Iterative Learning for Gravity Compensation in Impedance Control

Teng Li, *Graduate Student Member, IEEE*, Amir Zakerimanesh, Yafei Ou, Armin Badre, and Mahdi Tavakoli, *Senior Member, IEEE*

***Abstract*—Robot-assisted arthroscopic surgery has been increasingly receiving attention in orthopedic surgery. To build a robot-assisted system, dynamic uncertainties can be a critical issue that could bring robot performance inaccuracy or even system instability if cannot be appropriately compensated. Disturbance observer is a common tool to be used for disturbance estimation and compensation by taking all uncertainties as disturbances, but this will refuse human–robot interaction since the human-applied force will also be regarded as a disturbance by the observer. Iterative learning for gravity compensation can be another promising way to solve this problem when gravity compensation is the main concern. In this article, a gravity iterative learning (Git) scheme in Cartesian space for gravity compensation, integrating with an impedance controller, is presented. A steady-state scaling strategy is then proposed, which released the updating requirements of the learning scheme and also extended its validity to trajectory-tracking scenarios from set-point regulations. The deriving process and convergence properties of the Git scheme are presented and theoretically analyzed, respectively. A series of simulations and physical experiments are conducted to evaluate the validity of the scaling strategy, the learning accuracy of the Git scheme, and the effectiveness of the learning-based impedance controller. Both simulation and experimental results demonstrate good performance and properties of the Git scheme and the learning-based impedance controller.**

***Index Terms*—Gravity compensation, impedance control, iterative learning, physical human–robot interaction, robot-assisted arthroscopy.**

I. Introduction

ROBOT-ASSISTED minimally invasive surgery (MIS) has been becoming increasingly popular across various surgical specialties, such as orthopedics [1]. MIS can bring the benefits of a faster recovery rate and decreased pain to patients thus getting more favor. Robot-assisted surgeries are transforming traditional orthopedic surgeries by helping surgeons achieve more successful and precise surgical outcomes with the assistance of robots [2], [3], [4]. Elbow arthroscopy is a typical type of MIS in orthopedics that allows the management of elbow stiffness, arthritis, and fractures in a minimally invasive fashion [5]. During traditional elbow arthroscopy, the surgeon needs to hold an arthroscope with one hand while performing the surgical operations with the other hand, which can restrict the dexterity of the surgical performance and increase the cognitive load. This arouses the necessity to develop a robot-assisted arthroscope holder where the robot can hold the arthroscope for the surgeon during the surgery.

To build a robot-assisted system for assisting surgeons in holding with arthroscope during orthopedic surgery, some requirements need to be satisfied [6]. First, The robot can hold the arthroscope still at a specified pose (i.e., set-point regulation) while rejecting all possible disturbances (e.g., external disturbances delivered to the arthroscope via contact with the patient's body during surgery). Second, when the surgeon needs to move the arthroscope to a new pose (e.g., for adjusting the scope view perspective), the robot should allow the surgeon to move it around freely (i.e., human–robot interaction). Then, when a new pose is determined by the surgeon, the robot should keep the arthroscope still again while rejecting any disturbances. The main problem to build such a robot-assisted system is the dynamic model uncertainties and external disturbances, which could largely affect the robot's task performance accuracy and even stability if they are not appropriately compensated. More specifically, incomplete gravity compensation can be the main issue in this case since heavy surgical tools with unknown weights will be attached to the robot end-effector (EE).

Manuscript received 22 August 2023; revised 13 December 2023 and 6 March 2024; accepted 1 April 2024. Recommended by Technical Editor M. Rakotondrabe and Senior Editor K. J. Kyriakopoulos. This work was supported in part by the Canada Foundation for Innovation (CFI), in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, in part by the Canadian Institutes of Health Research (CIHR), in part by the Alberta Jobs, Economy and Innovation Ministry's Major Initiatives Fund to the Center for Autonomous Systems in Strengthening Future Communities, and in part by the Edmonton Civic Employee Charitable Assistance Fund. *(Corresponding author: Mahdi Tavakoli.)*

Teng Li, Amir Zakerimanesh, and Yafei Ou are with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: teng4@ualberta.ca; zakerima@ualberta.ca; yafei6@ualberta.ca).

Armin Badre is with the Western Hand and Upper Limb Facility, Sturgeon Hospital, St. Albert, AB T8N 6C4, Canada, and also with the Division of Orthopaedic Surgery, Department of Surgery, Faculty of Medicine and Dentistry, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: badre@ualberta.ca).

Mahdi Tavakoli is with the Department of Electrical and Computer Engineering and the Department of Biomedical Engineering, University of Alberta, Edmonton, AB T6G 1H9, Canada (e-mail: mahdi.tavakoli@ualberta.ca).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2024.3386407.

Digital Object Identifier 10.1109/TMECH.2024.3386407

Disturbance observer is a promising way to estimate and compensate for dynamic uncertainties. In our previous work [6], we have shown that by integrating impedance control and nonlinear disturbance observer (NDOB), an accurate impedance control can be achieved. In that work, the disturbance observer can accurately estimate and compensate for the lumped uncertainties including incomplete gravity compensation. However, the NDOB as well as other types of observers [7], such as generalized momentum observer (GMO) [8], joint velocity observer [8], extended state observer [9], and disturbance Kalman filter method [10], [11], always estimate a lumped uncertainty term and is not able to separate out any one component when several uncertainty sources exist. Moreover, the observer will refuse human–robot interaction since human-applied force will be taken as a part of the lumped disturbances thus being rejected.

Learning control has been developed to track repetitive trajectories for both rigid and nonrigid robots. De Luca and Ulivi presented a simple and efficient iterative learning algorithm for robots with joint elasticity [12]. In their work, a learning term was used to learn the necessary modification to the desired trajectory position. They demonstrated the algorithm's usefulness by good motion performance of simulations on a two-link planar robot. Based on a similar design methodology, an iterative learning scheme for gravity compensation in set-point regulation problems was initially proposed by De Luca and Panzieri [13], [14]. The learning scheme completes the required gravity compensation at the final steady state in set-point regulation tasks. It can iteratively learn the constant gravity without the need of introducing an integral error term or using high-gain feedback.

Based on the same contraction mapping theorem, Basovich et al. [15] developed an iterative output feedback controller for a 6-degree-of-freedom (DOF) precision positioning system when only position measurement is available. Their proposed controller can learn and compensate for the payload uncertainties with bounded error in set-point control tasks. Ji et al. [16] used the iterative learning method to autocalibrate gravity compensation when the robot has no contact with the environment, thus making the robot EE weightless.

Incomplete or absent gravity compensation will cause a constant steady-state error [13]. For impedance control, making the robot "stiffer" by tuning up the impedance gains can reduce the error to some extent, but not eliminate it. Especially when heavy but unknown external payloads are attached to the robot EE, the method of tuning up impedance gains will be largely limited and be difficult to achieve satisfying results.

In summary, in our target application scenario, i.e., robot-assisted arthroscopic surgery, gravity compensation, and physical human–robot interaction (pHRI) are the main concerns. There are various disturbance observers available for gravity compensation [7], e.g., NDOB, GMO, etc. However, the output of an observer is a lumped estimate on all uncertainties including gravity, and it will refuse human–robot interaction by taking it as a part of uncertainties [6]. Furthermore, it also requires the estimated dynamic parameters of the robot dynamics [7]. An adaptive controller [17], [18] can also deal with dynamic uncertainties including gravity. However, it is a controller rather than an independent approach for disturbance estimation, and it cannot provide compliant robot behavior for a safe human–robot interaction like an impedance controller can do. Therefore, a simple method that can focus on gravity compensation while enabling pHRI and avoiding the necessity of the robot dynamics is needed in our scenario.

Inspired by [13], in this article, we proposed a gravity iterative learning (Git) scheme for gravity compensation in Cartesian space and integrated it with an impedance controller. The convergence properties of the Git scheme are theoretically analyzed. The learning performance and effectiveness are then evaluated by a series of simulations and experiments in both trajectory tracking tasks and set-point regulation tasks. Finally, an application experiment in pHRI scenario is presented to show the effectiveness of the integrated controller. The main contributions in this work can be described as the following.

1) An adapted iterative learning scheme for gravity compensation in Cartesian space is presented, and the converging properties are theoretically analyzed.
2) A steady-state scaling strategy is proposed, which enables the iterative learning update law to be executed in each servo loop, and more importantly, it extends the validity of the learning scheme to general trajectory-tracking scenarios.

The rest of this article is organized as follows. Section II is devoted to introducing the proposed iterative learning scheme in impedance control. Section III presents simulations, experiments, and corresponding results in various scenarios for evaluating the scheme. Finally, Section IV concludes this article.

II. Methods

A. Robot Dynamics and Disturbances

A general dynamic model for an n-DOF rigid robot with revolute joints [19] can be given by

$$\underbrace{\mathbf{M}(\mathbf{q})}_{\hat{\mathbf{M}}+\Delta\mathbf{M}}\ddot{\mathbf{q}}+\underbrace{\mathbf{S}(\mathbf{q},\dot{\mathbf{q}})}_{\hat{\mathbf{S}}+\Delta\mathbf{S}}\dot{\mathbf{q}}+\underbrace{\mathbf{G}(\mathbf{q})}_{\hat{\mathbf{G}}+\Delta\mathbf{G}}+\boldsymbol{\tau}_{\mathbf{fric}}(\dot{\mathbf{q}})=\boldsymbol{\tau}+\underbrace{\boldsymbol{\tau}_{\mathbf{ext}}}_{\mathbf{J}^{\mathbf{T}}\mathbf{F}_{\mathbf{ext}}} \quad (1)$$

where $\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}\in\mathbb{R}^n$ are the joint position, velocity, and acceleration, respectively, $\mathbf{M}\in\mathbb{R}^{n\times n}$ 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 users' model estimates, while $\Delta\mathbf{M},\Delta\mathbf{S},\Delta\mathbf{G}$ are the corresponding estimate errors. $\boldsymbol{\tau}_{\mathbf{fric}}\in\mathbb{R}^n$ is joint friction, $\boldsymbol{\tau}\in\mathbb{R}^n$ is the commanded joint torque vector, $\boldsymbol{\tau}_{\mathbf{ext}}\in\mathbb{R}^n$ is the torque caused by external force, $\mathbf{F}_{\mathbf{ext}}\in\mathbb{R}^6$ is the external force in Cartesian space, and $\mathbf{J}\in\mathbb{R}^{6\times n}$ is the Jacobian matrix.

By collecting all the disturbances together, the dynamic model (1) of a robot can be rewritten as

$$\hat{\mathbf{M}}\ddot{\mathbf{q}}+\hat{\mathbf{S}}\dot{\mathbf{q}}+\hat{\mathbf{G}}=\boldsymbol{\tau}+\underbrace{\boldsymbol{\tau}_{\mathbf{ext}}-[\boldsymbol{\tau}_{\mathbf{fric}}+(\Delta\mathbf{M}\ddot{\mathbf{q}}+\Delta\mathbf{S}\dot{\mathbf{q}}+\Delta\mathbf{G})]}_{\boldsymbol{\tau}_{\mathbf{dist}}} \quad (2)$$

where $\boldsymbol{\tau}_{\mathbf{dist}}$ denotes the lumped uncertainties containing the model error $(\Delta\mathbf{M}\ddot{\mathbf{q}}+\Delta\mathbf{S}\dot{\mathbf{q}}+\Delta\mathbf{G})$, the joint friction $\boldsymbol{\tau}_{\mathbf{fric}}$, and the external disturbances $\boldsymbol{\tau}_{\mathbf{ext}}$.

In this article, we will focus on estimating and compensating for the gravity caused by external constant payloads using an iterative learning method. In order to clearly reveal the behavior of the iterative learning algorithm to learn the gravity of the external payloads, in the simulations, we assume that

a) an ideal dynamic model is available, i.e., $\hat{\mathbf{M}} = \mathbf{M}, \hat{\mathbf{S}} = \mathbf{S}, \hat{\mathbf{G}} = \mathbf{G}$, thus, $\Delta\mathbf{M} = \mathbf{0}, \Delta\mathbf{S} = \mathbf{0}, \Delta\mathbf{G} = \mathbf{0}$;
b) no joint friction, i.e., $\boldsymbol{\tau}_{\mathbf{fric}} = \mathbf{0}$;
c) only constant payloads exists for external disturbances. By applying these assumptions, the dynamic model (2) will become (3)

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{S}\dot{\mathbf{q}} + \mathbf{G} = \boldsymbol{\tau} + \underbrace{\mathbf{J}^{\mathbf{T}}\mathbf{F}_{\mathbf{ext}}}_{\boldsymbol{\tau}_{\mathbf{dist}}}. \tag{3}$$

Model (3) can be expressed in Cartesian space as

$$\mathbf{M}_{\mathbf{x}}\ddot{\mathbf{x}} + \mathbf{S}_{\mathbf{x}}\dot{\mathbf{x}} + \mathbf{G}_{\mathbf{x}} = \mathbf{J}^{-\mathbf{T}}\boldsymbol{\tau} + \mathbf{F}_{\mathbf{ext}} \tag{4}$$

where $\mathbf{M}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}}, \mathbf{G}_{\mathbf{x}}$ have

$$\begin{cases} \mathbf{M}_{\mathbf{x}} = \mathbf{J}^{-\mathbf{T}}\mathbf{M}\mathbf{J}^{-\mathbf{1}} \\ \mathbf{S}_{\mathbf{x}} = \mathbf{J}^{-\mathbf{T}}\mathbf{S}\mathbf{J}^{-\mathbf{1}} - \mathbf{M}_{\mathbf{x}}\dot{\mathbf{J}}\mathbf{J}^{-\mathbf{1}} \\ \mathbf{G}_{\mathbf{x}} = \mathbf{J}^{-\mathbf{T}}\mathbf{G} \end{cases} \tag{5}$$

where $\mathbf{M}_{\mathbf{x}}, \mathbf{S}_{\mathbf{x}}, \mathbf{G}_{\mathbf{x}}$ are the $\mathbf{M}, \mathbf{S}, \mathbf{G}$ expressed in Cartesian space, respectively.

B. Impedance Control

A desired impedance model [6], [20], [21] for robot-environment interaction can be expressed as

$$\begin{aligned} \mathbf{F}_{\mathbf{imp}} = {} & \mathbf{M}_{\mathbf{m}}(\ddot{\mathbf{x}} - \ddot{\mathbf{x}}_{\mathbf{d}}) \\ & + (\mathbf{S}_{\mathbf{x}} + \mathbf{D}_{\mathbf{m}})(\dot{\mathbf{x}} - \dot{\mathbf{x}}_{\mathbf{d}}) + \mathbf{K}_{\mathbf{m}}(\mathbf{x} - \mathbf{x}_{\mathbf{d}}) \end{aligned} \tag{6}$$

where $\mathbf{M}_{\mathbf{m}}, \mathbf{D}_{\mathbf{m}}, \mathbf{K}_{\mathbf{m}}$ are user-designed matrices for inertia, damping, and stiffness, respectively. Note that $\mathbf{x}_{\mathbf{d}}, \dot{\mathbf{x}}_{\mathbf{d}}, \ddot{\mathbf{x}}_{\mathbf{d}}$ are the desired position, velocity, and acceleration, respectively, in Cartesian space, while $\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}$ are the actual ones. $\mathbf{F}_{\mathbf{imp}}$ is the interaction force between the robot and the environment.

To avoid the measurement of external forces, the designed inertia matrix will be set as the inherent inertia matrix of the robot, i.e., $\mathbf{M}_{\mathbf{m}} = \mathbf{M}_{\mathbf{x}}$. Then, by substituting (6) into (4) with $\mathbf{F}_{\mathbf{ext}} = \mathbf{F}_{\mathbf{imp}}$, the impedance control law can be given by [6]

$$\begin{aligned} \boldsymbol{\tau} = {} & \mathbf{M}\mathbf{J}^{-\mathbf{1}}(\ddot{\mathbf{x}}_{\mathbf{d}} - \dot{\mathbf{J}}\mathbf{J}^{-\mathbf{1}}\dot{\mathbf{x}}_{\mathbf{d}}) + \mathbf{S}\mathbf{J}^{-\mathbf{1}}\dot{\mathbf{x}}_{\mathbf{d}} + \mathbf{G} \\ & + \mathbf{J}^{\mathbf{T}}[\mathbf{D}_{\mathbf{m}}(\dot{\mathbf{x}}_{\mathbf{d}} - \dot{\mathbf{x}}) + \mathbf{K}_{\mathbf{m}}(\mathbf{x}_{\mathbf{d}} - \mathbf{x})]. \end{aligned} \tag{7}$$

Note that when implementing the impedance controller (7) in practice for physical experiments, the estimates $\hat{\mathbf{M}}, \hat{\mathbf{S}}, \hat{\mathbf{G}}$ will be used for the calculation since an accurate model of a physical robot is usually not available.

For moving robot EE to a fixed point, i.e., set-point regulation, we have $\dot{\mathbf{x}}_{\mathbf{d}} = \mathbf{0}, \ddot{\mathbf{x}}_{\mathbf{d}} = \mathbf{0}$. Then, the impedance control law (7) can be simplified and reduced to (8), which is also known as task-space proportional-derivative (PD) controller with gravity compensation

$$\boldsymbol{\tau} = \mathbf{J}^{\mathbf{T}}[\mathbf{K}_{\mathbf{m}}(\mathbf{x}_{\mathbf{d}} - \mathbf{x}) - \mathbf{D}_{\mathbf{m}}\dot{\mathbf{x}}] + \mathbf{G}. \tag{8}$$

C. Iterative Learning for Gravity Compensation

One straightforward way to reduce the effect of dynamic uncertainties (including incomplete or absent gravity compensation/cancellation) is to make the robot stiffer by tuning up the spring gains ($\mathbf{K}_{\mathbf{m}}$) in the impedance model. This could be feasible in simulations where the gains can be set to be very large, but not feasible in practice where the robot may have chattering due to large gains. Especially when heavy external payloads are involved, solely tuning the impedance gains may not be able to obtain a satisfactory result. To solve this problem, we introduce an iterative learning scheme for gravity compensation in Cartesian space.

Inspired by [13] where iterative learning was integrated with a PD controller in joint space, a Cartesian-space impedance control law (at the ith iteration, $i = 1, 2, \ldots$) integrating with a Git scheme for gravity compensation is proposed, which can be expressed by

$$\begin{aligned} \boldsymbol{\tau}_{\mathbf{i}} = {} & \mathbf{M}\mathbf{J}^{-\mathbf{1}}(\ddot{\mathbf{x}}_{\mathbf{d}} - \dot{\mathbf{J}}\mathbf{J}^{-\mathbf{1}}\dot{\mathbf{x}}_{\mathbf{d}}) + \mathbf{S}\mathbf{J}^{-\mathbf{1}}\dot{\mathbf{x}}_{\mathbf{d}} \\ & + \mathbf{J}^{\mathbf{T}}[\mathbf{D}_{\mathbf{m}}(\dot{\mathbf{x}}_{\mathbf{d}} - \dot{\mathbf{x}}) + \gamma\mathbf{K}_{\mathbf{m}}(\mathbf{x}_{\mathbf{d}} - \mathbf{x})] + \mathbf{J}^{\mathbf{T}}\mathbf{u}_{\mathbf{i-1}} \end{aligned} \tag{9}$$

where $\mathbf{J}^{\mathbf{T}}\mathbf{u}_{\mathbf{i-1}}$ is an iterative learning term for gravity compensation instead of a gravity term $\mathbf{G}$. For set-point regulation, it will be reduced to be

$$\boldsymbol{\tau}_{\mathbf{i}} = \mathbf{J}^{\mathbf{T}}[\gamma\mathbf{K}_{\mathbf{m}}(\mathbf{x}_{\mathbf{d}} - \mathbf{x}) - \mathbf{D}_{\mathbf{m}}\dot{\mathbf{x}}] + \mathbf{J}^{\mathbf{T}}\mathbf{u}_{\mathbf{i-1}}. \tag{10}$$

The update law for the iterative learning $\mathbf{u}_{\mathbf{i}}$ can be given by

$$\mathbf{u}_{\mathbf{i}} = \gamma\mathbf{K}_{\mathbf{m}}(\mathbf{x}_{\mathbf{d}} - \mathbf{x}) + \mathbf{u}_{\mathbf{i-1}} \tag{11}$$

where $\boldsymbol{\gamma}$ is a positive scalar gain, and setting $\mathbf{u}_{\mathbf{0}} = \mathbf{0}$ for initialization. Also, different from [13] where one iteration was set as 3 s while in this article it updates itself in each sampling loop. This ensures the updated values of the iteration term are changing continuously and smoothly from one iteration to the next, and also extends its validity to more general tracking tasks from set-point regulation. Theoretical analysis will be introduced in detail later.

To avoid a sudden impulse at the moment of the robot starts up due to a potentially large initial error between the initial actual position and the initial desired position, a simple linear interpolating strategy is used, which is given by

$$\begin{cases} \mathbf{x}_{\mathbf{d}} = \mathbf{x}_{\mathbf{0}} + (\mathbf{x}_{\mathbf{d}} - \mathbf{x}_{\mathbf{0}})\frac{\mathbf{t}}{\mathbf{t}_{\mathbf{1}}} & \text{if } \mathrm{t} \le \mathrm{t}_1 \\ \dot{\mathbf{x}}_{\mathbf{d}} = \dot{\mathbf{x}}_{\mathbf{0}} + (\dot{\mathbf{x}}_{\mathbf{d}} - \dot{\mathbf{x}}_{\mathbf{0}})\frac{\mathbf{t}}{\mathbf{t}_{\mathbf{1}}} & \text{if } \mathrm{t} \le \mathrm{t}_1 \\ \ddot{\mathbf{x}}_{\mathbf{d}} = \ddot{\mathbf{x}}_{\mathbf{0}} + (\ddot{\mathbf{x}}_{\mathbf{d}} - \ddot{\mathbf{x}}_{\mathbf{0}})\frac{\mathbf{t}}{\mathbf{t}_{\mathbf{1}}} & \text{if } \mathrm{t} \le \mathrm{t}_1 \\ \mathbf{x}_{\mathbf{d}} = \mathbf{x}_{\mathbf{d}} & \text{if } \mathrm{t} > \mathrm{t}_1 \\ \dot{\mathbf{x}}_{\mathbf{d}} = \dot{\mathbf{x}}_{\mathbf{d}} & \text{if } \mathrm{t} > \mathrm{t}_1 \\ \ddot{\mathbf{x}}_{\mathbf{d}} = \ddot{\mathbf{x}}_{\mathbf{d}} & \text{if } \mathrm{t} > \mathrm{t}_1 \end{cases} \tag{12}$$

where $\mathbf{x}_{\mathbf{0}} = \mathbf{constant}, \dot{\mathbf{x}}_{\mathbf{0}} = \mathbf{0}, \ddot{\mathbf{x}}_{\mathbf{0}} = \mathbf{0}$ are the initial actual position, velocity, and acceleration, t_1 is the duration of the transition period defined by the user (in this article, $t_1 = 2$ s). Note that the $\mathbf{x}_{\mathbf{d}}, \dot{\mathbf{x}}_{\mathbf{d}}, \ddot{\mathbf{x}}_{\mathbf{d}}$ on the right-hand side of the equations represent the theoretical values from the predefined trajectory or setpoint, while those on the left-hand side represent the values used for calculation in the controller. As shown in (12), the desired position, velocity, and acceleration are set up increasingly

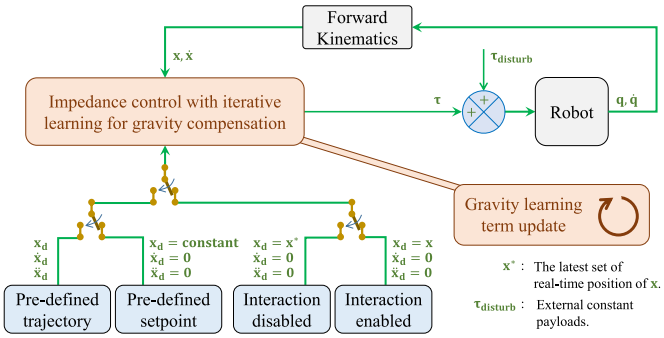

Fig. 1. Control block diagram of an impedance controller with iterative learning scheme for gravity compensation. When $\mathbf{x_d} = \mathbf{x}$, the position-dependent terms in the impedance controller and the Git update law vanish, meaning that the set-point regulation is released and interaction is enabled, and now the user can move the robot EE around. When $\mathbf{x_d} = \mathbf{x}^*$, a set-point regulation task is recovered and interaction is disabled. The latest set of position ($\mathbf{x}^*$) ensures seamless switching between the "interaction enabled" mode and the "interaction disabled" mode, which can be easily realized by a pedal switch.

from the initial actual ones $(\mathbf{x_0}, \dot{\mathbf{x}}_\mathbf{0}, \ddot{\mathbf{x}}_\mathbf{0})$ (at $t = 0$) to the desired ones $(\mathbf{x_d}, \dot{\mathbf{x}}_\mathbf{d}, \ddot{\mathbf{x}}_\mathbf{d})$ (at $t = t_1$) within the very first t_1 seconds. In other words, (12) ensures errors increase linearly from zeros (at $t = 0$) to the actual errors (at $t = t_1$) when the robot starts up. It should be noted that this smoothing strategy is independent of the control laws and only valid within the first t_1 seconds. The block diagram of the proposed iterative learning for gravity compensation in impedance control is illustrated in Fig. 1.

D. Analysis

In this section, the process of designing the iterative learning term and the corresponding update law will be presented in detail. Then, the convergence properties of the iterative learning scheme will be theoretically analyzed in a scenario of set-point regulation.

At the steady state $(\mathbf{q} = \mathbf{constant}, \dot{\mathbf{q}} = \ddot{\mathbf{q}} = \mathbf{0})$ of the ith iteration in the scenario of set-point regulation, it has

$$\begin{cases} \cancel{\mathbf{M}\ddot{\mathbf{q}}} + \cancel{\mathbf{S}\dot{\mathbf{q}}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}_\mathbf{i} \\ \boldsymbol{\tau}_\mathbf{i} = \mathbf{J}^\mathbf{T}[\gamma \mathbf{K_m}(\mathbf{x_d} - \mathbf{x}) - \cancel{\mathbf{D_m}\dot{\mathbf{x}}}] + \mathbf{J}^\mathbf{T}\mathbf{u_{i-1}}. \end{cases} \tag{13}$$

Combining the two equations in (13) as one equation, yields

$$\mathbf{G}(\mathbf{q_i}) = \mathbf{J}^\mathbf{T}\gamma \mathbf{K_m}(\mathbf{x_d} - \mathbf{x}) + \mathbf{J}^\mathbf{T}\mathbf{u_{i-1}}. \tag{14}$$

Based on (14), the update law of the iterative learning term can be designed as

$$\mathbf{J}^\mathbf{T}\mathbf{u_i} = \mathbf{J}^\mathbf{T}\gamma \mathbf{K_m}(\mathbf{x_d} - \mathbf{x}) + \mathbf{J}^\mathbf{T}\mathbf{u_{i-1}}. \tag{15}$$

Simplify (15), we obtain the update law given by (11). By designing the update law in this way and by comparing (14) with (15), we are actually assuming that at the steady state, the learning term converged to the gravity term, i.e.,

$$\mathbf{G}(\mathbf{q_i}) = \mathbf{J}^\mathbf{T}\mathbf{u_i}. \tag{16}$$

The subsequent part will analyze and show proof of the convergence capability of the designed iterative learning scheme.

The update law (11) can be rewritten as

$$\mathbf{u_i} - \mathbf{u_{i-1}} = \gamma \mathbf{K_m}(\mathbf{x_d} - \mathbf{x_i}). \tag{17}$$

Define the position error in Cartesian space as $\mathbf{e_i} = \mathbf{x_d} - \mathbf{x_i}$, (17) can be rewritten as

$$\mathbf{u_i} - \mathbf{u_{i-1}} = \gamma \mathbf{K_m}\mathbf{e_i}. \tag{18}$$

Also, the position error in Cartesian space between two adjacent iteration steps can be expressed as

$$\mathbf{x_i} - \mathbf{x_{i-1}} = \mathbf{x_i} - \mathbf{x_d} + \mathbf{x_d} - \mathbf{x_{i-1}} = -\mathbf{e_i} + \mathbf{e_{i-1}}. \tag{19}$$

Knowing that the derivative of gravity is bounded [13] by

$$\left\| \frac{\partial \mathbf{G}(\mathbf{q})}{\partial \mathbf{q}} \right\| \leq \boldsymbol{\alpha} \tag{20}$$

where $\boldsymbol{\alpha}$ is a positive constant. Rewrite (20) in the form of finite difference as

$$\left\| \frac{\mathbf{G}(\mathbf{q_i}) - \mathbf{G}(\mathbf{q_{i-1}})}{\mathbf{q_i} - \mathbf{q_{i-1}}} \right\| \leq \boldsymbol{\alpha}. \tag{21}$$

Also, the relationship between Cartesian velocity and joint velocity is given by

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}. \tag{22}$$

Assuming that the Jacobian matrix is invertible, i.e., $\mathbf{J^{-1}}$ exists. Rewrite (22) in the form of finite difference as

$$\mathbf{q_i} - \mathbf{q_{i-1}} = \mathbf{J^{-1}}(\mathbf{x_i} - \mathbf{x_{i-1}}). \tag{23}$$

From the relationship between gravity term and iterative learning term at steady state (16), it yields

$$\begin{aligned} \|\mathbf{u_i} - \mathbf{u_{i-1}}\| &= \|\mathbf{J^{-T}}\mathbf{G}(\mathbf{q_i}) - \mathbf{J^{-T}}\mathbf{G}(\mathbf{q_{i-1}})\| \\ &\leq \|\mathbf{J^{-T}}\|\|\mathbf{G}(\mathbf{q_i}) - \mathbf{G}(\mathbf{q_{i-1}})\| \\ &\leq \alpha\|\mathbf{J^{-T}}\|\|\mathbf{q_i} - \mathbf{q_{i-1}}\| \quad \text{[by (21)]} \\ &\leq \alpha\|\mathbf{J^{-T}}\|\|\mathbf{J^{-1}}(\mathbf{x_i} - \mathbf{x_{i-1}})\| \quad \text{[by (23)]} \\ &\leq \alpha\|\mathbf{J^{-T}}\|\|\mathbf{J^{-1}}(-\mathbf{e_i} + \mathbf{e_{i-1}})\| \quad \text{[by (19)]} \\ &\leq \alpha\|\mathbf{J^{-T}}\|\|\mathbf{J^{-1}}\|(\|\mathbf{e_i}\| + \|\mathbf{e_{i-1}}\|). \end{aligned} \tag{24}$$

Assuming that the minimum eigenvalue of the user-defined matrix $\mathbf{K_m}$ meets the condition of $\lambda_{\mathbf{min}}(\mathbf{K_m}) > \alpha$, then it can yield the following inequality property:

$$\gamma\alpha\|\mathbf{e_i}\| < \gamma\lambda_{\mathbf{min}}(\mathbf{K_m})\|\mathbf{e_i}\| \leq \|\gamma \mathbf{K_m}\mathbf{e_i}\|. \tag{25}$$

By combining (18), (24), (25), yields

$$\gamma\alpha\|\mathbf{e_i}\| < \|\gamma \mathbf{K_m}\mathbf{e_i}\| \leq \alpha\|\mathbf{J^{-T}}\|\|\mathbf{J^{-1}}\|(\|\mathbf{e_i}\| + \|\mathbf{e_{i-1}}\|). \tag{26}$$

By simplifying (26), yields

$$\gamma\|\mathbf{e_i}\| < \|\mathbf{J^{-T}}\|\|\mathbf{J^{-1}}\|(\|\mathbf{e_i}\| + \|\mathbf{e_{i-1}}\|). \tag{27}$$

Reorganizing (27), yields

$$\|\mathbf{e_i}\| < \frac{\beta}{\gamma - \beta}\|\mathbf{e_{i-1}}\| \tag{28}$$

where $\beta = \|\mathbf{J^{-T}}\|\|\mathbf{J^{-1}}\|$. In order for contraction mapping, requires

$$\frac{\beta}{\gamma - \beta} \leq 1. \tag{29}$$

Due to $\beta > 0$ is always true, yields

$$\boldsymbol{\gamma} \geq 2\boldsymbol{\beta}$$
$$\boldsymbol{\gamma} \geq 2\|\mathbf{J^{-T}}\|\|\mathbf{J^{-1}}\| \geq 2\|\mathbf{J^{-T}J^{-1}}\|. \tag{30}$$

Assuming that the Jacobian matrix $\mathbf{J}$ is bounded, then $\mathbf{J^{-T}}$ and $\mathbf{J^{-1}}$ are both bounded. Then, set the following boundness:

$$b \geq \|\mathbf{J^{-T}J^{-1}}\|. \tag{31}$$

Finally, it can conclude that, on the conditions of (1) Jacobian matrix is invertible and bounded, and (2) $\lambda_{\mathbf{min}}(\mathbf{K_m}) > \alpha$, then, $\boldsymbol{\gamma} \geq 2b$ can ensure the iterative learning term [$\mathbf{J^T u_{i-1}}$ in (10)] being a contraction mapping, in other words, can ensure the iterative learning term converges to the true gravity at the steady state. Note that the convergence condition here is only sufficient, which means that even if it is violated the iterative learning term may still converge. This is consistent with the conclusion made in [13]. Note that this convergence analysis result still holds true when model uncertainties exist (see Supplementary Material).[1]

E. Steady-State Scaling Strategy

For the iterative learning-based update law (11), to explicitly display an learning rate $\boldsymbol{\eta}$ (by default $\eta = 1$), the update law (11) can be rewritten as

$$\mathbf{u_i} = \gamma\eta\mathbf{K_m}(\mathbf{x_d} - \mathbf{x}) + \mathbf{u_{i-1}}. \tag{32}$$

It is worth noting that an important assumption has been made for the contraction mapping is "at steady state," and the update law (32) should be executed at steady-state theoretically. This is consistent with the drawback described in the prior work [12], [13], [14], where the steady state is set as 3 s in their simulations. Executing the update law only at steady state (e.g., every 3 s in [13]) is ok for simple simulations, but in practice, it would be a significant limitation.

As a further step in this article, analog to the concept of finite difference, we scale the common "steady-state" period (taking 1 s here as an example) down to the level of sampling time (0.001 s) such that the update law can be executed in each sampling loop. Since the default learning rate ($\eta = 1$) in (32) is corresponding to the common steady-state period (1 s), it also needs to be scaled down to be $\eta = 0.001$ in order to match with the scaled steady-state period (0.001 s). This enables the easy implementation of the update law (32) and allows it to be updated in each sampling loop.

More importantly, the steady-state scaling strategy enables the iterative learning scheme to be valid also for more general trajectory-tracking scenarios. Since learning-based impedance control law (10) (for set-point regulation scenario) is reduced from (9) (for the general trajectory-tracking scenario), impedance control law (9), and iterative learning update law (32) can be used for iterative learning on the gravity compensation in trajectory-tracking scenarios. This strategy will be evaluated with simulations and experiments in the following section.

TABLE I
PARAMETERIZATION FOR SIMULATIONS AND EXPERIMENTS

Parameters	Simulations	Experiments
Spring	$\mathbf{K_m = 38.44I}$	$\mathbf{K_m = 7.29I}$
Damper	$\mathbf{D_m = 12.40I}$	$\mathbf{D_m = 5.40I}$
Spring (increased stiffness)	NA	$\mathbf{K_m = 200I}$
Damper (increased stiffness)	NA	$\mathbf{D_m = 2I}$
Learning gain γ	1	1
Learning rate η	1/0.001/0.005/0.025	0.001

Note: $\mathbf{I} \in \mathbb{R}^{3\times3}$ denote identity matrix. NA, not applied. The parameters are determined via trial and error with a binary search strategy.

III. Simulations, Experiments, and Results

A. Robotic System

A 3-DOF PHANToM Premium 1.5 A robot (3-D Systems, Inc., Cary, NC, USA) is used for simulations and experiments in this article. For the simulations, we reconstruct the kinematic model and dynamic model of the PHANToM robot based on [22] and conduct the simulations using MATLAB/Simulink (version R2020a, MathWorks Inc., Natick, MA, USA). For the experiments, the physical robot is controlled via joint torque command, which is sent from MATLAB/Simulink using Quarc real-time control software (Quanser Inc., Markham, ON, Canada). The control rate of the robot is 1000 Hz. The MATLAB/Simulink and Quarc software run on a computer with a 3.33 GHz Intel(R) Core(TM) 2 i5 CPU with a Windows 7 Enterprise 64-bit operating system.

B. Parameterization

For all simulations and experiments in the remaining part of this article, the parameter values used in the learning-based impedance controller (9) and the iterative learning update law (32) are listed in Table I. In order to involve acute changes in position and velocity, a concaved-square trajectory is selected for the simulations and experiments, which can be expressed as a function of time given by

$$\begin{cases} x_d = R\cos^3(t) \\ y_d = R\sin^3(t) + R \\ z_d = 0 \end{cases} \tag{33}$$

where $R = 0.02$ m is a parameter of the concaved-square. Note that the described trajectory is in a vertical plane in the workspace of the 3DOF robot.

In the following sections, a series of simulations and experiments are conducted to evaluate the effectiveness of the proposed iterative learning scheme both in free-motion mode (i.e., trajectory tracking tasks) and in restricted-motion mode (i.e., set-point regulation tasks). A demonstration video of the experiments can be found in the supplementary material.[2]

[1]Online file: https://drive.google.com/file/d/1rLEKJfsgCqtWBDyvlROXtAm1qiD5VwGa/view?usp=sharing

[2]Online video: https://drive.google.com/file/d/1-7NaY11clzW_-7ldo3nk8-qGMRQ2Lmir/view?usp=sharing

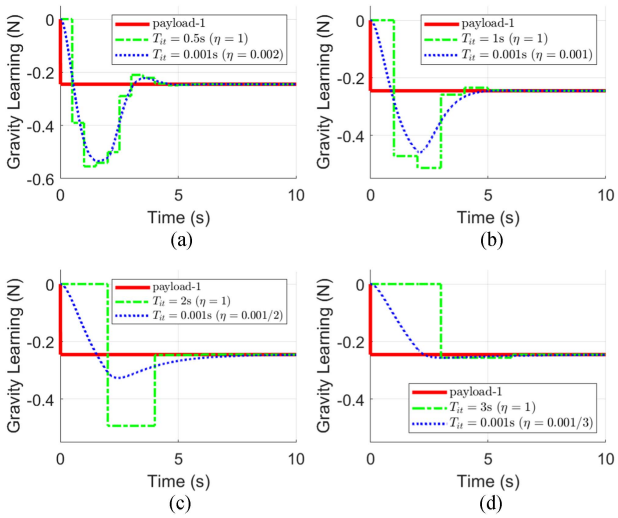

Fig. 2. Simulation results of scaling down the iterative update time (T_{it}, i.e., the steady-state time) while scaling down the iterative learning rate (η) accordingly. (a) Scaling down T_{it} from 0.5 s to 0.001 s. (b) Scaling down T_{it} from 1 s to 0.001 s. (c) Scaling down T_{it} from 2 s to 0.001 s. (d) Scaling down T_{it} from 3 s to 0.001 s. The reference is an external payload-1 (25 g). The setpoint is set as $[0.01, 0.04, 0]$ m in Cartesian space.

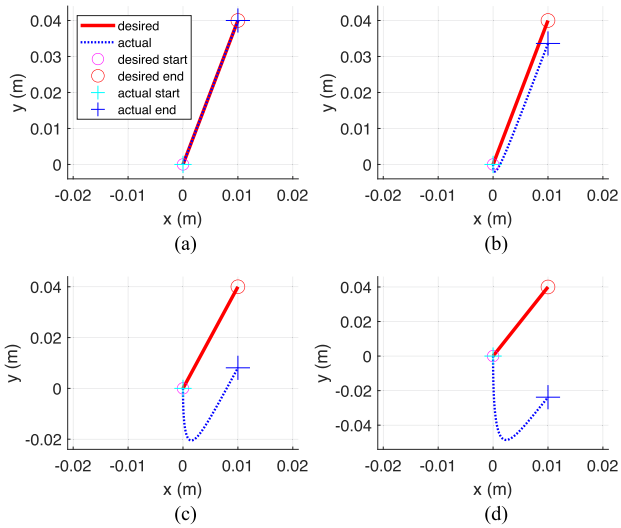

Fig. 3. Simulation results in a set-point regulation task with a pure impedance controller under an ideal dynamic model. (a) Case #0, no payloads. (b) Case #1, payload-1 (25 g). (c) Case #2, payload-2 (125 g). (d) Case #3, payload-3 (250 g). The setpoint is set as $[0.01, 0.04, 0]$ m in Cartesian space.

It should be noted that the gravity compensation estimated by the iterative learning scheme may include two main components, i.e., gravity term in the dynamic model, and (if applicable) all constant payloads attached to the robot EE or robot body. In order to clearly reveal the converging properties and learning performance of the learning scheme, in the simulations of this work, we will assume the gravity term is fully known in such a way the iterative learning term only learns and compensates for the unknown external payloads.

C. Simulation to Evaluate Steady-State Scaling Strategy

In the original work [13] where the iterative learning scheme was initially proposed with a PD controller in joint space, a significant drawback of the scheme is that the iterative update should be executed at steady state. This is also true for this work since the same "steady-state" assumption has been used during the theoretical analysis of the convergence properties. This drawback can largely limit the learning scheme to be implemented in practice.

To overcome this drawback, at the end of Section II, we proposed a strategy to scale down the steady-state period [equivalent to iterative update time for updating the update law (32)] to the same level as the sampling time in order to improve and generalize the iterative learning scheme. The steady-state scaling strategy requires the learning rate (η) to be scaled to the same level accordingly. In this section, we will evaluate this strategy with simulations.

In Fig. 2, a comparison of with-scaling and without-scaling the iterative update time is presented when the robot is in a set-point regulation task. For the without-scaling ($\eta = 1$) scenario, the gravity learning behavior under various conditions of iterative update time ($T_{it} = 0.5, 1, 2, 3$ s) is investigated as shown by the green dash-dot lines in the figure. Correspondingly, the blue dot lines represent the gravity learning behavior in the with-scaling scenario, where the iterative update time is scaled ($T_{it} = 0.001$ s) to be the same as the sampling time while the learning rate η is scaled accordingly ($\eta = 0.002, 0.001, 0.001/2, 0.001/3$ s). The comparison between the green dash-dot lines and the blue dot lines in Fig. 2 revealed that the steady-state scaling strategy is effective and reasonably sound. Especially in Fig. 2(a), the similarity between the with-scaling scenario and the without-scaling scenario is clearly revealed.

D. Simulation on Set-Point Regulation Task

Simulations in four cases are conducted in set-point regulation tasks (restricted-motion mode). Different cases are related to different external payloads attached to the robot EE, which can be described as follows.

1) Case #0, reference, no external payloads.
2) Case #1, payload-1 (25 g) attached.
3) Case #2, payload-2 (125 g) attached.
4) Case #3, payload-3 (250 g) attached.

A 3DOF robot is implemented with the iterative learning-based control law (10) and learning update law (32) as well as the steady-state scaling strategy, where the iterative learning scheme is used to iteratively learn and compensate the gravity of the external payloads in different cases. Fig. 3 shows the simulation results in a set-point regulation task with a pure impedance controller (8) [or equivalently controller (10) with setting the learning rate $\eta = 0$] under an ideal dynamic model (i.e., the dynamic model matrices $\mathbf{M}, \mathbf{S}, \mathbf{G}$ are fully known). As shown in Fig. 3(a) (Case #0), with the ideal dynamic model and without any external disturbances, the impedance controller can achieve very good set-point regulation performance. However,

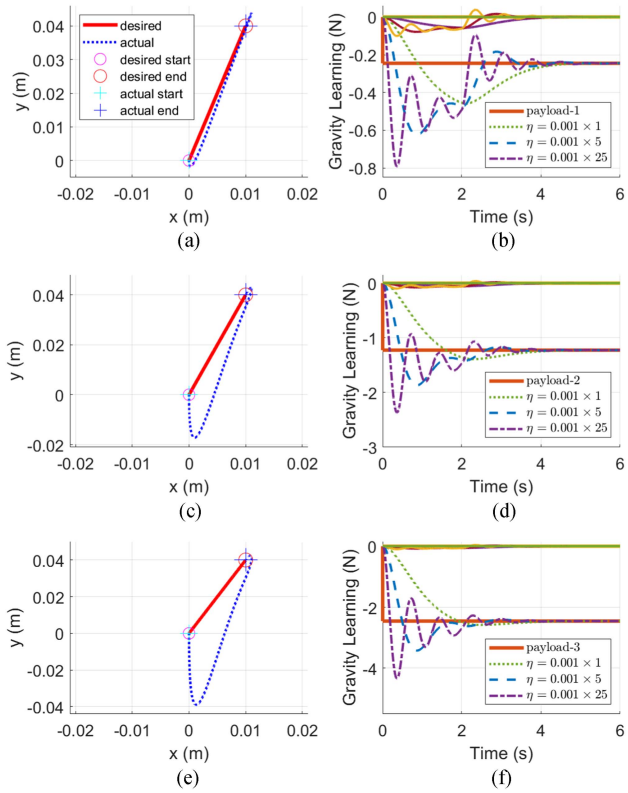

Fig. 4. Simulation results in a set-point regulation task under different iterative learning rates for learning gravity compensation. (a) Case #1 with payload-1 (25 g). (b) Gravity learning result in Case #1. (c) Case #2 with payload-2 (125 g). (d) Gravity learning result in Case #2. (e) Case #3 with payload-3 (250 g). (f) Gravity learning result in Case #3. Note that the solid lines converging to zero in (b), (d), and (f) are the learning results along the nongravity axes in Cartesian space and their legends are ignored for clarity purposes.

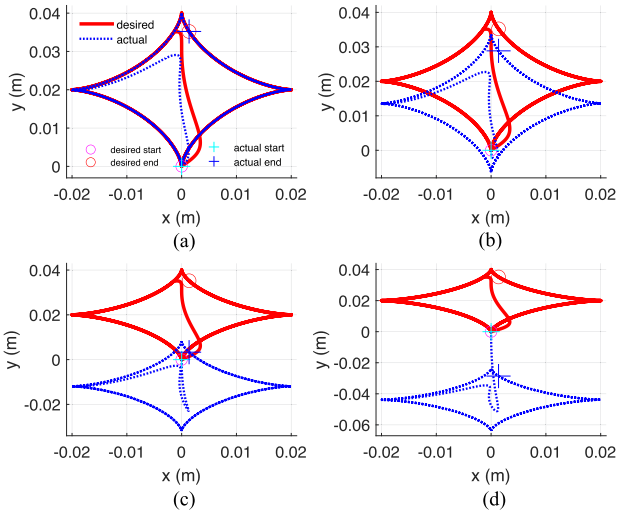

Fig. 5. Simulation results in a trajectory tracking task with a pure impedance controller under an ideal dynamic model. (a) Case #0, no payloads. (b) Case #1, payload-1 (25 g). (c) Case #2, payload-2 (125 g). (d) Case #3, payload-3 (250 g).

in Fig. 3(b) (Case #1), when an external payload-1 (25 g) is attached to the robot EE, the regulation result made by the same impedance controller shifted downward due to the incomplete gravity compensation. Furthermore, as the weight of the external payload increases, the shifts get worse as shown in Fig. 3(c) (Case #2) and in Fig. 3(d) (Case #3).

When the iterative learning-based controller (10) and the update law (32) are implemented, the external payloads can be accurately compensated via iterative learning thus accurate regulation performance is recovered. Fig. 4 shows the simulation results in a set-point regulation task under different iterative learning rates ($\eta = 0.001/0.005/0.025$). As shown in Fig. 4(a) and (b), when an external payload-1 (25 g) is attached to the robot EE (Case #1), the set-point regulation performance [see Fig. 4(a)] is recovered to be accurate with the help of iterative learning on gravity compensation, while different learning rates ($\eta = 0.001/0.005/0.025$) may result in different converging behavior [see Fig. 4(b)]. Specifically, a large learning rate (η) may result in an oscillate converging behavior while a smaller learning rate (η) may result in smooth converging behavior. Similar simulation results can be found when the weight of the payload increases [Fig. 4(c) and (d) for payload-2 (125 g), and Fig. 4(e) and (f) for payload-3 (250 g)]. The getting worse oscillation behavior as the learning rate increases also indicates that the learning rate should be matching the iterative update time, which again verified the reasonability of the proposed steady-state scaling strategy.

The simulation results in the regulation task demonstrate that the incomplete gravity compensation will cause the actual regulated position to shift downward, thus, the task performance is destroyed. However, with the steady-state scaling strategy and implementing the iterative learning scheme to learn for gravity compensation, the regulation accuracy can be recovered.

E. Simulation on Trajectory Tracking Task

By using the steady-state scaling strategy, the steady-state period can be scaled to be on the same level as the robot sampling time. By doing this, the iterative learning scheme can be extended to trajectory-tracking tasks theoretically. This will be evaluated by simulations in this section.

Similar to the procedures used in the set-point regulation task presented in the previous section, we repeat all the procedures in the trajectory tracking task. The trajectory of concaved-square (33) is employed for the trajectory tracking task. The same 3DOF robot model is employed and the iterative learning-based impedance controller (9) with the update law (32) is implemented.

The simulation results in the trajectory tracking task are similar to that in the set-point regulation task. Fig. 5 shows the simulation results in a trajectory tracking task with a pure impedance controller (7) (or equivalently controller (9) with setting the learning rate $\eta = 0$) under an ideal dynamic model where the $\mathbf{M}, \mathbf{S}, \mathbf{G}$ are fully known. As shown in Fig. 5(a) (Case #0), with the ideal dynamic model and without any external payloads attached, the impedance controller can achieve accurate trajectory tracking performance. However, in Fig. 5(b) (Case #1), when an external payload-1 (25 g) is attached to the robot EE, the actual trajectory made by the same impedance controller shifted downward. Again, as the weight of the external

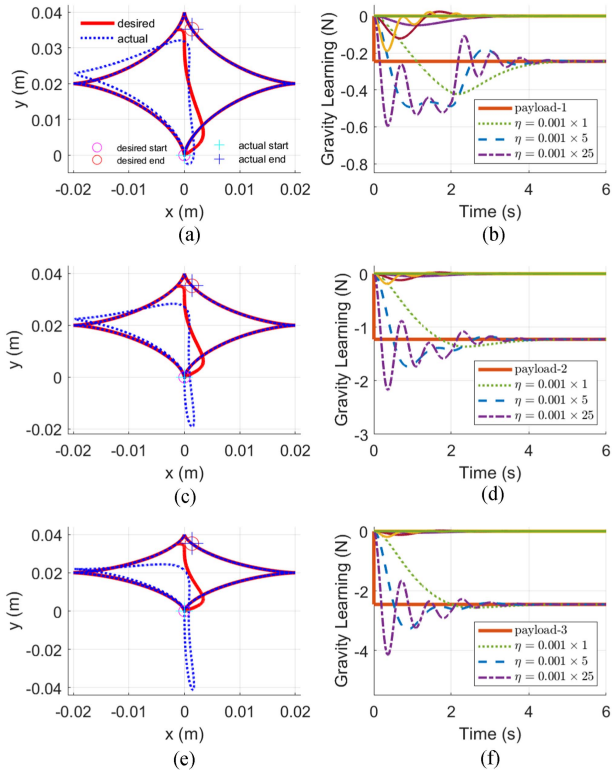

Fig. 6. Simulation results in a trajectory tracking task under different iterative learning rates for learning gravity compensation. (a) Case #1 with payload-1 (25 g). (b) Gravity learning result in Case #1. (c) Case #2 with payload-2 (125 g). (d) Gravity learning result in Case #2. (e) Case #3 with payload-3 (250 g). (f) Gravity learning result in Case #3. Note that the solid lines converging to zero in (b), (d), and (f) are the learning results along the nongravity axes in Cartesian space and their legends are ignored for clarity purposes.

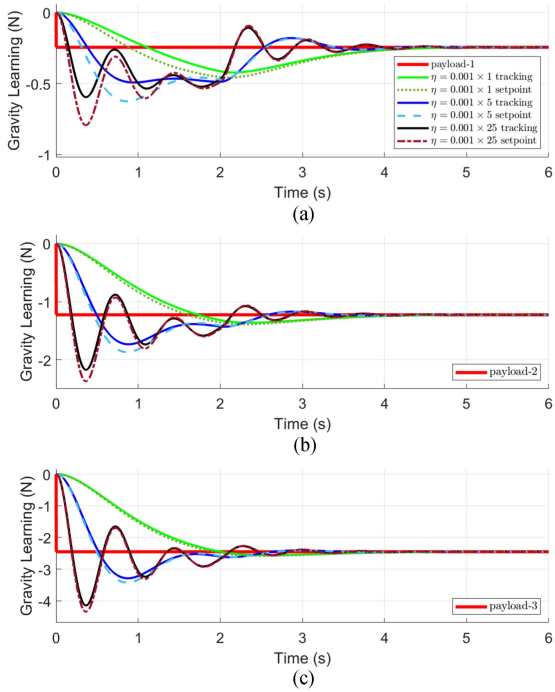

Fig. 7. Simulation results of comparing the gravity learning behavior in a trajectory tracking task and in a set-point regulation task under different iterative learning rates. (a) Gravity learning results in Case #1 with payload-1 (25 g). (b) Gravity learning results in Case #2 with payload-2 (125 g). (c) Gravity learning results in Case #3 with payload-3 (250 g).

payload increases, the shifted displacements get larger as shown in Fig. 5(c) (Case #2) and Fig. 5(d) (Case #3).

When the learning-based impedance controller (9) is implemented with the steady-state scaling strategy, the external payloads can be compensated via iterative learning thus accurate tracking performance can be recovered. Fig. 6 shows the simulated tracking performance under different iterative learning rates ($\eta = 0.001/0.005/0.025$). As shown in Fig. 6(a) and (b), when an external payload-1 (25 g) is attached to the robot EE (Case #1), the trajectory tracking performance [see Fig. 6(a)] is recovered to be accurate with the help of iterative learning on gravity compensation, while different learning rates may have different converging behaviors [see Fig. 6(b)], which is affected by the learning rate η. Similar to the observed phenomenons in the set-point regulation task, a larger learning rate may have an oscillate converging behavior while a smaller learning rate may have a slow but smooth converging behavior. Similar simulation results can be found when the weight of the payload increases [Fig. 6(c) and (d) for payload-2 (125 g), and Fig. 6(e) and (f) for payload-3 (250 g)].

The simulation results in the trajectory tracking task demonstrate that, by using the steady-state scaling strategy, the iterative learning scheme for gravity compensation is also valid when a robot is in a free-motion mode. With an appropriate setting on the learning rate, the iterative learning term is able to converge to the actual weight of the external payload.

If we take a comparison on the gravity learning behavior in the trajectory tracking task [see Fig. 6(b), (d), and (f)] with that in the set-point regulation task [see Fig. 4(b), (d), and (f)], and put them in a same figure as shown in Fig. 7, we can clearly found that the gravity learning behavior is very similar and has almost the same converging process. Especially in Fig. 7(c), the gravity learning behavior is almost the same in the two scenarios. The results in Fig. 7 indicate that by using the steady-state scaling strategy, the iterative learning scheme for gravity compensation can be used for both set-point regulation tasks and trajectory-tracking tasks, while their converging process are almost the same. This verified the feasibility of extending the iterative learning scheme to scenarios of robots in free motions.

F. Experiment on Trajectory Tracking Task

In contrast to simulations, a series of experiments are conducted to evaluate the presented iterative learning-based controller by using a 3DOF Phantom Premium 1.5 A robot. The trajectory of concaved-square (33) is employed.

The experimental results of trajectory tracking performance in different conditions are shown in Fig. 8. Fig. 8(a) shows the tracking performance when only an impedance controller (7) is implemented with relatively small impedance gains. Note that inherent uncertainties of the physical robot system, including but not limited to dynamic model error and unmodeled friction,

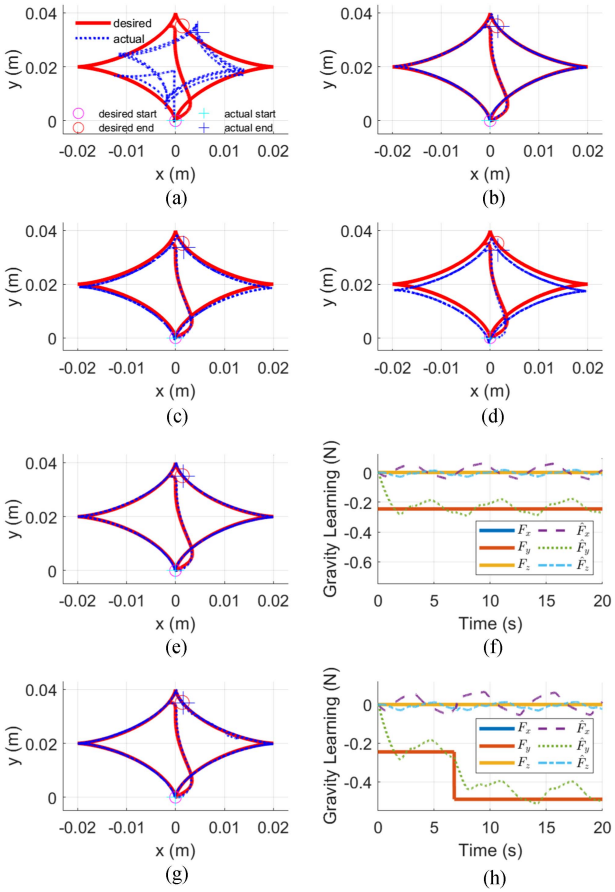

Fig. 8. Experimental results of a trajectory tracking task in different scenarios. (a) Impedance controller only. (b) Impedance controller only, but with increased robot stiffness. (c) Increased robot stiffness with payload-1 (25 g). (d) Increased robot stiffness with two payload-1 (50 g in total, both attached since the beginning). (e) Increased robot stiffness with payload-1 and iterative learning. (f) Iterative learning result with payload-1. (g) Increased robot stiffness with two separate payload-1 (25g+25 g) and iterative learning. (h) Iterative learning result with two separate payload-1. Note, for (g) and (h) where two payload-1 appear, the first payload-1 is attached since the beginning while the second payload-1 is attached at around the 6th second.

always exist in all physical experiments. As shown in Fig. 8(a), the tracking performance is significantly affected due to the inherent uncertainties.

A straightforward way to overcome the inherent uncertainties is to make the robot stiffer by tuning the impedance gains. When increasing the robot stiffness (see Table I for increased stiffness by tuning impedance gains), the inherent uncertainties can be overcome thus accurate tracking can be obtained. However, solely using increased stiffness to overcome the uncertainties is limited when external payloads are attached to the robot EE, especially for heavy external payloads. As shown in Fig. 8(c), when payload-1 (25 g) is attached, although most of its mass can be overcome by the increased stiffness, there still have significant shifting-down for the actual trajectory. Especially when two payload-1 (50 g in total, all attached since the beginning) are attached, the shift gets larger as shown in Fig. 8(d). The effect of increasing the stiffness could be very good in simulations in order to achieve accurate task performance, but it is limited in

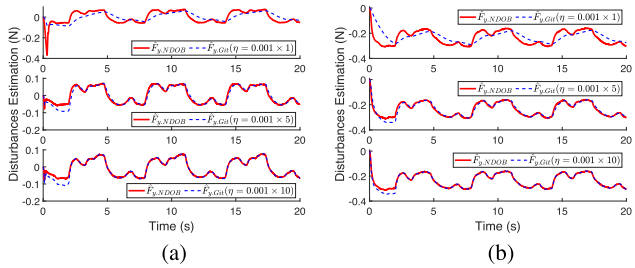

Fig. 9. Experimental results of disturbance estimation by NDOB and Git in trajectory tracking tasks. Note that only estimation along the y-axis is displayed for clarity since gravity is along the y-axis in this work. (a) Without external payloads. (b) With external payload-1 (25 g).

practice since too large stiffness can cause robot chattering and thus unstable. Therefore, in practice, smaller impedance gains are used at the cost of task performance accuracy. From Fig. 8(c) and (d), we can clearly observe that the inaccuracy part is a shift downward away from the desired trajectory which is mainly caused by incomplete gravity compensation.

To reduce the effect brought by incomplete gravity compensation, the iterative learning scheme is employed. By implementing the learning-based impedance controller (9), the effect of external payload-1 (25 g) can be effectively compensated [see Fig. 8(e) and (f)]. Furthermore, a second payload-1 (25 g) can also be effectively compensated [see Fig. 8(g) and (h)]. One can notice that in Fig. 8(f) and (h), the learning term has bounded errors and cannot converge to the exact weight of the payloads. By comparing with the simulation results on trajectory tracking tasks, we can reasonably conclude that the bounded learning errors are caused by inherent uncertainties in the physical robotic system (e.g., inaccurate dynamic model, joint friction, etc.). In other words, the iterative learning term estimates gravity plus a part of the other uncertainties. This is verified with a further experiment where NDOB is employed to estimate the lumped uncertainties, and the results are shown in Fig. 9. In the figure, we can see that with an appropriately high learning rate ($\eta = 0.001 \times 10$), the Git algorithm can accurately estimate the lumped uncertainties as the same as the NDOB does. While with a low learning rate ($\eta = 0.001 \times 1$), the iterative learning algorithm can still accurately estimate the gravity part [see Fig. 9(b)], but only a rough estimation for the other uncertainties [see Fig. 9(a)]. Note that NDOB is a specific type of observer among a variety of disturbance observers, and it is selected here as a reference due to its high accuracy in estimating the lumped uncertainties and its ability to estimate the nonlinearities in the dynamics [7].

G. Experiment on Set-Point Regulation Task

By implementing the iterative learning-based controller (10), experiments on set-point regulation involving pHRI are conducted in two scenarios, i.e., pHRI disabled, and pHRI enabled. Fig. 10 shows the set-point regulation performance under the iterative learning-based controller when pHRI is involved. As shown in Fig. 10(a) and (b), when the pHRI is disabled the robot will reject human-applied force and keep the robot EE

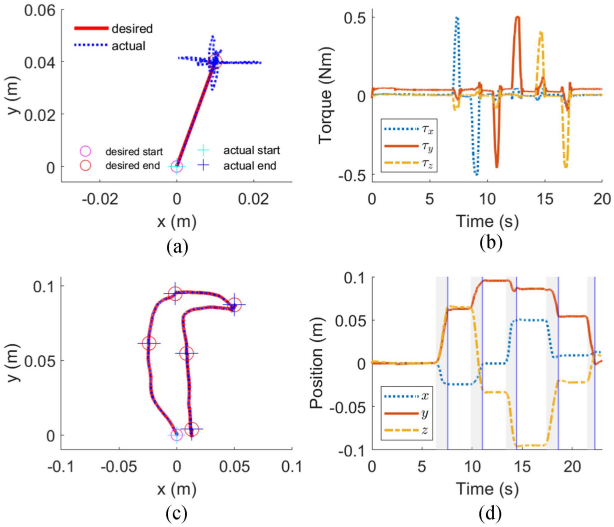

Fig. 10. Experimental results in a set-point regulation task involving pHRI. (a) Trajectory when pHRI disabled. (b) Computed torque when pHRI disabled. (c) Trajectory when pHRI enabled. (d) Actual position when pHRI enabled. Note, the five shaded areas in (d) indicate five times of interaction during which the user moves the robot EE from one point to another as shown in (c). The five vertical blue lines in (d) are the time points corresponding to the five actual endpoints in (c).

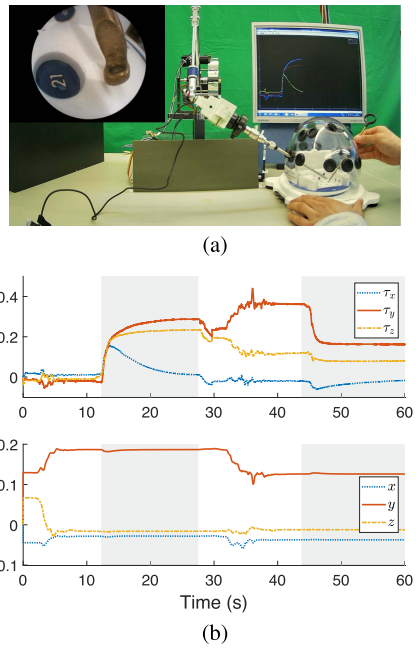

Fig. 11. Robot-assisted arthroscopy with a FAST simulator. Gray areas represent two periods of holding the arthroscope still by the robot with different scope views. (a) Setup scenario. (b) Torque and position.

remain at a fixed position. This realizes one expected condition in our application, i.e., the robot holds with an arthroscope and keeps it still while rejecting all potential disturbances. When pHRI is enabled [see Fig. 10(c) and (d)], the robot EE can be freely moved by the human user to wherever the user wanted. This realizes another expected condition in our application, i.e., the robot allows the surgeon to freely move it to a new position for adjusting the arthroscope view when necessary.

A further evaluation is to implement the controller in an application scenario mimicking robot-assisted arthroscopic surgery with a fundamentals of arthroscopic surgery training (FAST) simulator as shown in Fig. 11(a). In the application scenario, the robot EE is expected to hold with an arthroscope still while rejecting all potential disturbances. And when necessary, the arthroscope can be freely moved to a new position for adjusting the scope view. The experimental results of this application scenario are shown in Fig. 11(b). The two shaded gray areas in Fig. 11(b) represent two periods of holding the arthroscope still by the robot with different scope views. And during these two periods, we can see that the robot EE position is accurately kept constant, which verified the effectiveness of the implemented iterative learning-based controller.

H. Comparing With Other Methods

The presented Git scheme in this work extended the validity of the prior work [13] into both trajectory tracking tasks and set-point regulation tasks. The simulation and experimental results demonstrate the good performance of the Git scheme in learning and compensating for gravity. The disturbance observers like NDOB can accurately estimate the lumped uncertainties including gravity, but it is not suitable for pHRI scenarios since it will prevent human–robot interaction [6], [7]. An adaptive controller can also compensate for dynamic uncertainties including gravity [17], [18], but it is a controller rather than an independent strategy of disturbance estimation. Moreover, as a controller, it cannot provide compliant robot behavior like an impedance controller can do. The main differences among these methods are summarized in Table II.

TABLE II
COMPARING WITH METHODS IN LITERATURE

Methods	Uncertainties	Applicability	Requirements
Git	Gravity	①, ②	$\mathbf{x}, \mathbf{x_d}, \mathbf{J^T}$
Git [13]	Gravity	①	$\mathbf{q}, \mathbf{q_d}$
NDOB [7]	All	①, ②	$\dot{\mathbf{q}}, \mathbf{M^{-1}}, \mathbf{S}, \mathbf{G}$
Adapt [17], [18]	Dynamics	①, ②	$\mathbf{q}, \dot{\mathbf{q}}, \mathbf{q_d}, \dot{\mathbf{q}}_\mathbf{d}, \ddot{\mathbf{q}}_\mathbf{d}, \mathbf{Y}$

Note: ① Setpoint regulation tasks; ② Trajectory tracking tasks; subscript $_d$ means "desired"; $\mathbf{Y}$ is the regressor matrix in a linearized dynamic model which is derived via sophisticated process based on the general dynamic model, which means that $\mathbf{M}, \mathbf{S}, \mathbf{G}$ are implicitly required in order to obtain $\mathbf{Y}$.

In future work, we will further investigate the similarities and differences in the performance of simulations and experiments by comparing with the methods in the literature, e.g., NDOB, adaptive controller, and conventional PID controller. Also, we will use the full impedance model without simplifications, which will enable the inertia term to be tunable.

IV. CONCLUSION

In this article, we presented a simple and compact Git scheme for gravity compensation in Cartesian space. The whole process of developing the Git scheme is presented in detail, including motivation, theoretical analysis, simulations, experiments, and application. First, the convergence properties are theoretically analyzed. Then, a steady-state scaling strategy is proposed to

improve the Git scheme, which also extends its validity to more general trajectory tracking scenarios. By integrating the Git scheme with an impedance controller, an iterative learning-based impedance controller is constructed, where the Git algorithm can accurately learn for gravity compensation while the impedance controller can provide a robot with compliant behavior, thus ensuring a safe human–robot interaction in *p*HRI scenario. The learning accuracy of the Git scheme together with the scaling strategy are verified by simulations on both set-point regulation tasks and trajectory tracking tasks. The effectiveness of the learning-based controller is further validated by physical experiments on both trajectory tracking tasks and set-point regulation tasks. An application experiment in a simplified scenario of robot-assisted arthroscopic surgery also evaluated the effectiveness of the implemented learning-based controller. The results demonstrated that the integrated controller can achieve good tracking performance and regulation accuracy when heavy external payloads are attached to the robot EE. Moreover, it allows seamless switching between set-point regulation and human–robot interaction.

The major benefits of the presented Git scheme for gravity compensation can include the following:

1) simple and compact formulation and no need for the robot dynamics;
2) no need for any information about external payloads;
3) no need for higher impedance gains for reducing the effects of incomplete gravity compensation;
4) it is valid for both set-point regulation tasks and trajectory tracking tasks.

REFERENCES

[1] M. G. Fujie and B. Zhang, "State-of-the-art of intelligent minimally invasive surgical robots," *Front. Med.*, vol. 14, no. 4, pp. 404–416, 2020.

[2] R. H. Taylor, A. Menciassi, G. Fichtinger, P. Fiorini, and P. Dario, "Medical robotics and computer-integrated surgery," in *Springer Handbook of Robotics*. Cham, Switzerland: Springer, 2016, pp. 1657–1684.

[3] D. J. Jacofsky and M. Allen, "Robotics in arthroplasty: A comprehensive review," *J. Arthroplasty*, vol. 31, no. 10, pp. 2353–2363, 2016.

[4] J. M. McDonnell et al., "Surgeon proficiency in robot-assisted spine surgery: A narrative review," *Bone Joint J.*, vol. 102, no. 5, pp. 568–572, 2020.

[5] K. Bennett and S. Kamineni, "History of elbow arthroscopy," *J. Arthroscopic Surg. Sports Med.*, vol. 1, no. 1, pp. 23–31, 2020.

[6] T. Li, A. Badre, H. D. Taghirad, and M. Tavakoli, "Integrating impedance control and nonlinear disturbance observer for robot-assisted arthroscope control in elbow arthroscopic surgery," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 11172–11179.

[7] T. Li et al., "A brief survey of observers for disturbance estimation and compensation," *Robotica*, vol. 41, no. 12, pp. 3818–3845, 2023.

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

[9] G. Sebastian, Z. Li, V. Crocher, D. Kremers, Y. Tan, and D. Oetomo, "Interaction force estimation using extended state observers: An application to impedance-based assistive and rehabilitation robotics," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1156–1161, Apr. 2019.

[10] J. Hu and R. Xiong, "Contact force estimation for robot manipulator using semiparametric model and disturbance Kalman filter," *IEEE Trans. Ind. Electron.*, vol. 65, no. 4, pp. 3365–3375, Apr. 2018.

[11] S. Liu, L. Wang, and X. V. Wang, "Sensorless force estimation for industrial robots using disturbance observer and neural learning of friction approximation," *Robot. Comput.- Integr. Manuf.*, vol. 71, 2021, Art. no. 102168.

[12] A. De Luca and G. Ulivi, "Iterative learning control of robots with elastic joints," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1992, pp. 1920–1926.

[13] A. De Luca and S. Panzieri, "A simple iterative scheme for learning gravity compensation in robot arms," in *Proc. 36th ANIPLA Annual Conf. (Automation 1992)*, 1992, Genova, I, pp. 459–471.

[14] A. de Luea and S. Panzieri, "An iterative scheme for learning gravity compensation in flexible robot arms," *IFAC Proc. Volumes*, vol. 26, no. 2, pp. 575–582, 1993.

[15] S. Basovich, S. A. Arogeti, Y. Menaker, and Z. Brand, "Magnetically levitated six-DOF precision positioning stage with uncertain payload," *IEEE/ASME Trans. Mechatron.*, vol. 21, no. 2, pp. 660–673, Apr. 2016.

[16] H. Ji, S. A. Arogeti, Y. Menaker, and Z. Brand, "Improving teleoperation through human-aware haptic feedback: A distinguishable and interpretable physical interaction based on the contact state," *IEEE Trans. Human-Mach. Syst.*, vol. 53, no. 1, pp. 24–34, Feb. 2022.

[17] J.-J. E. Slotine and W. Li, "On the adaptive control of robot manipulators," *Int. J. Robot. Res.*, vol. 6, no. 3, pp. 49–59, 1987.

[18] S. Ito, M. Darainy, M. Sasaki, and D. J. Ostry, "Computational model of motor learning and perceptual change," *Biol. Cybern.*, vol. 107, pp. 653–667, 2013.

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

[20] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Berlin, Germany: Springer, 2010.

[21] 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.

[22] 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 high performance control," *Presence: Teleoperators Virtual Environ.*, vol. 11, no. 6, pp. 555–568, 2002.

Teng Li (Graduate Student Member, IEEE) received the M.E. degree in mechanical manufacturing and automation from the Tianjin University of Science and Technology, Tianjin, China, in 2014 and the Ph.D. degree in mechanical design and theory from Beihang University, Beijing, China, in 2019. He is currently working toward the Ph.D. degree in electrical and computer engineering with the University of Alberta, Edmonton, AB, Canada.

His research interests include surgical robotics, robot control systems, physical human–robot interaction, and haptics.

Amir Zakerimanesh received the B.Sc. and M.Sc. degrees in electrical engineering (control-system) from the Faculty of Electrical and Computer Engineering, University of Tabriz, Tabriz, Iran, in 2012 and 2015, respectively, and the Ph.D. degree in control systems from the University of Alberta, Edmonton, AB, Canada, in 2024.

He is currently a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Alberta. His research interests include control engineering, robotics, machine learning, teleoperation systems, intelligent transportation systems, and computer vision.

Yafei Ou received the B.Sc. degree in mechanical design, manufacturing and automation from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2021. He is currently working toward the Ph.D. degree in electrical and computer engineering with the University of Alberta, Edmonton, AB, Canada.

His research interests focus on surgical robotics and automation.

Armin Badre received the graduation degree from the Faculty of Medicine, University of Alberta, Edmonton, AB, Canada, in 2012, and the M.Sc. degree in surgery, with a focus on elbow biomechanics, from Roth McFarlane HULC, London, ON, Canada, in 2019.

In 2017, he completed his Orthopaedic Surgery training at the University of Alberta. Upon graduation, he went to the Roth McFarlane HULC, a world-renowned upper extremity specialized center, to subspecialize in the management of complex elbow, hand, and wrist reconstruction and trauma.

He joined the Western Hand and Upper Limb Facility (WULF), Sturgeon Hospital, St. Albert, AB, in 2019. His clinical practice is focused on the management of various elbow, hand, and wrist conditions, including arthroscopy, arthroplasty, and upper extremity trauma. He has an academic appointment with the Faculty of Medicine, University of Alberta, and is involved with the teaching of medical students, residents, and fellows. He is quite keen on the advancement of knowledge through high-quality clinical and biomechanical research and is currently the research lead with WULF. He has authored or coauthored his work in a number of prestigious journals and presented at various national and international scientific meetings.

Mahdi Tavakoli (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Western Ontario, London, ON, Canada, in 2005.

He is currently a Professor with the Electrical and Computer Engineering Department and the Biomedical Engineering Department and a Senior University of Alberta Engineering Research Chair in Healthcare Robotics. He is also a Scientific Vice-Director for the Institute for Smart Augmentative and Restorative Technologies (iSMART), University of Alberta, Edmonton, AB, Canada. From 2006 to 2008, he was a Postdoctoral Researcher with Canadian Surgical Technologies and Advanced Robotics (CSTAR), London, ON, Canada, and an NSERC Postdoctoral Fellow with Harvard University, Cambridge, MA, USA. He is the lead author of *Haptics for Teleoperated Surgical Robotic Systems* (World Scientific, 2008) and the Specialty Chief Editor for *Frontiers in Robotics and AI* (Robot Design Section). His research interests involve medical robotics, image-guided surgery, and rehabilitation robotics.

Dr. Tavakoli is currently an Associate Editor for the *International Journal of Robotics Research*, IEEE TRANSACTIONS ON MEDICAL ROBOTICS AND BIONICS, IEEE/ASME TRANSACTIONS ON MECHATRONICS' Focused Section with Advanced Intelligent Mechatronics, and *Journal of Medical Robotics Research*.