Gaussian Process-Enhanced Impedance Iterative Learning for Robot Interaction Control

Yongping Pan
School of Advanced Manufacturing
Sun Yat-sen University
Shenzhen 518100, China
panyongp@mail.sysu.edu.cn

Wei Li
School of Software Engineering
Sun Yat-sen University
Guangzhou 510006, China
liwei363@mail2.sysu.edu.cn

Tian Shi
School of Computer Science and Engineering
Sun Yat-sen University
Guangzhou 510006, China
shit23@mail2.sysu.edu.cn

***Abstract*—Impedance control is an interaction control approach that allows robots to perform ideal characteristics similar to a mass-spring-damping model for safety and naturalness considerations. Iterative learning (IL) is effective in learning desired impedance parameters for robots under unstructured environments. In this paper, we propose an enhanced impedance IL method to speed up learning convergence and reduce the space complexity of data storage, where a sparse online Gaussian process is used to model a variable impedance strategy and is updated in the same iteration by removing similar data points from previous iterations while learning impedance parameters in multiple iterations. Numerical results based on a robot manipulator with 7 degrees of freedom have shown that the proposed method accelerates iterative convergence compared to the classical impedance IL method.**

***Index Terms*—Impedance variation, iterative learning, Gaussian process, robot learning, interaction control.**

I. Introduction

Impedance control is a major interaction control approach that allows robots to perform ideal characteristics similar to a mass-spring-damping model to ensure safe and natural interaction [1]. In many situations, impedance control with fixed impedance parameters does not meet task requirements [2], and robots may be required to change impedance parameters during continuous interaction with environments. Nevertheless, efficiently learning and adjusting robot impedance parameters to adapt to dynamic environments is still challenging.

Iterative learning (IL) is effective in impedance learning for robots under unknown environments, where desired impedance parameters are iteratively updated during robot-environment interaction to minimize an objective function so as to achieve the desired robot behavior [3], [4]. But IL has two major drawbacks: First, it needs to be re-iterated if the initial position or desired trajectory is changed [5]; second, there is no suitable way to guide the learning rate setting, making it difficult to ensure the convergence rate [4]. Gaussian process (GP) is a nonparametric Bayesian approach that has the ability to model complicated functions using only a limited number of data [6], and sparse online GP (SOGP) has the online learning ability to update the GP model with arriving time-series data [7]–[10].

This study proposes an enhanced impedance learning method to accelerate impedance learning convergence, where an SOGP is used to model a variable impedance strategy and is updated in the same iteration by removing similar data points from previous iterations while learning impedance parameters in multiple iterations. During the same iteration, impedance parameters learned at the current instant can immediately affect the prediction at the next instant, speeding up the convergence process. Besides, the variable impedance strategy resulting from the proposed IL-SOGP method is not task-dependent such that it can effectively exploit the original knowledge when the initial robot position or task trajectory is changed, and thus, the number of iterations can be shortened to accommodate a new task.

Through this paper, $\mathbb{R}$, $\mathbb{R}^+$, $\mathbb{R}^n$, and $\mathbb{R}^{m\times n}$ denote the spaces of real numbers, positive real numbers, real n-vectors, and real $m \times n$-matrices, respectively, $\|\boldsymbol{x}\|_2$ and $\|\boldsymbol{x}\|_\infty$ denote the 2-norm and ∞-norm of $\boldsymbol{x}$, respectively, $\mathcal{GP}(m(\boldsymbol{x}), \kappa(\boldsymbol{x}, \boldsymbol{x}'))$ is a GP model with a mean function $m : \mathbb{R}^n \mapsto \mathbb{R}$ and a kernel function $\kappa : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}$, $\mathrm{diag}(x_1, x_2, \cdots, x_n)$ is a diagonal matrix with diagonal elements x_1 to x_n, I is an identity matrix with a proper dimension, where $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n$, $x_i \in \mathbb{R}$, $i = 1$ to n, and n and m are positive integers.

II. The Proposed Method

A. Iterative Impedance Learning Framework

Consider a robot-environment interaction problem, where the robot kinematics is described by

$$\boldsymbol{x} = \boldsymbol{f}(\boldsymbol{q}),\ \dot{\boldsymbol{x}} = J(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^6$ is the pose in the Cartesian space, $\boldsymbol{q}(t) \in \mathbb{R}^n$ is the position in the joint space, $\boldsymbol{f} : \mathbb{R}^n \mapsto \mathbb{R}^6$ maps from the joint to Cartesian spaces, $J(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$ is the Jacobian matrix, and n is the number of degrees of freedom (DoFs). The robot dynamics is described by

$$M(\boldsymbol{q})\ddot{\boldsymbol{q}} + C(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + G(\boldsymbol{q}) + F(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau} - J^T(\boldsymbol{q})\boldsymbol{f}_e, \tag{2}$$

where $M(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is an inertia matrix, $C(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n\times n}$ is a centripetal-Coriolis matrix, $F(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^n$ denotes a friction torque, $G(\boldsymbol{q}) \in \mathbb{R}^n$ is a gravitational torque, $\boldsymbol{\tau} \in \mathbb{R}^n$ is a control torque, and $\boldsymbol{f}_e(t) \in \mathbb{R}^6$ is an interaction force at the robot end-effector that can be measured by a force/torque sensor.

This work was supported in part by the Fundamental Research Funds for the Central Universities, Sun Yat-sen University, China, under Grant 23lgzy004 (Corresponding Author: Yongping Pan).

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00121

An unknown environment is represented by a mass-damping-spring model in the Cartesian space as follows [11]:

$$M_{\mathrm{e}}(t)\ddot{\boldsymbol{x}} + B_{\mathrm{e}}(t)\dot{\boldsymbol{x}} + K_{\mathrm{e}}(t)\boldsymbol{x} = \boldsymbol{f}_{\mathrm{e}}, \tag{3}$$

in which $\boldsymbol{x}(t) \in \mathbb{R}^6$ is the pose of the environment, and $M_e(t) \in \mathbb{R}^{6\times6}$, $B_e(t) \in \mathbb{R}^{6\times6}$, and $K_e(t) \in \mathbb{R}^{6\times6}$ are diagonal inertia, damping and stiffness matrices of the environment, respectively. Then, (3) can be rewritten into a state-space form

$$\begin{cases}\dot{\boldsymbol{\chi}} = A(t)\boldsymbol{\chi} + B(t)\boldsymbol{f}_{\mathrm{e}} \\ \boldsymbol{v} = C\boldsymbol{\chi}\end{cases} \tag{4}$$

with $\boldsymbol{v}(t) \in \mathbb{R}^6$ and $\boldsymbol{\chi}(t) := [\boldsymbol{x}^T(t), \dot{\boldsymbol{x}}^T(t), (\int_0^t \boldsymbol{f}_{\mathrm{e}}(\varsigma)\mathrm{d}\varsigma)^T]^T \in \mathbb{R}^{18}$, where $A(t) \in \mathbb{R}^{18\times18}$ and $B(t) \in \mathbb{R}^{18\times6}$ are given by

$$A = \begin{bmatrix}\mathbf{0} & I & \mathbf{0} \\ -M_{\mathrm{e}}^{-1}K_{\mathrm{e}} & -M_{\mathrm{e}}^{-1}B_{\mathrm{e}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix}, B = \begin{bmatrix}\mathbf{0} \\ -M_{\mathrm{e}}^{-1} \\ I\end{bmatrix}$$

and $C \in \mathbb{R}^{6\times18}$ is determined based on the desired interaction effect. Different interaction effects can be achieved by setting a proper C to change the expected output $\boldsymbol{v}$. When the elements corresponding to the position $\boldsymbol{x}$ in C are set larger, robots are in pursuit of position tracking, and the force tracking performance may be degraded. In scenarios focusing on compliant interaction, the elements regarding the integrated force $(\int_0^t \boldsymbol{f}_{\mathrm{e}}(\varsigma)\mathrm{d}\varsigma)$ in C can be chosen larger and the position tracking error may increase. For (4), the interaction force $\boldsymbol{f}_{\mathrm{e}}(t)$ is iteratively updated by

$$\boldsymbol{f}_{\mathrm{e}}^k(t) = \boldsymbol{f}_{\mathrm{e}}^{k-1}(t) + \alpha'[\dot{\boldsymbol{v}}_{\mathrm{d}}(t) - \dot{\boldsymbol{v}}^k(t)] \tag{5}$$

with $\boldsymbol{v}^k(0) = \boldsymbol{v}_{\mathrm{d}}(0)$, where k is the iteration number, $\boldsymbol{v}_{\mathrm{d}}(t) \in \mathbb{R}^6$ is the desired output, and $\alpha' \in \mathbb{R}^+$ satisfies $\|I - \alpha' BC\|_\infty < 1$ with CB being nonsingular. Then, it follows from the generic betterment scheme [4] that $\boldsymbol{v}^k(t) \to \boldsymbol{v}_{\mathrm{d}}(t)$ uniformly in $t \in [0, t_f]$ as $k \to \infty$, where $t_f \in \mathbb{R}^+$ is the iteration period.

The target impedance model is described by

$$M_{\mathrm{d}}(t)(\ddot{\boldsymbol{x}}_{\mathrm{r}} - \ddot{\boldsymbol{x}}_{\mathrm{d}}) + B_{\mathrm{d}}(t)(\dot{\boldsymbol{x}}_{\mathrm{r}} - \dot{\boldsymbol{x}}_{\mathrm{d}}) + K_{\mathrm{d}}(t)(\boldsymbol{x}_{\mathrm{r}} - \boldsymbol{x}_{\mathrm{d}}) = \boldsymbol{f}_{\mathrm{e}}, \tag{6}$$

where $\boldsymbol{x}_{\mathrm{d}}(t) \in \mathbb{R}^6$ and $\boldsymbol{x}_{\mathrm{r}}(t) \in \mathbb{R}^6$ are desired and reference poses of the end-effector, respectively, and $M_{\mathrm{d}}(t) \in \mathbb{R}^{6\times6}$, $B_{\mathrm{d}}(t) \in \mathbb{R}^{6\times6}$ and $K_{\mathrm{d}}(t) \in \mathbb{R}^{6\times6}$ are desired inertia, damping, and stiffness parameters, respectively. Consider an objective function

$$\mathcal{L}(t) = \|\boldsymbol{v}(t) - \boldsymbol{v}_{\mathrm{d}}(t)\|_2. \tag{7}$$

Then, B_{d} and K_{d} can be iteratively updated by

$$\begin{cases}B_{\mathrm{d}}^k(t) = B_{\mathrm{d}}^{k-1}(t) - \beta\left(\frac{\partial \boldsymbol{f}^k(t)}{\partial B_{\mathrm{d}}^k(t)}\right)^T\left(\frac{\partial \mathcal{L}^k(t)}{\partial \boldsymbol{f}_e^k(t)}\right)^T \\ K_{\mathrm{d}}^k(t) = K_{\mathrm{d}}^{k-1}(t) - \beta\left(\frac{\partial \boldsymbol{f}_e^k(t)}{\partial K_{\mathrm{d}}^k(t)}\right)^T\left(\frac{\partial \mathcal{L}^k(t)}{\partial \boldsymbol{f}_e^k(t)}\right)^T\end{cases} \tag{8}$$

with $\beta \in \mathbb{R}^+$ being a learning rate. Note that M_{d} is not updated as arbitrarily setting M_{d} may cause instability. It is clear to get $(\partial \boldsymbol{f}^k(t)/\partial K_{\mathrm{d}}^k(t))^T = \boldsymbol{e}^k(t)$ and $(\partial \boldsymbol{f}^k(t)/\partial B_{\mathrm{d}}^k(t))^T = \dot{\boldsymbol{e}}^k(t)$, in which $\boldsymbol{e}^k(t) := \boldsymbol{x}^k(t) - \boldsymbol{x}_{\mathrm{d}}(t)$ is a position tracking error at the kth iteration. According to (5), one has $\boldsymbol{v}^k(t) \to \boldsymbol{v}_{\mathrm{d}}(t)$ such that the objective function $\mathcal{L}$ can be minimized. Therefore, B_{d}^k and K_{d}^k can be iteratively updated by

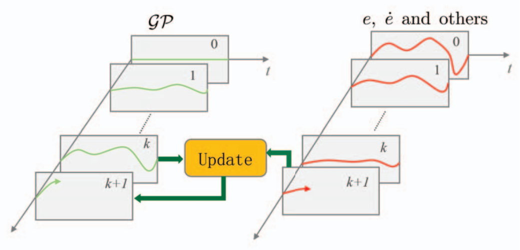


Fig. 1. The fundamental principle of the proposed IL-SOGP impedance learning method. Note that the proposed method models a variable impedance strategy with an iterative update of the SOGP, where the update part includes the task level (k-iteration) and the time level (t-axis).

$$\begin{cases}B_{\mathrm{d}}^k(t) = B_{\mathrm{d}}^{k-1}(t) - \alpha\dot{\boldsymbol{e}}^k(t)(\dot{\boldsymbol{v}}^k(t) - \dot{\boldsymbol{v}}_{\mathrm{d}}(t))^T \\ K_{\mathrm{d}}^k(t) = K_{\mathrm{d}}^{k-1}(t) - \alpha\boldsymbol{e}^k(t)(\dot{\boldsymbol{v}}^k(t) - \dot{\boldsymbol{v}}_{\mathrm{d}}(t))^T\end{cases} \tag{9}$$

where $\alpha \in \mathbb{R}^+$ is a learning rate that absorbs β and α'. In summary, the impedance parameters B_{d} and K_{d} can be iteratively updated by (9) to minimize $\mathcal{L}$. However, the impedance IL needs to store learned impedance parameters at every instant of the previous iteration and is applicable only for a single task [5].

B. Impedance Learning with Gaussian Process

The basic principle of the proposed IL-SOGP method for impedance learning can be seen in Fig. 1. Here, we apply the position $\boldsymbol{x}$ and velocity $\dot{\boldsymbol{x}}$ at the robot end-effector, the desired position $\boldsymbol{x}_{\mathrm{d}}$ and velocity $\dot{\boldsymbol{x}}_{\mathrm{d}}$, and the interaction force $\boldsymbol{f}_{\mathrm{e}}$ at the robot end-effector to estimate the impedance parameters B and K [12]. To enhance the generalization ability of IL-based impedance learning and to reduce the space complexity of data storage, we define a variable impedance strategy

$$\begin{cases}B_{\mathrm{d}}(t) = \pi_{B_{\mathrm{d}}}(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), \boldsymbol{x}_{\mathrm{d}}(t), \dot{\boldsymbol{x}}_{\mathrm{d}}(t), \boldsymbol{f}_{\mathrm{e}}(t)) \\ K_{\mathrm{d}}(t) = \pi_{K_{\mathrm{d}}}(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), \boldsymbol{x}_{\mathrm{d}}(t), \dot{\boldsymbol{x}}_{\mathrm{d}}(t), \boldsymbol{f}_{\mathrm{e}}(t))\end{cases} \tag{10}$$

where $\pi_{B_{\mathrm{d}}}$ and $\pi_{K_{\mathrm{d}}}$ are represented by iterative GPs [13]

$$\pi_{B_{\mathrm{d}}} \sim \mathcal{GP}^k(0, \kappa(\boldsymbol{z}, \boldsymbol{z}')),\ \pi_{K_{\mathrm{d}}} \sim \mathcal{GP}^k(0, \kappa(\boldsymbol{z}, \boldsymbol{z}')) \tag{11}$$

with $\boldsymbol{z}, \boldsymbol{z}' \in \mathcal{Z} := \{(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), \boldsymbol{x}_{\mathrm{d}}(t), \dot{\boldsymbol{x}}_{\mathrm{d}}(t), \boldsymbol{f}_{\mathrm{e}}(t)) | t \geq 0\}$, in which $\mathcal{GP}^k$ denotes the GP at the kth iteration, and $\kappa(\boldsymbol{z}, \boldsymbol{z}') : \mathbb{R}^{18} \times \mathbb{R}^{18} \to \mathbb{R}$ is a Gaussian kernel function that represents a measure of the correlation of two points $(\boldsymbol{z}, \boldsymbol{z}')$. We employ an SOGP because it can provide stable prediction while keeping a small size of the representative data set. Later, this set will be called a basis vector ($\mathcal{BV}$) set. Note that the proposed IL-SOGP only requires the signals $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$, $\boldsymbol{x}_{\mathrm{d}}$, $\dot{\boldsymbol{x}}_{\mathrm{d}}$, and $\boldsymbol{f}_{\mathrm{e}}$, where $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ can be measured by encoders, $\boldsymbol{x}_{\mathrm{d}}$ and $\dot{\boldsymbol{x}}_{\mathrm{d}}$ are desired signals, and $\boldsymbol{f}_{\mathrm{e}}$ is generated by the target impedance model (6).

SOGP keeps the $\mathcal{BV}$ set based on the metric of data points in the reproducing kernel Hilbert space (RKHS) by adding and deleting data schemes. In the update of a SOGP, for a new arrival point $\boldsymbol{z}_* \in \mathcal{Z}$, the residual error γ_{z_*} is given by [7]

$$\gamma_{z_*} = \mathcal{K}_{z_* z_*} - \mathcal{K}_{z_* Z}\mathcal{K}_{ZZ}^{-1}\mathcal{K}_{z_* Z}^{T} \qquad (12)$$

where Z collects all points in the $\mathcal{BV}$ set, and $\mathcal{K}_{ZZ}$ denotes a covariance matrix. Note that γ_{z_*} can be seen as the "novelty" measure of $\boldsymbol{z}_*$ for the $\mathcal{BV}$ set. If γ_{z_*} is greater than a threshold $\epsilon \in \mathbb{R}^+$, $\boldsymbol{z}_*$ is added to the $\mathcal{BV}$ set. Then, if the size of the $\mathcal{BV}$ set is excessive, a data point is selected for deletion based on an RKHS-based similar metric.

It is not feasible to use SOGPs directly for modeling the variable impedance strategy in (10). With the continuous iterations of the task, the SOGP will not be updated for similar inputs. Hence, a pre-deletion operation can be added at the beginning of the update process to delete the points that are similar to the previous iterations. We introduce a hyperparameter $w \in [0, 1]$ as the threshold of similarity and delete the points whose similarity exceeds the threshold w in the $\mathcal{BV}$ set. Let $\boldsymbol{z}_i^{k-1}$ be a data point in the $\mathcal{BV}$ set before the previous $(k-1)$th iteration and $\boldsymbol{z}^k \in \mathcal{Z}$ be a new arrival point at the current kth iteration, where $i = 1$ to N, and N denotes the size of the $\mathcal{BV}$ set. Then, the similarity between the two data points is measured as follows:

$$d_i = \kappa(\boldsymbol{z}_i^{k-1}, \boldsymbol{z}^k), i = 1 \text{ to } N, \qquad (13)$$

where the data points $\boldsymbol{z}_i^{k-1}$ with the similarity $d_i > w$ compared to the new arrival point $\boldsymbol{z}^k$ need to be deleted before each update of the SOGP. Thus, the proposed IL-SOGP method has an extra computational burden compared to the classical IL method, because of the selection and deletion operations of data points in SOGP. The detailed algorithm is shown in Algorithm 1.

Let t be the "current" instant at any iteration cycle. If w is set too small, it causes the deletion of many data points in the $\mathcal{BV}$ set from the $(k-1)$th iteration. Although this allows learning new knowledge rapidly, leading to a lack of data points in the $\mathcal{BV}$ set beyond the instant t of the $(k-1)$th iteration, resulting in possibly less accurate prediction. If w is set too large, few data points will be deleted, which leads to a lack of data points from the current kth iteration. This means that the SOGP has difficulty learning new knowledge, so its predicted values converge to those in the $(k-1)$th iteration. The setting of w is related to the hyperparameters of the kernel function κ, which can be a value where some newly arriving data points are inserted into the $\mathcal{BV}$ set. With the exploitation of both the knowledge of the current kth iteration and the previous $(k-1)$ iterations, the proposed IL-SOGP is expected to have better learning convergence than the original IL. Note that stability analysis is not a focus of this study, and it can be referred to [14].

Algorithm 1 Update process of IL-SOGP in kth iteration

```
Require: $\boldsymbol{z}^k$, $\mathcal{BV}$ set, $\epsilon$
  Compute $d_i = \kappa(\boldsymbol{z}_i^{k-1}, \boldsymbol{z}^k)$ by (13)
  Delete the corresponding point in $\mathcal{BV}$ set with $d_i > w$
  Compute the residual error $\gamma_{\boldsymbol{z}^k}$ by (12)
  if $\gamma_{\boldsymbol{z}^k} > \epsilon$ then
      Add the new arrival point $\boldsymbol{z}^k$ to $\mathcal{BV}$ set
  else
      Only update the corresponding variables of SOGP
  end if
  if size($\mathcal{BV}$) $> N$ then
      Delete a point based on RKHS-based scheme [7]
  end if
```

III. Numerical Verification

A. Simulation Environment Setup

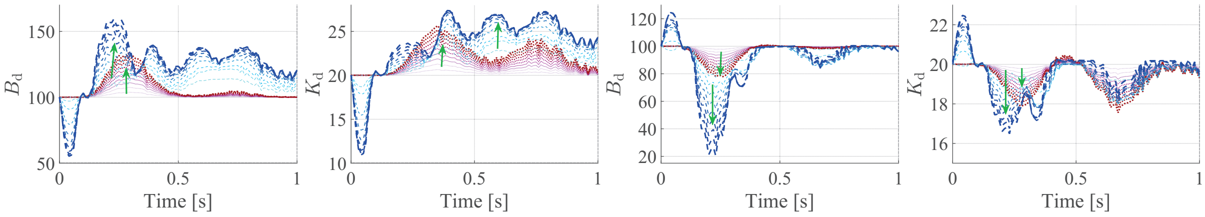
This section validates the convergence performance of the proposed IL-SOGP method based on a 7-DoF robot manipulator called Franka Emika Panda in MATLAB, where the dynamic parameters are from the identified result in [15], and more details on the robot kinematics can be seen in [16]. The inner loop uses a computed torque position controller with a sampling delay of 1 ms, and the outer-loop sampling delay for admittance control is 10 ms. We only consider variable impedance for the z-axis translation direction for clear illustrations.

The environment (3) is to be an uneven object with a height $h = 0.5 + 0.003 \sin(80\, x_z(t) + 1.2) + \varepsilon_n$, where $x_z(t) \in \mathbb{R}$ is the z-direction of $\boldsymbol{x}(t)$, $\varepsilon_n \sim \mathcal{N}(0, 0.0006^2)$ is Gaussian noise with mean 0 and variance 0.0006^2, $M_\mathrm{e} = 0.01\Phi(t)I$, $B_\mathrm{e} = 0.1\Phi(t)I$ and $K_\mathrm{e} = 450\Phi(t)I$ with $\Phi(t) = (0.5\sin(2\pi t) + 0.5)^2$ [4]. The initial position of the end-effector is $\boldsymbol{x}_0 = [0.555, 0, 0.515]^T$ m, the desired interaction force is $\boldsymbol{f}_\mathrm{d} = 0$ N, and the desired motion is the end-effector moving downward the object and sliding a distance after contact with the object surface.

The damping parameter B_d and the stiffness parameter K_d in the z-axis direction are modeled by two GPs as in (11), and are initialized as 100 and 20, respectively. Choose a squared exponential kernel $\kappa(\boldsymbol{z}, \boldsymbol{z}') = \sigma_s^2 \exp(-(\boldsymbol{z}-\boldsymbol{z}')^T\Lambda(\boldsymbol{z}-\boldsymbol{z}')/2)$ and set $w = 0.8$, $\epsilon = 0.001$, $N = 15$, $\sigma_s = 1$, $\sigma_n = 0.001$, and $\Lambda = \mathrm{diag}(3, 3, 3, 3, 4)$ for both GPs, where $\sigma_s^2 \in \mathbb{R}^+$ is a signal variance, and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_5)$ with $\lambda_i \in \mathbb{R}^+$ ($i = 1$ to 5) is a length-scale matrix of the input vector.

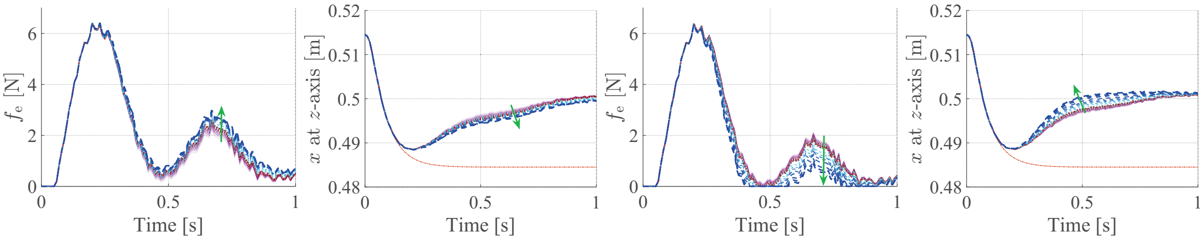
B. Comparison of Convergence Performance

The convergence performance of the classical IL method in [4] and the proposed IL-SOGP method is explored under two cases: Case 1 for accurate tracking with $\alpha = 10$ and $C = [25, 0, 1]$ and Case 2 for compliant interaction with $\alpha = 20$ and $C = [100, 0, 0.1]$. From Figs. 2 and 3, it is clear that in Case 1, the damping parameter B_d and the stiffness parameter K_d become larger for better position tracking, and in Case 2, B_d and K_d become smaller, resulting a smaller interaction force $\boldsymbol{f}_e$ for compliant interaction. Yet, the convergence rate of the proposed IL-SOGP is higher than that of the classical IL for both B_d and K_d. This is because the proposed IL-SOGP exploits the new iteration values learned from the previous instants in the same iteration, which provides a trend for the subsequent prediction and thus accelerates convergence. The proposed IL-SOGP exhibits more sharp changes of B_d and K_d, which is caused by the low frequency of the admittance control part and results in a sharp acceleration when data points are added to the $\mathcal{BV}$ set. But for the variable impedance control, these low-frequency changes are subtle, and no sudden changes in the



(a) Case1: $B_{\mathrm{d}}(t)$ (b) Case1: $K_{\mathrm{d}}(t)$ (c) Case2: $B_{\mathrm{d}}(t)$ (d) Case2: $K_{\mathrm{d}}(t)$

Fig. 2. Convergence performances by two learning methods, where red curves are by the classical IL method, blue curves are by the proposed IL-SOGP method, deeper color denotes higher iterative times, and arrows indicate the evolving direction of the iterations from $k = 1$ to 10.



(a) Case1: $\boldsymbol{f}_{\mathrm{e}}(t)$ (b) Case1: $\boldsymbol{x}$ at z-axis (c) Case2: $\boldsymbol{f}_{\mathrm{e}}(t)$ (d) Case2: $\boldsymbol{x}$ at z-axis

Fig. 3. Interaction processes by two learning methods, where red curves are by the classical IL, blue curves are by the proposed IL-SOGP, orange curves are the desired trajectories, deeper color denotes higher iterative times, and arrows indicate the evolving direction of the iterations from $k = 1$ to 10.

interaction force $\boldsymbol{f}_e$ or the robot position $\boldsymbol{x}$ will be introduced during the interaction. In summary, compared to the classical IL, the proposed IL-SOGP has better position tracking and faster iterations in Case 1 and has a smaller interaction force $\boldsymbol{f}_e$, resulting in more compliant interaction in Case 2.

IV. Conclusions

This paper has proposed an IL-SOGP impedance learning framework to improve the convergence speed during robot-environment interaction. Numerical verification has shown that compared with the classical impedance IL method, the proposed method provides better convergence in modeling the variable impedance strategy for a 7-DoF robot arm. However, hardware experiments of our method are still challenging due to the high computational cost of SOGP [17], and more efficient GP models can be resorted to overcome this problem in further studies.

References

[1] G. Kang, H. S. Oh, J. K. Seo, U. Kim, and H. R. Choi, "Variable admittance control of robot manipulators based on human intention," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 3, pp. 1023–1032, 2019.

[2] M. Sharifi, A. Zakerimanesh, J. K. Mehr, A. Torabi, V. K. Mushahwar, and M. Tavakoli, "Impedance variation and learning strategies in human-robot interaction," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6462–6475, 2022.

[3] R.-E. Precup, S. Preitl, J. K. Tar, M. L. Tomescu, M. Takacs, P. Korondi, and P. Baranyi, "Fuzzy control system performance enhancement by iterative learning control," *IEEE Trans. Ind. Electron.*, vol. 55, no. 9, pp. 3461–3475, 2008.

[4] Y. Li and S. S. Ge, "Impedance learning for robots interacting with unknown environments," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 4, pp. 1422–1432, 2014.

[5] S. Riaz, L. Hui, M. S. Aldemir, and F. Afzal, "A future concern of iterative learning control: A survey," *J. Stat. Manag. Syst.*, vol. 24, no. 6, pp. 1301–1322, 2021.

[6] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4405–4423, 2020.

[7] L. Csato and M. Opper, "Sparse on-line Gaussian processes," *Neural Comput.*, vol. 14, no. 3, pp. 641–668, 2002.

[8] J. S. De La Cruz, W. Owen, and D. Kulic, "Online learning of inverse dynamics via Gaussian process regression," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Vilamoura, Algarve, Portugal, 2012, pp. 3583–3590.

[9] L. Deng, W. Li, and Y. Pan, "Data-efficient Gaussian process online learning for adaptive control of multi-DoF robotic arms," *IFAC-PapersOnLine*, vol. 55, no. 2, pp. 84–89, 2022.

[10] W. Li, Z. Li, Y. Liu, and Y. Pan, "Learning inverse robot dynamics using sparse online Gaussian process with forgetting mechanism," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics*, Sapporo, Japan, 2022, pp. 638–643.

[11] K. Dupree, C.-H. Liang, G. Hu, and W. E. Dixon, "Adaptive Lyapunov-based control of a robot and mass–spring system undergoing an impact collision," *IEEE Trans. Syst. Man Cybern. B*, vol. 38, no. 4, pp. 1050–1061, 2008.

[12] C. Li, Z. Zhang, G. Xia, X. Xie, and Q. Zhu, "Efficient learning variable impedance control for industrial robots," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 67, no. 2, 2019.

[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT press, 2006.

[14] R.-E. Precup, R.-C. Roman, and A. Safaei, *Data-Driven Model-Free Controllers*. Boca Raton, USA: CRC press, 2021.

[15] C. Gaz, M. Cognetti, A. Oliva, P. R. Giordano, and A. De Luca, "Dynamic identification of the Franka Emika Panda robot with retrieval of feasible parameters using penalty-based optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 4147–4154, 2019.

[16] L. Deng, Z. Li, and Y. Pan, "Sparse online Gaussian process impedance learning for multi-DoF robotic arms," in *Proc. IEEE Int. Conf. Adv. Robot. Mechatron.*, Chongqing, China, 2021, pp. 199–206.

[17] B. Wilcox and M. C. Yip, "SOLAR-GP: Sparse online locally adaptive regression using Gaussian processes for Bayesian robot model learning and control," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2832–2839, 2020.