbol{e} = \boldsymbol{q} - \boldsymbol{q}_d$ will converge to zero as time approaches infinity.

3.2 Adaptive Force Controller

In the force control loop, a position-based impedance control is developed based on (14). Since the environment position $\boldsymbol{X}_e$ is assumed to have been obtained in advance, however, the environmental stiffness coefficient $\boldsymbol{K}_e$ can not be exactly known in advance. Hence, the stiffness needs to be estimated to achieve the desired contact force $\boldsymbol{F}_d$. In this work, a simple online stiffness estimator is developed to estimate the K_e

coefficient utilizing a gradient descent algorithm in every control cycle. The update law is designed as follows:

$$\hat{\boldsymbol{K}}_e(t+1) = \hat{\boldsymbol{K}}_e(t) - \Delta\hat{\boldsymbol{K}}_e(t) \tag{25a}$$

$$\Delta\hat{\boldsymbol{K}}_e(t) = -\boldsymbol{\xi}_{K_e}\frac{\partial V_F(t)}{\partial\hat{\boldsymbol{K}}_e(t)} \tag{25b}$$

$$\boldsymbol{V}_F = \frac{1}{2}(\boldsymbol{F}_e - \hat{\boldsymbol{K}}_e(\boldsymbol{X} - \boldsymbol{X}_e))^2 \tag{25c}$$

where $\boldsymbol{\xi}_{K_e}$ is the learning rate, and $\boldsymbol{V}_F$ is the loss function. The convergence of the stiffness estimation can be guaranteed by choosing a suitable learning rate. Thus, in this paper, we use the following theorem by the Lyapunov approach to tune the learning rate properly.

Theorem 2 *Consider the adaptive force control as described above. The convergence of $\hat{K}_e$ is guaranteed if ξ_{K_e} is selected such that it satisfies the following condition:*

$$0 < \frac{\xi_{K_e}}{2}\left(\frac{\partial \boldsymbol{e}_F}{\partial\hat{\boldsymbol{K}}_e}\right)^2 \leq 1. \tag{26}$$

where ξ_{K_e} is a time-variant learning rate, and $e_F = F_d - F_{ext}$ represents the force tracking error.

Proof of Theorem 2 Redefine (25c) as

$$V_F(k) = \frac{1}{2}\boldsymbol{e}_F^T(k)\boldsymbol{e}_F(k) \tag{27}$$

where k denotes the time-instant for the update cycle. Let $\Delta\boldsymbol{e}_F(k) = \boldsymbol{e}_F(k+1) - \boldsymbol{e}_F(k)$, we can obtain:

$$\begin{aligned}\Delta V_{F_{ext}}(k) &= V_{F_{ext}}(k+1) - V_{F_{ext}}(k)\\ &= \frac{1}{2}\left[\boldsymbol{e}_F(k+1) + \boldsymbol{e}_F(k)\right]^T\Delta\boldsymbol{e}_F(k)\\ &= \frac{1}{2}\Delta\boldsymbol{e}_F^T(k)\boldsymbol{e}_F(k)\\ &\quad + \boldsymbol{e}_F^T(k)\Delta\boldsymbol{e}_F(k)\end{aligned} \tag{28}$$

Based on the gradient descent algorithm, we obtain:

$$\Delta\boldsymbol{e}_F(k) \approx \frac{\partial\boldsymbol{e}_F}{\partial\hat{\boldsymbol{K}}_e}\Delta\hat{\boldsymbol{K}}_e \tag{29}$$

According to the above result, rewrite Eq. (28) as follows:

$$\begin{aligned}\Delta V_F = &-\xi_{K_e}\boldsymbol{e}_F^2\left(\frac{\partial\boldsymbol{e}_F}{\partial\hat{\boldsymbol{K}}_e}\right)^2\\ &\left(1 - \frac{1}{2}\xi_{K_e}\left(\frac{\partial\boldsymbol{e}_F}{\partial\hat{\boldsymbol{K}}_e}\right)^2\right)\end{aligned} \tag{30}$$

The stability of the force tracking error can be ensured when the chosen learning rate satisfies the condition (26) such that $\Delta V_{F_{ext}}(k) \leq 0, \forall k > 0$, meaning the force tracking error will approach zero as k approaches infinity. Consequently, based on the condition (26), a time-varying learning are designed as follows:

$$\xi_{K_e} = 2\left(\lambda\frac{\partial\hat{K}_e}{\partial e_f}\right)^2 \tag{31}$$

where $\lambda \in (0, 1]$ is a static learning parameter that influences the update rate of the estimator and can be chosen based on the sensitivity requirements of force control in the application scenario. □

Subsequently, the stiffness of the environment can be updated online for every control cycle by (25a). The ideal target trajectory $\boldsymbol{X}_d^*$ such that $\boldsymbol{F}_{ext} = \boldsymbol{F}_d$ is calculated using the impedance control model. According to the literature [15], the adaptive force control can be implemented by the following steps:

$$\Delta\boldsymbol{X}(t) = \hat{\boldsymbol{K}}^{-1}(t)\boldsymbol{e}_{\boldsymbol{F}}(t) = \boldsymbol{X}^*(t) - \boldsymbol{X}(t) \tag{32a}$$

$$\boldsymbol{X}^*(t) = \boldsymbol{X}(t) + \Delta\boldsymbol{X}(t) \tag{32b}$$

$$\begin{aligned}\boldsymbol{X}_d^*(t+1) = &\boldsymbol{K}_m^{-1}(\boldsymbol{K}_m - \hat{\boldsymbol{K}}_e)(\boldsymbol{X}^*(t)\\ &- (\boldsymbol{K}_m + \hat{\boldsymbol{K}}_e)^{-1}\hat{\boldsymbol{K}}_e\boldsymbol{X}_e(t)),\end{aligned} \tag{32c}$$

where $\boldsymbol{X}_d^*$ represents the optimal position command. This command is calculated to modify the original reference command trajectory, ensuring that the end-effector of the robot manipulator exerts the desired contact force on the environment.

4 Simulations and Experimental Results

In this section, the proposed adaptive control scheme was implemented on the 6-axis articulated robot manipulator, AR605, manufactured by the Taiwan Industrial Technology Research Institute (ITRI), as shown in Fig. 3. Simulations and actual robot manipulator experiments were conducted to evaluate the performance and effectiveness of the control scheme. Based on the aforementioned description, the dynamics model of the robot manipulator, including $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{g}$, and $\boldsymbol{\tau}_f$, are not precisely known and are approximated using FNNs. The structure of the FNN for each of $\hat{\boldsymbol{M}}$, $\hat{\boldsymbol{C}}$, $\hat{\boldsymbol{g}}$, and $\hat{\boldsymbol{\tau}}_f$ is designed with seven fuzzy memberships for each input variable. Each input variable (i.e., $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$) is re-scaled to the range of $[-1, 1]$ using min-max normalization according to the joint angle limits and the maximum joint override speed specified by the product manual of the robot. The the centers of the fuzzy memberships are $[-0.9, -0.54, -0.18, 0.18, 0.54, 0.9]$ and the width of each membership function is 0.36. Additionally, six fuzzy rules

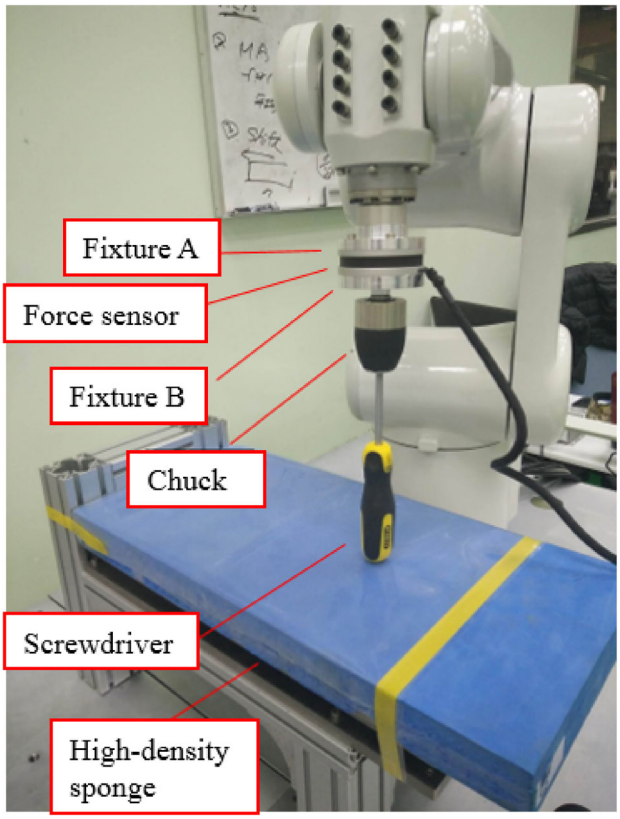

Fig. 3 Experimental setup for ITRI Robot manipulator, AR605

are chosen. The initial output weights of the FNNs are randomly selected from the range [0, 1]. The update law gain matrices are chosen as follows:

$$\boldsymbol{\eta}_M^{-1} = \text{diag}(\kappa_1, \ldots, \kappa_6), \quad \kappa_i = 10^{-3},\ i = 1, \ldots, 6 \tag{33a}$$

$$\boldsymbol{\eta}_C^{-1} = \text{diag}(\kappa_1, \ldots, \kappa_6), \quad \kappa_i = 7 \times 10^{-3},\ i = 1, \ldots, 6 \tag{33b}$$

$$\boldsymbol{\eta}_g^{-1} = \text{diag}(\kappa_1, \ldots, \kappa_6), \quad \kappa_i = 10^{-4}, i = 1, \ldots, 6 \tag{33c}$$

$$\boldsymbol{\eta}_f^{-1} = \text{diag}(\kappa_1, \ldots, \kappa_6), \quad \kappa_i = 3 \times 10^{-3},\ i = 1, \ldots, 6. \tag{33d}$$

where the update laws are selected with small values to prevent dramatic responses in both the simulations and the actual robot experiments.

The adaptive position/force controller proposed in this paper was implemented with the initial desired trajectory $\boldsymbol{X}_d$ designed in advance. In practice, trajectory planning is essential for ensuring smooth and continuous robot motion [36,37]. This paper utilizes a sigmoid function to generate the velocity profile in Cartesian space, which contains acceleration interval, constant velocity interval, and deceleration interval. The velocity profiles of each interval are calculated as follows:

$$v(t) = \begin{cases} \frac{V_{max}[\tanh(0.5w_a((t-t_0)-t_a))+1]}{2} & t \in [t_0, t_1) \\ V_{max} & t \in [t_1, t_2) \\ \frac{V_{max}[\tanh(0.5w_a(t_b-(t-t_0)))+1]}{2} & t \in [t_2, t_f] \end{cases} \tag{34}$$

where t_0, t_1, t_2 and t_f represent starting time, acceleration time, deceleration time and destination time, respectively. The $v(t)$ and V_{max} are the velocity trajectory and the specified constant velocity, respectively. $w_a > 0$ is a factor determining the steepness of acceleration. Once the acceleration time t_1 and maximum velocity V_{max} are decided, the initial desired trajectory $\boldsymbol{X}_d(t)$ can be obtained.

4.1 Simulation Results

In the simulation phase, it is necessary to build a virtual model of robot dynamics to evaluate the proposed control strategies. The actual dynamics properties of RA605, including the link masses, the geometry of the center of mass for each link, and the second moment of inertia, are obtained from the CAD model; detailed information can be found in [38]. Additionally, the simulated joint friction models were determined through experimental design for system identification on the actual robot RA605. For the modeling of joint friction, we primarily refer to the work presented in [39], as follows:

$$\tau_f(\dot{q}) = \tau_c\,\text{sgn}(\dot{q}) + (\tau_s - \tau_c)e^{-|\frac{\dot{q}}{\dot{q}_s}|^{\delta_s}}\text{sgn}(\dot{q}) + \tau_v\dot{q}^{\frac{1}{3}} \tag{35}$$

where τ_f is the velocity-dependent joint friction. The notation sgn$(\cdot)$ represents the signum function. The symbols τ_c, τ_v and τ_s represent the Coulomb friction, viscosity coefficient, and static friction, respectively. The parameter $\dot{q}_s$ is the Stribeck velocity, and δ_s is a geometric parameter within the range $0.5 \le \delta_s \le 1.0$. Detailed methodologies of the system identification processes can be found in [38]. The identification results for each joint friction are shown in Fig. 4, depicting the regression curves of joint friction with respect to angular velocity. The blue star points denote the collected data, and the red line represents the identified friction model.

The simulation results and discussions are illustrated in the following cases. In the simulation cases, the control gains of the adaptive position controller are selected as $\boldsymbol{K}_d = 20\boldsymbol{I}_{66}$ and $\boldsymbol{\Lambda} = 30\boldsymbol{I}_{66}$.

Case 1 *Text "Chris" trajectory and Z-direction force control.*

In this simulation case, the robot manipulator is driven to draw "Chris" on the X-Y plane with a constant force, as depicted in Fig. 5. The initial conditions of the robot manipulator's end-effector are $x(0) = 0$, $y(0) = 0$, $z(0) = 0$,

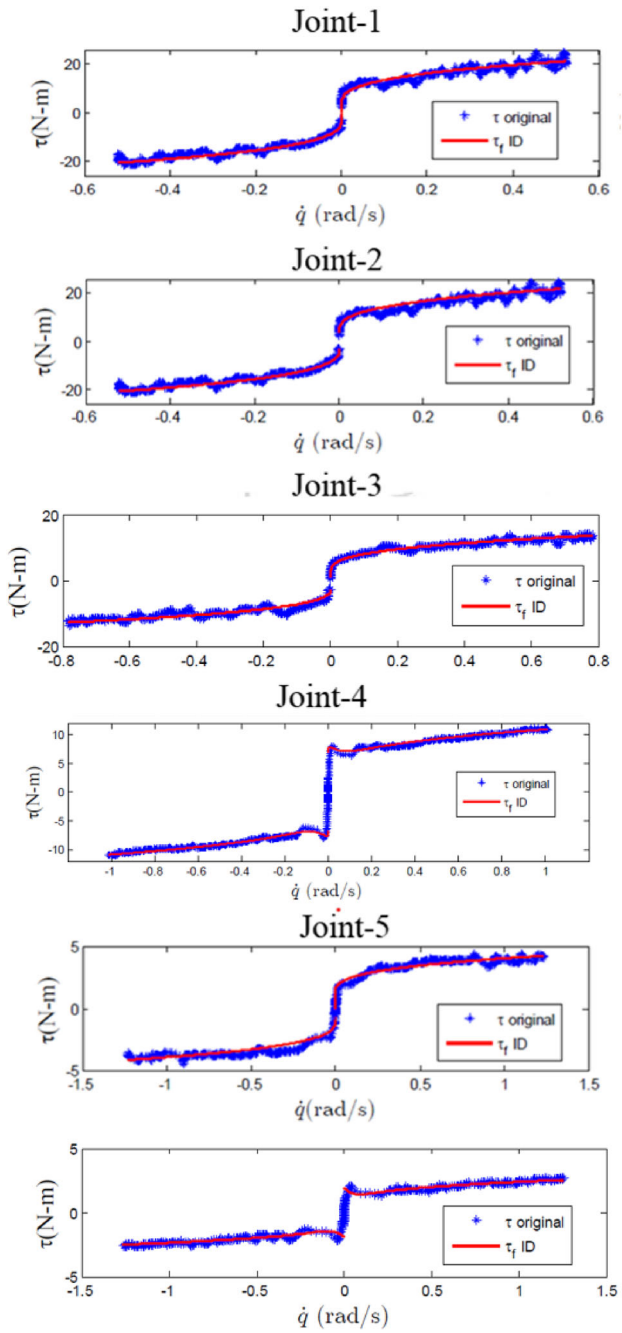

Fig. 4 The identified joint friction curves

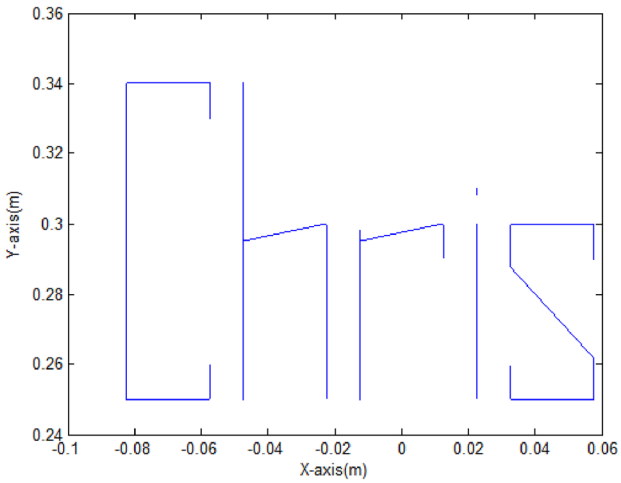

Fig. 5 Desired trajectory for simulation Case 1 ("Chris" text)

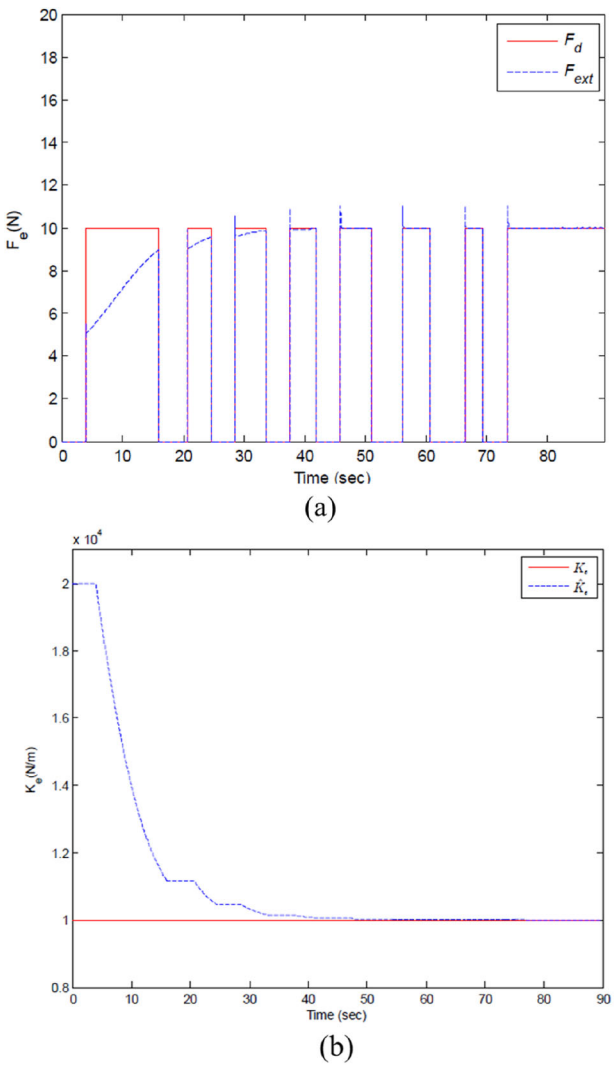

Fig. 6 Simulation result of Case 1: **a** Force control results, **b** Convergence of estimation for $\hat{K}_e$

$\phi(0) = -\pi$, $\theta(0) = 0°$, and $\psi(0) = 0°$. The initial estimated stiffness coefficient is $K_e = 20{,}000\,\text{N/m}$, while the actual environment stiffness coefficient is $K_e = 10{,}000\,\text{N/m}$. The learning parameter is set as $K_m = 100000$, and the simulation time is $t = [0, 89]$ sec. The desired force is 10 N. The simulation results are illustrated in Fig. 6b–d. Figure 6b presents the response of the force control, while the corresponding convergence of the estimated stiffness is depicted in Fig. 6c. Additionally, Fig. 6d illustrates the tracking con-

dition of the 6-axis in joint space. As shown in Fig. 6b–d (red solid-line: desired value, blue dashed-line: actual force), the response of the force control is happened when the end-effector contacts with environment. We can observe that the external force converges to desired force (Fd), the corresponding estimated stiffness coefficient also converges to actual stiffness coefficient. The results of force tracking and the position tracking are shown in Fig. 6d (blue solid-line: actual, red dashed-dotted-line: desired) and Fig. 7, the actual trajectory of 6-axis is following the desired trajectory, and satisfied our control goal.

Case 2 *Discussion on selection of initial estimated environment stiffness for force control.*

This simulation case aimed to verify the influence of the selection of initial estimated environment stiffness $\hat{K}_e$ in the force control phase. In this case, the robot was driven to draw a straight line trajectory in the X-direction while performing force control in the Z-direction, where the actual environment stiffness is $K_e = 10{,}000\,\mathrm{N/m}$. According to the proposed adaptive force control scheme, we chose three different initial values to verify the convergence of force tracking control and the environment stiffness estimation: (1) a relatively small initial environment stiffness estimation $\hat{K}_e = 500\,\mathrm{N/m}$, (2) a median initial environment stiffness estimation $\hat{K}_e = 15{,}000\,\mathrm{N/m}$, and (3) a relatively large initial environment stiffness estimation $\hat{K}_e = 20{,}000\,\mathrm{N/m}$. The learning rate is $\lambda = 0.1$, and the desired force is $F_d = 10\,\mathrm{N}$. The initial condition of the robot end-effector is $x(0) = -0.15\,\mathrm{m}$, $y(0) = 0.3694\,\mathrm{m}$, $z(0) = 0.2212\,\mathrm{m}$, $\phi(0) = -177°$, $\theta(0) = 0°$, and $\psi(0) = 22°$. The desired robot stiffness is chosen as $K_m = 100{,}000\,\mathrm{N/m}$. The total displacement of the desired trajectory is 0.3 m, and the end-effector of the robot contacted the X–Y plane at the beginning. The convergence of the environment stiffness estimation and the force tracking control are shown in Fig. 8. According to the simulation results, the proposed online stiffness estimation approach has the capability to approximately converge toward the actual environmental stiffness. However, in the force tracking results, we observed that the contact force has a larger overshoot when a smaller initial environment stiffness is chosen. Therefore, we recommend selecting a relatively high value for the initial environment stiffness to avoid hazards, even if the actual environment stiffness is unclear.

Case 3 *The robustness of joint friction compensation in FNNs.*

In the machinery field, friction effects are particularly critical for servo control, as the induced uncertainties might cause positioning errors, limit cycles, stick–slip motions, etc. [40,41]. In addition, during the identification experiments of the joint frictions of AR605, as mentioned above, we found that joint friction might change with operating conditions such as operating time and warm-up temperature. As a result, the objective of this simulation case was to evaluate the ability of FNNs in tolerating variations in joint frictions. In this case, we assume the ideal friction exists, $\tau_f^* = \tau_f^{ID} + \Delta\tau_f$, where τ_f^* is the ideal joint friction, τ_f^{ID} is the identified friction, and $\Delta\tau_f$ is the variations in joint friction. We employed three different friction compensation methods to verify the robustness of the proposed FNNs: (1) the proposed FNN-based friction approximation $\hat{\tau}_f$, (2) the identified velocity-dependent friction models, and (3) excluding the joint friction model in the proposed FNNs approximated dynamics. In this case, we assume the ideal joint friction has ten percent variations, that is, $\tau_f^* = \tau_f^{ID} \pm 0.1\tau_f^{ID}$. The comparison results of the position tracking errors for the X-direction line trajectory are shown in Fig. 9 (blue line: the proposed FNN friction model, red line: the identified friction model, pink line: without friction compensation), and the RMSE of each position tracking error is listed in Table 1. We observe that the proposed approach performs well with smaller tracking errors compared to directly using the identified friction model and the proposed FNNs without the friction model. As a result, the proposed approach has the advantage of substituting the conventional system identification for joint frictions, which involves cumbersome procedures, and the FNN-based approximation approach can be effectively applied to systems with uncertainties.

Table 1 Simulation of Case 3: RMS value of position tracking errors for different friction compensation methods

Friction compensation	X (mm)	Y (mm)	Z (mm)
FNN model	0.032	0.059	0.013
Identified model	0.9	0.14	0.21
Without compensation	4.9	0.86	1.2

Case 4 *Comparisons of the adaptive position control strategy between FNNs and BRBFNNs.*

In this simulation case, we compared the proposed FNN-based adaptive controller with an adaptive BRBFNN-based controller described in [10]. Both controllers have a similar network structure and are designed as a state-feedback sliding mode control scheme [32,42]. In the comparisons, a frequency-varying sinusoidal reference trajectory was employed on the joint-2 and joint-3 to evaluate the position tracking performances, the reference trajectories was designed as follows:

$$q_{d_2}(t) = a(t)\sin(2\pi f_0 b(t)t) \tag{36a}$$

$$q_{d_3}(t) = a(t)\cos(2\pi f_0 b(t)t) \tag{36b}$$

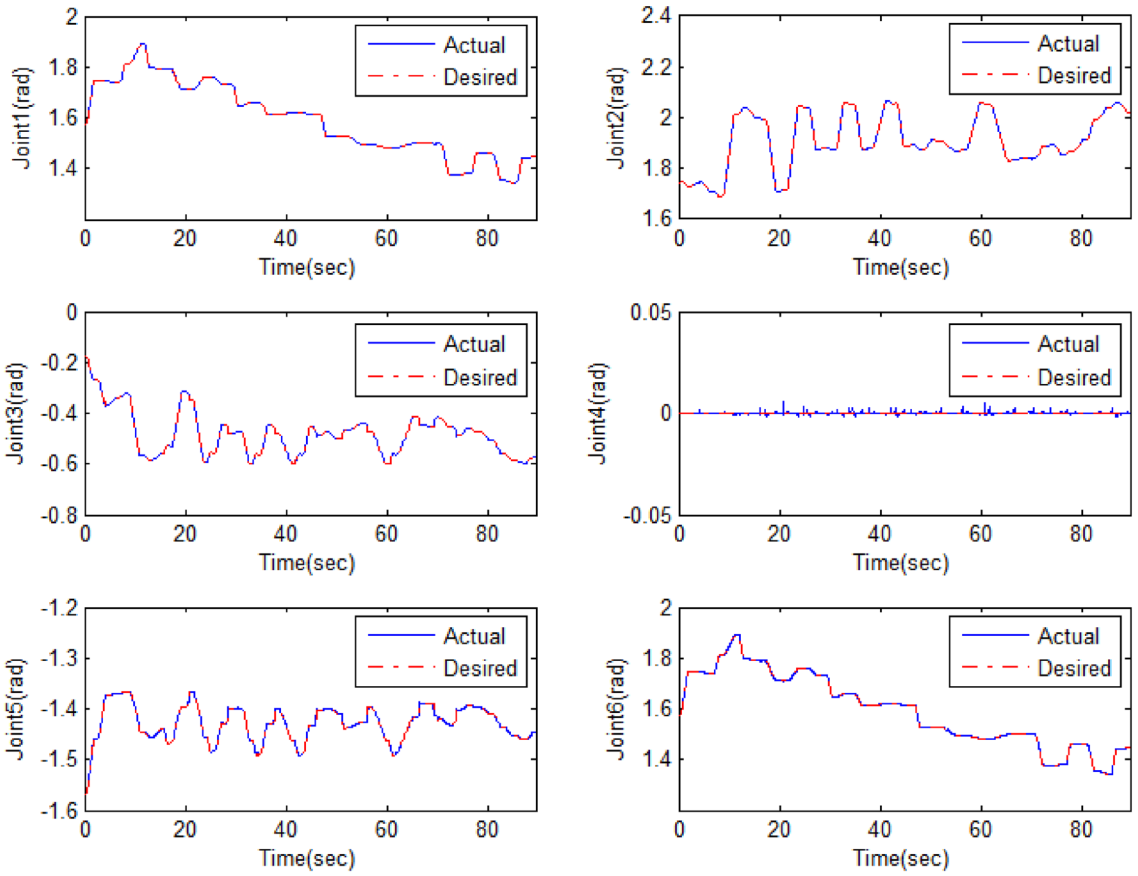

Fig. 7 Simulation result of Case 1: position Tracking results for each joint

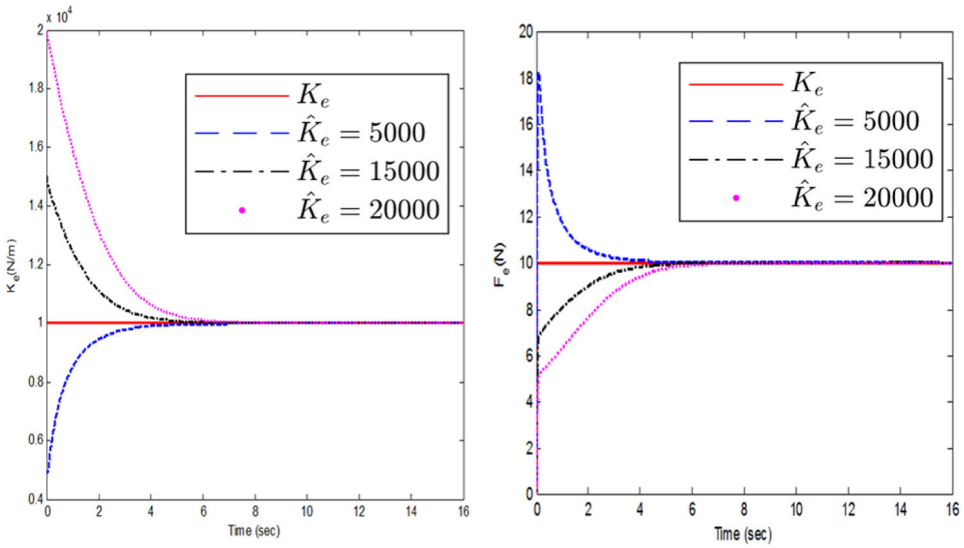

Fig. 8 Simulation of Case 2: convergences of environment stiffness estimation and force tracking results

where

$$a(t) = \begin{cases} (4/t_f)(t - t_0) & t \in [t_0, t_f/2] \\ (-4/t_f)(t - t_f/2) & t \in (t_f/2, t_f] \end{cases} \tag{36c}$$

$$b(t) = \begin{cases} (3/t_f)(t - t_0) & t \in [t_0, t_f/2] \\ (-3/t_f)(t - t_f/2) & t \in (t_f/2, t_f] \end{cases} \tag{36d}$$

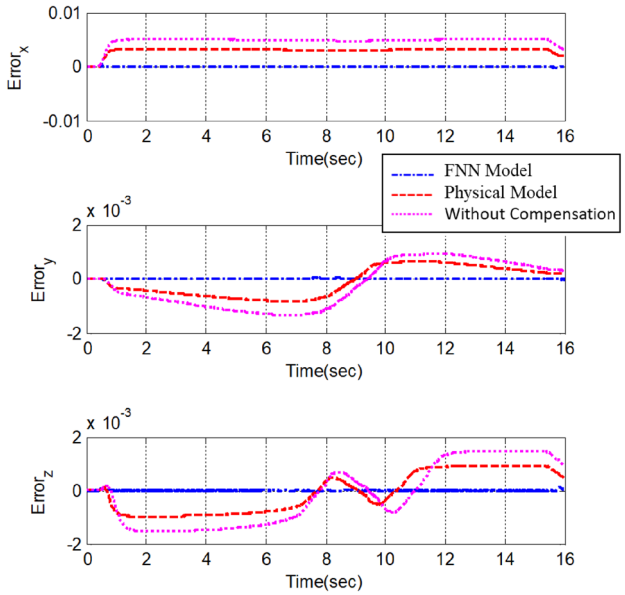

Fig. 9 Simulation of Case 3: position tracking errors for different joint friction compensation methods

Table 2 The RMS value of position tracking errors for FNN-based controller and BRBFNN-based controller (unit: rad)

Joint	FNN-based	BRBFNN-based [10]
J_2	0.01715	0.017012
J_2 with disturbances	0.01733	0.020642
J_3	0.016808	0.015678
J_3 with disturbances	0.016905	0.021771

Table 3 Comparison of the network parameter size between FNNs and BRBFNNs for n-link robot

Approximation model	Fuzzy membership node	Radial basis function node [10]
$\hat{\boldsymbol{M}}$	$p \times n$	–
$\hat{\boldsymbol{C}}$	$2p \times n$	–
$\hat{\boldsymbol{G}}$	$p \times n$	–
$\hat{\boldsymbol{\tau}}_f$	$p \times n$	–
Total	$5p \times n$	p^{3n}

where t_0 and t_f denote the initial time and total simulation time, respectively, while f_0 is the fundamental frequency of the sinusoidal trajectory. Subsequently, the the FNNs and the BRBFNNs were responsible for the role of dynamics estimators to estimate the unknown robot dynamics models. The total simulation time was 120 s. In order to evaluate the capabilities of both controllers to handle the inherent uncertainties of the dynamics and the disturbances, an 80 Nm external disturbance was applied to each joint for 10 s at the 50-s, Additionally, a 20% variation was conducted in the

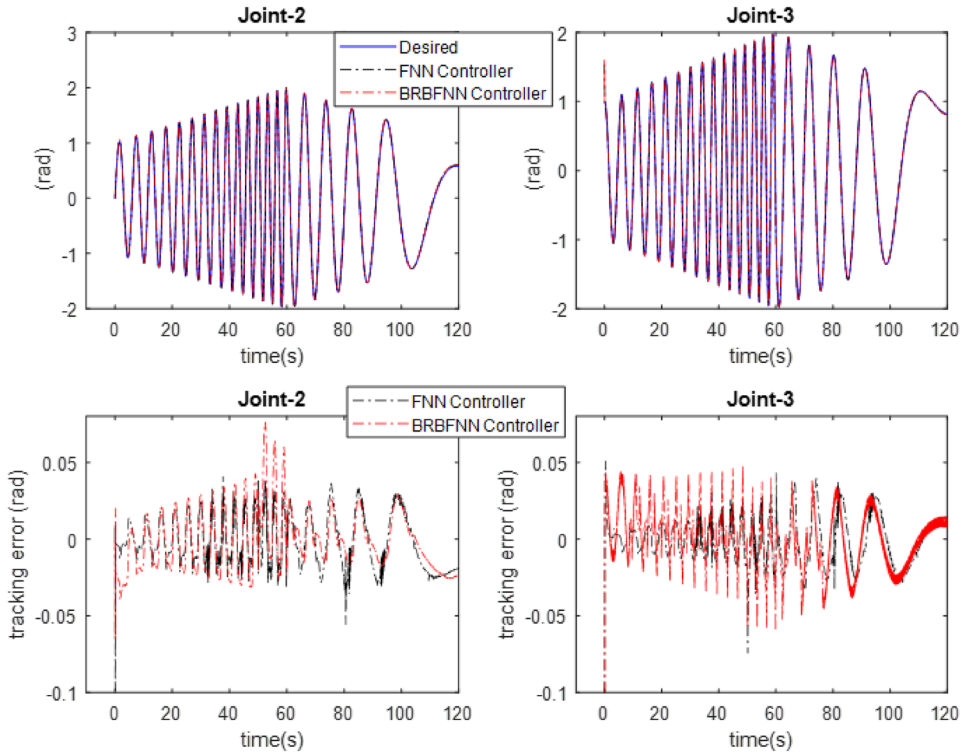

Fig. 10 Simulation of Case 4: position tracking performance of joints 2 and 3 under an external disturbance, and 20% variations in friction and gravity

joint frictions and gravity torques. In the dynamics estimators, we selected three fuzzy memberships and three radial basis nodes for each input variable (i.e. $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$) in FNNs and BRBFNNs, respectively. The centers and the width of Gaussian basis for each input variable were $[-1, 0, 1]$ and 1. In the adaptive controllers, we chose the same state-feedback gain $\boldsymbol{K_d} = \text{diag}(40, 30)$ and the same sliding surface slope $\boldsymbol{\Lambda} = \text{diag}(42, 42)$ for both the proposed FNN-based controller and the BRBFNN-based controller. The simulation results are illustrated in the Fig. 10 and the Table 2. In the Table 2, firstly, two controllers were operating under the ideal conditions without disturbances and variations in the dynamics, and both controllers performed good position tracking results. Subsequently, the external disturbances and the variation of the joint frictions and gravity torques were taken into account. As shown in the Table 2, the proposed controller exhibited slightly better robustness compared to the BRBFNN, although the latter also demonstrated a good ability to mitigate the disturbances. In the following Table 3, we compared the network parameter size between the FNNs and BRBFNNs, where the quantity n is the degree-of-freedom of dynamics system, the quantity p represents the fuzzy membership numbers and the numbers of radial basis node for each input variable in FNNs and BRBFNNs, respectively. As the comparison results, there is a significant difference in network size, the proposed FNNs have the capability to use a lighter network structure to achieve efficient tracking performance and maintain better robustness.

4.2 Experimental Results

In the experiment phases, the proposed adaptive controller is executed in the hard real-time system, RTX64, with a control cycle of 1 ms. The contact forces are measured by the 6-axis Force/Torque sensor, HEX-70-CE-2000N, produced by OPTOFORCE.

Case 5 *Force control on a straight line trajectory: $\hat{K}_e$ learning rate selection discussion.*

In this experimental case, we designed an interaction scenario, as shown in Fig. 3, where a screwdriver with a round head was used to make contact with the environment. The control goal was to maintain a constant contact force on the processing surface in the Z-direction. We selected a high-density sponge as the experimental environment, which has a higher stiffness coefficient to ensure the robot's stability. The initial position of the end-effector was $x(0) = -0.15\,\text{m}$, $y(0) = 0.3694\,\text{m}$, $z(0) = 0.2212\,\text{m}$, $\phi(0) = -177°$, $\theta(0) = 0°$, and $\psi(0) = 22°$. The desired robot stiffness and the initial estimated environment stiffness were chosen as $K_e = 20{,}000\,\text{N/m}$, $K_m = 100{,}000\,\text{N/m}$. The desired force was set to 10 N, the maximum feed rate was set to 0.02 m/s, and the control gains were chosen as $\boldsymbol{K}_d = \text{diag}(169, 169, 190, 402, 275, 248)$, and $\boldsymbol{\Lambda} = \text{diag}(84, 84, 95, 201, 137, 123)$. In the stiffness estimator, we employed three different learning rates for the gradient descent algorithm to compare the convergence of the force control: $\xi_{K_e} = 0.075$, $\xi_{K_e} = 0.1$, $\xi_{K_e} = 0.15$ and $\xi_{K_e} = 0.2$. The experimental results for different parameters of the constant target force control are shown in Fig. 11. Figure 11a shows the response of the force control, where the green star line indicates the contact forces without force control. The results of force tracking for each learning parameter show that the highest learning parameter (ξ_{K_e}) converges to the desired force the fastest. In addition, we observed a larger force generated at approximately 2 s; this is likely due to the plane of the workpiece and the plane of the robot manipulator not being horizontal, or the workpiece not being completely flat. This phenomenon demonstrates the robustness of our force control. The convergence of the estimated stiffness coeffi-

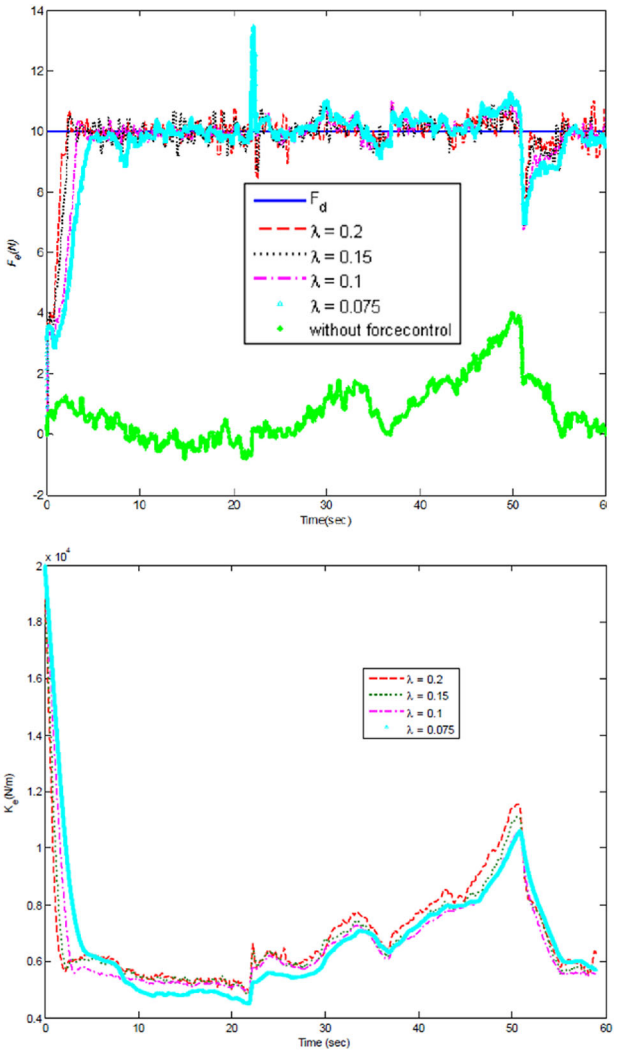

Fig. 11 Results of force control for Case 5: **a** force tracking results, **b** Convergence condition of K_e for different learning rate

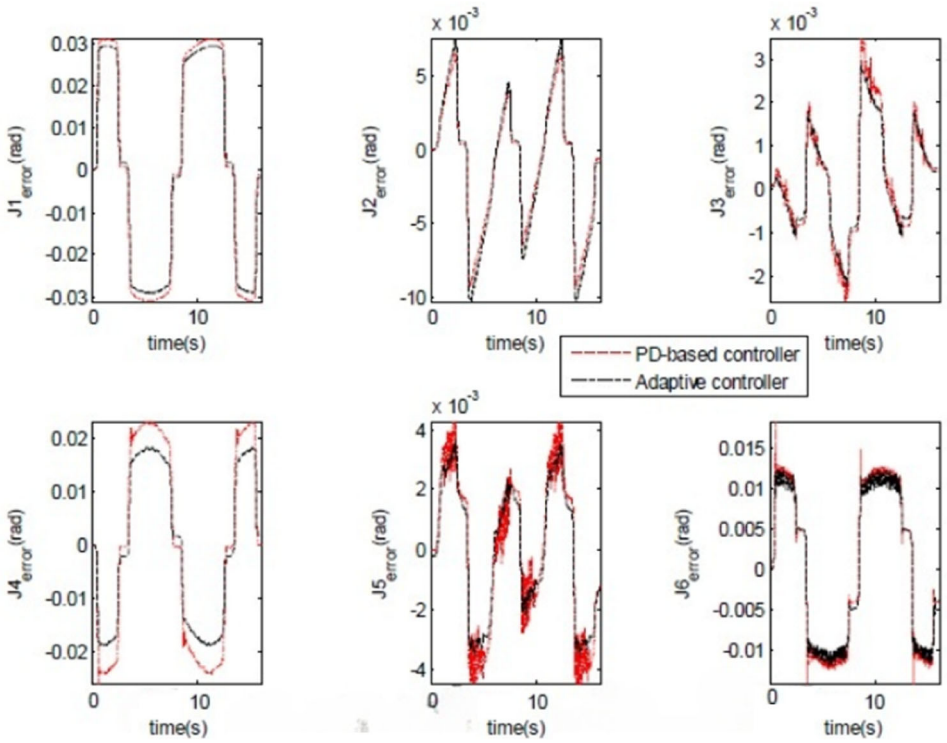

Fig. 12 Case 6: position tracking errors of two controllers for each joint

cient is shown in Fig. 11b. It is clear that the final values of the estimated stiffness coefficient are almost the same, with the difference in learning parameters affecting the rate of convergence. Force control can be achieved on an uneven level of the workpiece by changing the estimated stiffness coefficient. Fig. 11c shows the change in the learning rate.

Case 6 *Comparison of the position tracking between the proposed controller and PD controller.*

In this experimental case, we compare the performance of the position tracking results between the PD controller (the original controller of AR605) and the proposed adaptive controller in free-space motion. The desired trajectory is a straight-line path among the following five point-to-point positions: $(x, y, z) = (0.0, 0.46, 0.24)$, $(0.1, 0.46, 0.24)$, $(-0.1, 0.46, 0.24)$, $(0.1, 0.46, 0.24)$ m. The motion feed rate is set to 0.05 m/s. The orientation of the Euler angles is fixed at $\phi = -130°$, $\theta = 0°$, and $\psi = 0°$. The matrices of the control gains are selected as $\boldsymbol{K}_d = \text{diag}(169, 169, 190, 402, 275, 248)$, and $\boldsymbol{\Lambda} = \text{diag}(84, 84, 95, 201, 137, 123)$. The position tracking errors for the PD controller and the proposed adaptive controller are shown in Fig. 12. Table 4 shows the RMS values of tracking errors for the proposed FNN-based adaptive controller and the PD controller. Compared to the PD controller, all joints exhibited good tracking performance with the proposed adaptive controller except for the second joint was slightly inferior. In this experiment, the criteria for the control gains selection was focusing on the position tracking ability of the end-effector in Cartesian space. As a result, when adjusting the controller gains, we considered all robot joints tracking conditions for an overall adjustment.

Table 4 RMS of position tracking error for the PD controller and the proposed controller (unit for J_1–J_6: 10^{-3} rad; unit for X, Y, Z: mm)

controller	J_1	J_2	J_3	J_4	J_5	J_6	X	Y	Z
PD	26	4	1	19	3	11	11	0.5	1
FNN	24	4.5	1	15	2	9	10	0.4	1

5 Conclusion

This paper proposes an adaptive hybrid position/force control methodology for n-link robot manipulators operating in environments with unknown characteristics. FNNs are employed to mitigate the inherent model dynamics' uncertainty. Additionally, the gradient method is used to derive update laws for accurately estimating environmental stiffness coefficients. In motion control, the controller's stability and the update laws for the FNNs are rigorously established via Lyapunov theorem. Finally, we apply the developed adaptive force controller and path planning algorithm to an indus-

trial robot manipulator (model: AR605) developed by ITRI. The proposed control strategy's efficacy is validated through experimental results.

Funding Open Access funding enabled and organized by National Yang Ming Chiao Tung University

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

References

1. Al-Khedher, M.A., Alshanmasin, M.S.: SCARA robot control using neural networks. In: Int. Conf. Intell. Adv. Sys., pp. 126–130. Amman (2012). https://doi.org/10.1109/ICIAS.2012.6306173
2. Duleba, I., Opalka, M.: A comparison of Jacobian-based methods of inverse kinematics for serial robot manipulators. Int. J. Appl. Math. Comput. Sci. **23**(2), 373–382 (2013). https://doi.org/10.2478/amcs-2013-0028
3. Du, Y.Y.: ITRI robot dynamic controller: introduction of user defined function. J. Mechatron. Ind. **400**, 17–28 (2016). (**In Chinese**)
4. Grinits, E.V., Bottura, C.P.: Adaptive neural-based backstepping control of uncertain MIMO nonlinear systems. In: Int. Joint Conf. Neural Networks, pp. 4468–4475, BC, Canada (2006). https://doi.org/10.1109/IJCNN.2006.1716719
5. Lee, C.H., Chung, B.R.: Adaptive backstepping controller design for nonlinear uncertain system using fuzzy neural systems. Int. J. Sys. Sci. **43**(10), 1855–1869 (2012). https://doi.org/10.1080/00207721.2011.554915
6. Olsson, H., Astrom, K.J., de Wit, C.C., Gafvert, M., Lischinsky, P.: Friction models and friction compensation. Eur. J. Control **4**(3), 176–195 (1997). https://doi.org/10.1016/S0947-3580(98)70113-X
7. Tanaka, Y., Tsuji, T.: On-line learning of robot arm impedance using neural networks. In: IEEE Int. Conf. Robotics Biomimetics, pp. 941–946, China (2004). https://doi.org/10.1016/j.robot.2005.06.005
8. Wai, R.J., Muthusamy, R.: Fuzzy-neural-network inherited sliding-mode control for robot manipulator including actuator dynamics. IEEE Trans. Neural Learn. Sys. **24**(2), 274–287 (2013). https://doi.org/10.1109/TNNLS.2012.2228230
9. Wang, L., Chai, T., Yang, C.: Neural-network-based contouring control for robot manipulators in operational space. IEEE Trans. Control Sys. Technol. **20**(4), 1073–1080 (2012). https://doi.org/10.1109/TCST.2011.2147316
10. Liu, Q., Li, D., Ge, S.S., Ji, R., Ouyang, Z., Tee, K.P.: Adaptive bias RBF neural network control for a robotic manipulator. Neurocomputing **447**, 213–223 (2021). https://doi.org/10.1016/j.neucom.2021.03.033
11. Xu, S., Li, D., Yang, C., Huang, C., Wu, X.: A robot motion learning method using broad learning system verified by small-scale fish-like robot. IEEE Trans. Cybern. **53**, 6053–6065 (2023). https://doi.org/10.1109/TCYB.2023.3269773
12. Huang, H., Yang, C., Chen, C.L.: Optimal robot–environment interaction under broad fuzzy neural adaptive control. IEEE Trans. Cybern. **51**(7), 3824–3835 (2021). https://doi.org/10.1109/TCYB.2020.2998984
13. He, W., Kong, L., Dong, Y., Yu, Y., Yang, C., Sun, C.: Neural network control-based adaptive learning design for nonlinear systems with full-state constraints. IEEE Trans. Syst. Man Cybern. **49**, 543–554 (2019). https://doi.org/10.1109/TSMC.2017.2749124
14. Ren, Y., Sun, Y., Liu, L.: Fuzzy disturbance observers-based adaptive fault-tolerant control for an uncertain constrained automatic flexible robotic manipulator. IEEE Trans. Fuzzy Syst. **32**, 1144–1158 (2023). https://doi.org/10.1109/TFUZZ.2023.3319392
15. Lee, C.H., Wang, W.C.: Robust adaptive position and force controller design of robot manipulator using fuzzy neural networks. Nonlinear Dyn. **85**(1), 343–354 (2016). https://doi.org/10.1007/s11071-016-2689-1
16. Xu, S., Wu, Z.: Adaptive learning control of robot manipulators via incremental hybrid neural network. Neurocomputing **568**, 127045 (2024). https://doi.org/10.1016/j.neucom.2023.127045
17. Chen, C.L.P., Liu, Z.: Broad learning system: an effective and efficient incremental learning system without the need for deep architecture. IEEE Trans. Neural Netw. Learn. Syst. **29**, 10–24 (2018). https://doi.org/10.1109/TNNLS.2017.2716952
18. Karayiannidis, Y., Doulgeri, Z.: Blind force/position control on unknown planar surfaces. IET Control Theory Appl. **3**, 595–603 (2009). https://doi.org/10.1049/iet-cta.2008.0239
19. Yu Jhan, Z., Lee, C.H.: Adaptive impedance force controller design for robot manipulator including actuator dynamics. Int. J. Fuzzy Syst. **19**, 1739–1749 (2017). https://doi.org/10.1007/s40815-017-0358-2
20. Coutinho, F., Cortesao, R.: Comparison of position and force-based techniques for environment stiffness estimation in robotic tasks. In: IEEE Int. Conf. Intell. Robots Sys., pp. 4933–4938 (2012). https://doi.org/10.1109/IROS.2012.6386161
21. He, W., Dong, Y., Sun, C.: Adaptive neural impedance control of a robotic manipulator with input saturation. IEEE Trans. Syst. **46**(3), 334–344 (2016). https://doi.org/10.1109/TSMC.2015.2429555
22. Jung, S., Hsia, T.C., Bonitz, R.G.: Force tracking impedance control of robot manipulators under unknown environment. IEEE Trans. Control Syst. Technol. **12**(3), 474–483 (2004). https://doi.org/10.1109/TCST.2004.824320
23. Mallapragada, V., Erol, D., Sarkar, N.: A new method of force control for unknown environments. In: IEEE Int. Conf. Intell. Robots Sys., pp. 4509–4514, Beijing, China (2006). https://doi.org/10.1109/IROS.2006.282089
24. Mendes, N., Neto, P.: Indirect adaptive fuzzy control for industrial robots: a solution for contact applications. Expert Syst. Appl. **42**(22), 8929–8935 (2015). https://doi.org/10.1016/j.eswa.2015.07.047
25. Raibert, M.H., Craig, J.J.: Hybrid position/force control of manipulators. J. Dyn. Syst. Meas. Control **103**(2), 2–12 (1981)
26. Roy, J., Whitcomb, L.L.: Adaptive force control of position/velocity controlled robots: theory and experiment. IEEE Trans. Robotics Autom. **18**(2), 121–137 (2002). https://doi.org/10.1109/TRA.2002.999642
27. Wang, F., Luo, Z., Liu, H., Wang, L.: Impedance model based fuzzy force control for robot manipulators. In: IEEE Int. Conf.

on Intelligent Robots and Systems (2010). https://doi.org/10.1109/ROBIO.2010.5723561
28. Cui, D., Chadli, M., Xiang, Z.: Fuzzy fault-tolerant predefined-time control for switched systems: a singularity-free method. IEEE Trans. Fuzzy Syst. **32**(3), 1223–1232 (2024). https://doi.org/10.1109/TFUZZ.2023.3321688
29. Cui, D., Ahn, C.K., Sun, Y., Xiang, Z.: Mode-dependent state observer-based prescribed performance control of switched systems. IEEE Trans. Circuits Syst. II Express Briefs **71**(8), 3810–3814 (2024). https://doi.org/10.1109/TCSII.2024.3370865
30. Yen, S.H., Tang, P.C., Lin, Y.C., Lin, C.Y.: Development of a virtual force sensor for a low-cost collaborative robot and applications to safety control. Sensors **19**(11), 2603 (2019). https://doi.org/10.3390/s19112603
31. Spong, M.W., Hutchinson, S., Vidyasagar, M.: Robot Modeling and Control. John Wiley & Sons, New York (2020)
32. Slotine, J.E., Li, W.: Applied Nonlinear Control. Prentice Hall, Englewood Cliffs (1991)
33. Mazare, M., Tolu, S., Taghizadeh, M.: Adaptive variable impedance control for a modular soft robot manipulator in configuration space. Meccanica **57**(1), 1–15 (2022). https://doi.org/10.1007/s11012-021-01436-x
34. Zhu, Z., Zhang, H., Liu, G., Zhang, D.: Position-based force tracking adaptive impedance control strategy for robot grinding complex surfaces system. J. Field Robot **40**(5), 1097–1114 (2023). https://doi.org/10.1002/rob.22169
35. Duan, J., Gan, Y., Chen, M., Dai, X.: Adaptive variable impedance control for dynamic contact force tracking in uncertain environment. Robot. Auton. Syst. **102**, 54–65 (2018). https://doi.org/10.1016/j.robot.2018.01.009
36. Bilal, H., Yin, B., Kumar, A., Ali, M., Zhang, J., Yao, J.: Jerk-bounded trajectory planning for rotary flexible joint manipulator: an experimental approach. Soft. Comput. **27**(7), 4029–4039 (2023). https://doi.org/10.1007/s00500-023-07923-5
37. Fang, Y., Qi, J., Hu, J., Wang, W., Peng, Y.: An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints. Mech. Mach. Theory **153**, 103957 (2020). https://doi.org/10.1016/j.mechmachtheory.2020.103957
38. Chiang, J.Y.: Adaptive force controller design and implementation for robot manipulator with uncertainty and friction. Master thesis, Dept. Mech. Eng., National Chung Hsing University, Taichung city, Taiwan (2017)
39. Waiboer, R.: Dynamic modeling, identification and simulation of industrial robots. PhD thesis, University of Twente (2007)
40. Simoni, L., Beschi, M., Legnani, G., Visioli, A.: Friction modeling with temperature effects for industrial robot manipulators. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3524–3529. IEEE (2015)
41. Bona, B., Indri, M.: Friction compensation in robotics: an overview. In: Proceedings of the 44th IEEE Conference on Decision and Control, pp. 4360–4367. IEEE (2005)
42. Sabanovic, A.: Variable structure systems with sliding modes in motion control—a survey. IEEE Trans. Ind. Inform. **7**(2), 212–223 (2011). https://doi.org/10.1109/TII.2011.2123907

Bo-Ru Tseng was born in Taichung, Taiwan, in 1993. He received the B.S. degree from the Department of Industrial Education and Technology, National Changhua University of Education, Changhua, Taiwan, in 2015, and the M.S. degree from the Department of Mechanical Engineering, National Chung Hsing University, Taichung, Taiwan, in 2021. He worked at the UNAN SEIKI Co. for two years from 2017 and was responsible for PLC programming and mechatronics engineering. He is currently pursuing the Ph.D. degree with the Intelligent Control and Applications Laboratory, Institute of Electrical and Control Engineering, National Yang Ming Chiao Tung University, Hsinchu, Taiwan. His current research interests include robotics control, manipulator force control, signal processing, and optimization.

Jun-Yi Jiang was born in Nantou, Taiwan, in 1993. He received the B.S. degree from the Department of Vehicle Engineering, National Pingtung University of Science and Technology, Pingtung, Taiwan, and the M.S. degree from the Department of Mechanical Engineering, National Chung Hsing University, Taichung, Taiwan. After graduation, he worked at the Industrial Technology Research Institute for five years and then at Expetech for one year. He is currently a research and development engineer at OSCARMAX. His research interests include software development for automation production lines.

Ching-Hung Lee (Senior Member, IEEE) was born in Taiwan in 1969. He received his B.S. and M.S. degrees from the Department of Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, in 1992 and 1994, respectively, and his Ph.D. degree from the Department of Electrical and Control Engineering, National Chiao Tung University, in 2000. He is currently the Ding-Hua Hu Chair Professor and Distinguished Professor in the Department of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Hsinchu. He received the Researcher Excellence Award and the Wu Ta-Yu Medal (Young Researcher Award) from the National Science and Technology Council, Taiwan, in 2023 and 2008, respectively. He was also awarded the Fellow, Youth, and Excellent Automatic Control Engineering Awards from the Chinese Automatic Control Society, Taiwan, in 2019, 2009, and 2016, respectively. His research interests include artificial intelligence, smart manufacturing, fuzzy neural systems, signal processing, nonlinear control systems, robotics control, and CNC motion control and optimization.