BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 67, No. 2, 2019
DOI: 10.24425/bpas.2019.128116

Efficient learning variable impedance control for industrial robots

C. LI, Z. ZHANG*, G. XIA, X. XIE, and Q. ZHU

College of Automation, Harbin Engineering University, Harbin 150001, China

Abstract. Compared with the robots, humans can learn to perform various contact tasks in unstructured environments by modulating arm impedance characteristics. In this article, we consider endowing this compliant ability to the industrial robots to effectively learn to perform repetitive force-sensitive tasks. Current learning impedance control methods usually suffer from inefficiency. This paper establishes an efficient variable impedance control method. To improve the learning efficiency, we employ the probabilistic Gaussian process model as the transition dynamics of the system for internal simulation, permitting long-term inference and planning in a Bayesian manner. Then, the optimal impedance regulation strategy is searched using a model-based reinforcement learning algorithm. The effectiveness and efficiency of the proposed method are verified through force control tasks using a 6-DoFs Reinovo industrial manipulator.

Key words: variable impedance control; reinforcement learning; efficient; Gaussian process; industrial robots.

1. Introduction

In recent decades, industrial robots are increasingly expected to complete operational tasks that involve physical contacts, such as grinding, deburring, robot-assisted operation and automatic assembly of explosive components. These tasks are usually sensitive to the contact force, and may be executed repeatedly in different environments. However, too many physical interactions with the environment are often infeasible and could lead to danger. It is a challenge for industrial robots to learn to perform tasks efficiently while accurately controlling the contact force in unstructured environments.

Impedance control [1] is a prominent interaction control approach. The core of the impedance control is to regulate dynamic response of the end-effector to contact force by establishing a suitable virtual mass-spring-damper system. In classical impedance control, the impedance parameters are fixed throughout the task. For increased flexibility, the impedance can be programmed to vary during the task [2–4]. Recently, many researchers have explored the benefits of varying the impedance during the tasks [5, 6]. Lee [7] designed a variable stiffness control scheme, and it achieves force tracking by adjusting the target stiffness without estimating the environment stiffness. The joint torque and the joint stiffness are independently and optimally modulated using the optimal variable stiffness control in [8]. Variable impedance control requires more complex task models than fixed impedance controlled systems, as the impedance profile must be represented. Deriving an effective variable impedance strategy usually involves advanced knowledge about designing and parameterizing such controllers [9]. Besides, manual programming of robot motions often requires a large amount of engineering knowledge about the robot and the task.

One promising way is to learn the control strategy automatically using reinforcement learning (RL). One of the primary goals of RL is to produce fully autonomous agents that interact with environments to learn optimal behaviors without human's participation [10, 11]. The RL problem is modeled as a finite horizon Markov Decision Process. The main idea of RL algorithm is that, given only a reward function, the learning algorithm finds suitable strategies that yield high reward through trial and error. The agent learns the desired task by gathering experience directly from its environment. The RL algorithms could be classified into two categories: model-free and model-based, and a review on RL for robotics is summarized in [12]. The model-free RL can be easier extended to high-dimensional spaces, and it is used more frequently in practice. The model-based RL is more data-efficient than the model-free RL, but more computationally intensive. Nagabandi [13] proposed to train multi-layer neural network dynamics models for model-based RL achieving satisfactory learning efficiency, and it realized learning of high-dimensional locomotion tasks. PILCO [14, 15] is a promising model-based RL algorithm for physical platforms given its natural handling of continuous states and controls. Its strategy output is directly used to control the plant and it achieves excellent sample efficiency relative to existing methods by learning probabilistic Gaussian process model. The first revolution in RL was the development of an algorithm that could learn to play a variety of Atari 2600 video games at a superhuman lever [16]. Especially, AlphaGo Zero [17] achieved superhuman performance in the game of Go based solely on RL, achieving the state-of-the-art performance. In real world, deep RL has been used for robots to learn to perform physics experiments [18] and manipulation tasks [19]. Most applications of RL in robotics focus on trajectories learning, and few involve the learning of contact force control.

*e-mail: zhangzhi1981@hrbeu.edu.cn

Manuscript submitted 2018-05-09, revised 2018-07-02, initially accepted for publication 2018-08-04, published in April 2019.

Data-driven RL approach appears as a promising route to learn to regulate the impedance properties of the robot and to control the contact force automatically. Several frameworks have already emerged that are capable of learning compliant behavior in this fashion [20–24] and have demonstrated the usefulness of learning variable impedance control. Kronander [22] and Li [25] addressed the problem of compliance adjusting in robot learning from demonstrations (RLfD), in which a robot could learn to adapt the stiffness based on human-robot interaction. Du [26] proposed a variable admittance control method based on fuzzy RL for physical human-robot interaction using a minimally invasive surgery manipulator. It improves the positioning accuracy and reduces the required energy by dynamically regulating the virtual damping in the admittance controller. Buchli proposed a novel model-free RL algorithm, PI^2 [27–29], which realized variable impedance control by regulating the motion trajectory and impedance gains simultaneously using dynamic movement primitives. Considering the coupling between DoFs, Winter [30] developed a C-PI^2 algorithm based on PI^2, whose learning speed was much higher than that of previous algorithms. Only 120 rollouts are required to get the satisfactory strategy.

The existing learning variable impedance control methods are usually based on model-free RL algorithm, and hundreds or thousands of interactions are required to achieve good performance. However, their high sample complexity has limited these methods to learn the force-sensitive tasks, because too many interactions maybe cause damage or danger. Improving the efficiency of learning variable impedance control is critical for industrial robots to learn to perform repetitive force-sensitive tasks.

In this paper, we propose an efficient learning variable impedance control method to make the industrial robots effectively learn to control the contact force accurately in the unstructured environment. This method attempts to construct a nonparametric Gaussian process (GP) model as the transition dynamics of the robot-environment. The probabilistic model is then used to predict and pass the uncertainties of the states in a Bayesian manner. Here, the model-based RL algorithm is used to search the optimal control strategy, taking full advantage of its data-efficiency [12, 13]. This method regulates the target stiffness and damping directly, instead of the motion trajectory. The performance of the proposed method is verified through experiments on a 6-DoFs industrial manipulator. This method outperforms other learning variable impedance control methods by at least one order of magnitude in terms of learning speed.

2. Position-based variable impedance control for industrial robots

Impedance control [1] has been widely applied in robotic interaction tasks for its good adaptability and robustness. It provides a unified framework for both constrained and unconstrained motion. The contact force can be controlled indirectly by establishing a suitable virtual mass-spring-damper model. Generally, the impedance control methods can be classified as two types. In force-based impedance control, the joint torques, which are usually calculated using the inverse dynamics, are controlled according to the end-effector displacement. In position-based impedance control, the compliant positions of the robot, which are usually calculated using the kinematics, are controlled according to the measured contact force. However, most commercial industrial robots emphasize the accuracy of position trajectory following, and only provide a position control mode for users. It implies that the compliant motion control have to be realized using the robot kinematics and the joint position controller of the robot. Therefore, force-based impedance control was impossible on these robots. Alternatively, position-based impedance control is recognized as a practical approach to achieve compliant interaction of position controlled robots.

The specification of impedance parameter is highly dependent on tasks. It is easy to specify the appropriate fixed impedance parameters for tasks executed in a structured environment with known characteristics, while it is extremely difficult to complete complex force control tasks because of the environmental conditions, including nonlinear and time-varying factors. If the impedance parameters could be regulated dynamically according to the task and the environment, the control performance will be significantly improved.

In the following, we consider the general industrial robotic manipulators that only provide a position control mode for users. The force can be controlled indirectly using the position-based impedance control scheme. This control structure includes an inner position control loop and an outer indirect force control loop. In order to achieve the desired dynamic properties of the end-effector, a second-order impedance model is used:

$$M_d\left(\ddot{X}-\ddot{X}_d\right)+B_d\left(\dot{X}-\dot{X}_d\right)+K_d\left(X-X_d\right)=F-F_d, \quad (1)$$

where M_d, B_d, and K_d are the positive definite matrices of the desired inertia, damping and stiffness of the impedance model, respectively. $\ddot{X}$, $\dot{X}$, and X denote the actual acceleration, velocity, and position of the end-effector in Cartesian space, respectively, while $\ddot{X}_d$, $\dot{X}_d$, and X_d are the desired acceleration, velocity, and position; F_d is the desired contact force, and F is the actual contact force.

The transfer-function of the impedance model is:

$$H(s)=\frac{\delta X(s)}{E(s)}=\frac{1}{M_d s^2+B_d s+K_d}, \quad (2)$$

where $E(s)$ is the error of the contact force. To compute the desired position increment, discretize (2) using bilinear transformation:

$$H(z)=H(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}}=\frac{T^2(z+1)^2}{\omega_1 z^2+\omega_2 z+\omega_3}, \quad (3)$$

$$\omega_1=4M_d+2B_dT+K_dT^2, \quad (4)$$

$$\omega_2=-8M_d+2K_dT^2, \quad (5)$$

$$\omega_3=4M_d-2B_dT+K_dT^2. \quad (6)$$

Here, T is the control cycle. The desired position increment of the end-effector is derived as follows:

$$\delta X(n) = \omega_1^{-1}\{T^2[E(n) + 2E(n-1) + E(n-2)] - \\ - \omega_2 \delta X(n-1) - \omega_3 \delta X(n-2)\}. \tag{7}$$

To simplify the calculation, the target inertial matrix is chosen as $M_d = I$. Consequently, the target stiffness K_d and the damping B_d are the only parameters that should be tuned in variable impedance control.

3. Scheme of the efficient learning variable impedance control

Learning variable impedance control is an ideal compliant control method, which can automatically get the optimal task-specific control strategy through trial-and-error. According to the learned strategy, the variable impedance controller automatically regulates the impedance parameters to track the desired contact force.

The scheme of the proposed efficient learning variable impedance control is illustrated in Fig. 1. According to the sampled data, a probabilistic GP model is approximated to simulate the dynamics of the system. Then the model-based reinforcement learning algorithm is used to search the optimal impedance control strategy π while predicting the system evolution using the GP model. The impedance control parameters $u = [K_d \ B_d]$ are calculated using the learned strategy, which are then transferred to the variable impedance controller to control the force.

The desired position increments of the end-effector δX are calculated according to the force error F_e. X_d is the desired reference trajectory. The desired joints positions q_d are calculated using inverse kinematics. The actual Cartesian position of the end-effector X could be achieved using the measured joints positions q by means of forward kinematics. K_E and B_E are the unknown stiffness and damping of the environment, respectively.

The variable impedance control strategy is defined as $\pi : x \mapsto u = \pi(x, \theta)$, where the inputs of the control strategy $x = [X \, F] \in \mathbb{R}^D$ are the observed states, the outputs of the control strategy are target stiffness K_d and damping B_d which can be written as matrix $u = [K_d \, B_d] \in \mathbb{R}^F$, and θ are the control strategy parameters that to be learned. Here, the GP controller is chosen as the control strategy π:

$$\pi_t = \pi(x_t, \theta) = \sum_{i=1}^{n} \beta_{\pi,i} k(x_\pi, x_t) = \beta_\pi^T K(x_\pi, x_t), \tag{8}$$

$$\beta_\pi = \left(K_\pi(X_\pi, X_\pi) + \sigma_{\varepsilon,\pi}^2 I\right)^{-1} y_\pi, \tag{9}$$

$$k(x_\pi, x_t) = \sigma_{f,\pi}^2 \exp\left(-\frac{1}{2}(x_{\pi,i} - x_t)^T \Lambda^{-1}(x_{\pi,i} - x_t)\right), \tag{10}$$

where $X_\pi = [x_{\pi,1}, \ldots, x_{\pi,n}]$ are the training inputs, and they are the centers of the Gaussian basis functions. n is the number of the basis functions. y_π is the training targets, which are initialized to values close to zero. $\Lambda = diag(l_1^2, \ldots, l_D^2)$ is the length scales, $\sigma_{f,\pi}^2$ is the signal variance, which is fixed to one here, $\sigma_{\varepsilon,\pi}^2$ is the measurement noise variance. $\theta = [X_\pi, y_\pi, l_1, \ldots, l_D, \sigma_{f,\pi}^2, \sigma_{\varepsilon,\pi}^2]$ is the hyper-parameters of the controller. Using the GP controller, more advanced nonlinear tasks could be performed thanks for its flexibility and smoothing effect. Obviously, the GP controller is functionally equivalent to a regularized RBF network if $\sigma_{f,\pi}^2 = 1$ and $\sigma_{\varepsilon,\pi}^2 \neq 0$. The impedance parameters are calculated in real-time according to the variable impedance control strategy π and the states x_t. The relationship between the impedance parameters and the control strategy can be written as follows:

$$[K_d \ \ B_d] = u = \pi(x_t, \theta) = \beta_\pi^T K(x_\pi, x_t). \tag{11}$$

For a Gaussian distributed state $x_t \sim \mathcal{N}(\mu_t, \Sigma_t)$, the mean of u, which is transferred to the variable impedance controller, can be calculated as:

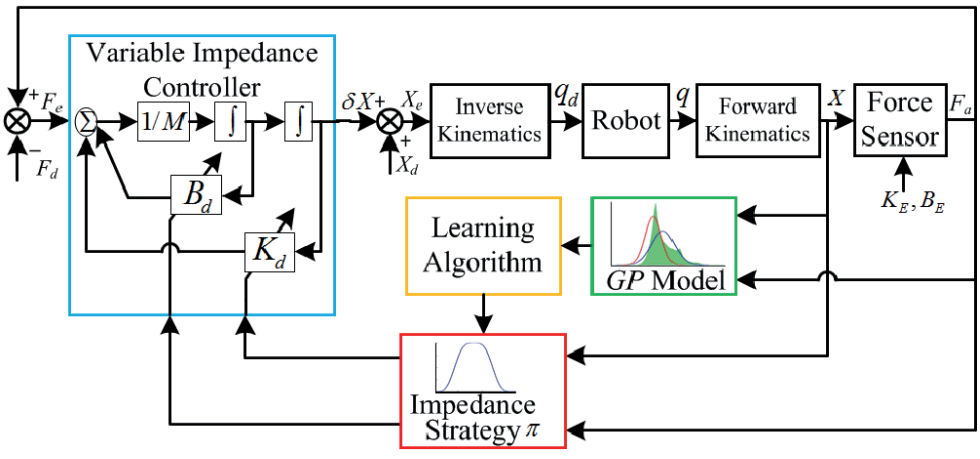

Fig. 1. Scheme of the efficient learning variable impedance control

$$\mathbb{E}[u] = \mathbb{E}[\pi(x_t, \theta)] = \beta_\pi^T \mathbb{E}[K(X_\pi, x_t)] = \\ = \beta_\pi^T \int K(X_\pi, x_t) p(x_t) dx_t = \beta_\pi^T q_a, \tag{12}$$

$$q_{a_i} = \frac{\exp\left[-\frac{1}{2}(\mu_t - x_i)^T (\Sigma_t + \Lambda)^{-1} (\mu_t - x_i)\right]}{\sqrt{\Sigma_t \Lambda^{-1} + I}}, \tag{13}$$

where $i = 1,\ldots, N$ and $a = 1,\ldots, F$.

In practical systems, the physical limits of the control signal u should be considered. To account for the control limits coherently, the preliminary strategy π is squashed through a bounded and differentiable squashing function that limits the amplitude of the final strategy. Specifically, consider the third-order Fourier series expansion of a trapezoidal wave $\kappa(x) = [9\sin(x) + \sin(3x)]/8$, which is normalized to the interval $[-1, 1]$. Given the boundary conditions, the saturation function is defined as:

$$S(\pi_t) = u_{\min} + u_{\max} + u_{\max} \frac{9\sin\pi_t + \sin(3\pi_t)}{8}. \tag{14}$$

If the function is considered on the domain $[3\pi/2, 2\pi]$, the function is monotonically increasing, and the control signal u is squashed to the interval $[u_{\min} \;\; u_{\min} + u_{\max}]$ with $u_{\min} > 0$, $u_{\max} > 0$.

4. Learning process of the variable impedance control strategy

The learning process of the variable impedance control strategy consists of five main steps:

1) Learning the GP model that represents the system transition dynamics using the actual sampled data.
2) Inferring and predicting the long-term evolution of the states $p(x_1|\pi), \ldots, p(x_T|\pi)$ using the GP model, and evaluating the total expected cost $J^\pi(\theta)$ in T steps.
3) Calculating the gradients of the cost $dJ^\pi(\theta)/d\theta$ with respect to the strategy parameters and searching the optimal policy $\pi^* \leftarrow \pi(\theta)$ using the gradient-based policy search algorithm.
4) Calculating the variable impedance parameters and then applying the controller to execute force tracking task and saving the sampled data simultaneously.
5) Repeating steps 1–4 until the force tracking performance is satisfactory.

4.1. Probabilistic Gaussian process model. The unknown function that describes the system dynamics can be written as:

$$x_t = f(x_{t-1}, u_{t-1}), \tag{15}$$

$$y_t = x_t + \varepsilon_t, \tag{16}$$

with continuous state inputs $x \in \mathbb{R}^D$, control inputs $u \in \mathbb{R}^F$, training targets $y \in \mathbb{R}^E$, unknown transition dynamics f, and i.i.d. system noise $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$. In order to take the model uncertainties into account during prediction and planning, the algorithm does not make a certainty equivalence assumption on the learned model. Instead, it learns a probabilistic dynamics model and infers the posterior distribution over plausible function f from the noisy observations using GP.

A GP model [31] is a nonparametric probability model and can be described by a mean function $m(\cdot)$ and a positive semi-definite covariance function $k(\cdot,\cdot)$, which is also called a kernel. In this paper, for computation convenience, we consider a prior mean $m \equiv 0$ and the squared exponential kernel:

$$f(x) \sim \mathcal{GP}\big(m(x), k(x, x')\big), \tag{17}$$

$$k(x, x') = \alpha^2 \exp\left(-\frac{1}{2}(x - x')^T \Lambda^{-1} (x - x')\right) + \sigma_\varepsilon^2 I, \tag{18}$$

where α^2 is the variance of the latent function f, $\Lambda = diag([l_1^2, \ldots, l_D^2])$ depends on the characteristic length-scale l_i of each input dimension. Given N training inputs $X = [x_1, \ldots, x_n]$ and corresponding training targets $y = [y_1, \ldots, y_n]^T$, the GP hyper-parameters $[\Lambda \;\; \alpha^2 \;\; \sigma_\varepsilon^2]$ could be learned using evidence maximization algorithm.

Given a deterministic test input x_*, the posterior prediction $p(f_*|x_*)$ of the function value $f_* = f(x_*)$ is Gaussian distributed:

$$p(f_*|x_*) \sim \mathcal{N}(\mu_*, \Sigma_*), \tag{19}$$

$$\mu_* = m(x_*) + k(x_*, X)(K + \sigma_\varepsilon^2 I)^{-1}\big(y - m(X)\big) = \\ = m(x_*) + k(x_*, X)\beta, \tag{20}$$

$$\Sigma_* = k(x_*, x_*) - k(x_*, X)(K + \sigma_\varepsilon^2 I)^{-1} k(X, x_*), \tag{21}$$

where $\beta = (K + \sigma_\varepsilon^2 I)^{-1}\big(y - m(X)\big)$, and $K = k(X, X)$ is the kernel matrix.

Here, the function of the GP model we used is $f : \mathbb{R}^{D+F} \to \mathbb{R}^E$, $(x_{t-1}, u_{t-1}) \mapsto \Delta_t = x_t - x_{t-1} + \delta_t$, and $\hat{x}_{t-1} = (x_{t-1}, u_{t-1})$ is the training input tuples. Take the state increments as training targets $\Delta_t = x_t - x_{t-1} + \delta_t$, where $\delta_t \sim \mathcal{N}(0, \Sigma_\varepsilon)$ is the i.i.d. measurement noise. Since the state differences vary less than the absolute values, the underlying function that describes these differences varies less. Therefore, this implies that the learning process is easier and that less data are needed to find an accurate model. Moreover, when the predictions leave the training set, the prediction will remain constant.

4.2. Long-term planning through approximate inference. To derive the optimal control strategy $\pi : x \mapsto u = \pi(x, \theta)$, the controller parameters θ^* that minimize the total cost $J^\pi(\theta)$ need to be found according to the long-term predictions of states evolution. The states distributions $p(x_1), \ldots, p(x_T)$ could be obtained by cascading one-step predictions. The GP model can be used as a faithful transition dynamics of the real system, and it can map the Gaussian-distributed states space to targets space. The uncertainties of the inputs can pass through the model. Consequently, the uncertainties of the model are taken into account

in the long-term planning. In summary, the one-step prediction of the states can be described as follows:

$$p(x_{t-1}) \rightarrow p(u_{t-1}) \rightarrow p(x_{t-1}, u_{t-1}) \rightarrow p(\Delta_t) \rightarrow p(x_t). \quad (22)$$

If $p(x_1)$ is known, for predicting $p(x_t)$, we require a joint distribution $p(x_{t-1}, u_{t-1}) = p(\hat{x}_{t-1})$ based on the control signal $u_{t-1} = \pi(x_{t-1})$. We calculate the predictive control signal $p(u_{t-1})$ firstly and subsequently the cross-covariance $\mathrm{cov}[x_{t-1}, u_{t-1}]$. Then, $p(x_{t-1}, u_{t-1})$ is approximated by a Gaussian distribution:

$$p(\hat{x}_{t-1}) = p(x_{t-1}, u_{t-1}) = \mathcal{N}(\hat{\mu}_{t-1}, \hat{\Sigma}_{t-1}) = \\ = \mathcal{N}\left(\begin{bmatrix} \mu_{x_{t-1}} \\ \mu_{u_{t-1}} \end{bmatrix}, \begin{bmatrix} \Sigma_{x_{t-1}} & \Sigma_{x_{t-1}, u_{t-1}} \\ \Sigma^T_{x_{t-1}, u_{t-1}} & \Sigma_{u_{t-1}} \end{bmatrix}\right). \quad (23)$$

The distribution of the training targets Δ_t are predicted as follows:

$$p(\Delta_t) = \int p\big(f(\hat{x}_{t-1}) | \hat{x}_{t-1}\big) p(\hat{x}_{t-1}) d\hat{x}_{t-1}, \quad (24)$$

where the posterior predictive distribution of the transition dynamics $p\big(f(\hat{x}_{t-1}) | \hat{x}_{t-1}\big)$ could be calculated using the formulas (19–21). Using moment matching [14], $p(\Delta_t)$ could be approximated as a Gaussian distribution $\mathcal{N}(\mu_\Delta, \Sigma_\Delta)$. Then a Gaussian approximation to the desired state distribution $p(x_t)$ is given as follows:

$$p(x_t | \hat{\mu}_{t-1}, \hat{\Sigma}_{t-1}) \sim \mathcal{N}(\mu_t, \Sigma_t), \quad (25)$$

$$\mu_t = \mu_{t-1} + \mu_\Delta, \quad (26)$$

$$\Sigma_t = \Sigma_{t-1} + \Sigma_\Delta + \mathrm{cov}[x_{t-1}, \Delta_t] + \mathrm{cov}[\Delta_t, x_{t-1}], \quad (27)$$

$$\mathrm{cov}[x_{t-1}, \Delta_t] = \mathrm{cov}[x_{t-1}, u_{t-1}] \Sigma_u^{-1} \mathrm{cov}[u_{t-1}, \Delta_t]. \quad (28)$$

Because $u_{t-1} = \pi(x_{t-1})$ is a function of state x_{t-1} and $p(x_{t-1})$ is known, the calculation of $p(x_t)$ depends on the parameters θ of policy π.

4.3. Cost function. Task-specific regulation of impedance allows humans to learn a specific control strategy, combining the advantages of high stiffness and compliance: increase arm stiffness through muscle contraction to ensure accurate tracking or to suppress unknown perturbations; increase arm compliance through muscle relaxation to guarantee the security.

The cost function in RL usually penalizes the distance from the current state to the target state, without considering other prior knowledge. In order to make robots have the ability of compliance, the control gains should not be high. There are several desirable properties if the control gains are low, such as robustness and less wear and tear. Generally, high gains lead to high energy consumption. This is similar to the impedance regulation rules of humans which are compliant as much as possible and stiffen up only when the task requires it. In other words, increasing the impedance ensures tracking accuracy while decreasing impedance ensures safety; energy consumption should be reduced as much as possible. In this way, a tradeoff between the minimization of error and energy could be realized. To make the robots with these impedance characteristics and an enhanced ability of compliance, we add an item of energy consumption in the cost function. The suitable impedance gains are reduced by punishing the control actions. The instantaneous cost function is defined as follows:

$$c_t = c_b(x_t) + c_e(u_t), \quad (29)$$

$$c_b(x_t) = 1 - \exp\left(-\frac{1}{2\sigma_c^2} d\big(x_t, x_{t\,\mathrm{arg}\,et}\big)^2\right) \in [0, 1], \quad (30)$$

$$c_e(u_t) = c_e\big(\pi(x_t)\big) = \zeta \cdot (u_t / u_{\max})^2, \quad (31)$$

Here, $c_b(x_t)$ is the cost caused by the state error, denoted by a quadratic binary saturating function, which saturates at unity for large deviations to the desired target state. $d(\cdot)$ is the Euclidean distance and σ_c is the width of the cost function. $c_e(u_t)$ is the cost caused by the energy consumption (i.e. the mean squared energy penalty of impedance gains). ζ is the energy penalty coefficient. u_t is the current control signal, and $u_{\max}$ is the maximum control signal amplitude.

To evaluate the performance of the strategy π, we take the total expected cost $J^\pi(\theta)$ in T steps as the evaluation criteria:

$$J^\pi(\theta) = \sum_{t=0}^T \mathbb{E}_{x_t}[c(x_t)], x_0 \sim \mathcal{N}(\mu_0, \Sigma_0), \quad (32)$$

$$\mathbb{E}_{x_t}[c_t] = \int c_t \mathcal{N}(x_t | \mu_t, \Sigma_t) dx_t, \quad (33)$$

where $c(x_t)$ is the instantaneous cost at time t, and $\mathbb{E}_{x_t}[c(x_t)]$ is the expected values of the instantaneous cost with respect to the predictive state distributions.

4.4. Calculation of the cost gradients. The gradients of the expected cost $J^\pi(\theta)$ with respect to the control strategy parameters θ are given by:

$$\frac{dJ^\pi(\theta)}{d\theta} = \frac{d\sum_{t=0}^T \mathbb{E}_{x_t}[c(x_t)]}{d\theta} = \sum_{t=0}^T \frac{d\mathbb{E}_{x_t}[c(x_t)]}{d\theta}, \quad (34)$$

The expected immediate cost $\mathbb{E}_{x_t}[c(x_t)]$ requires averaging with respect to the state distribution $p(x_t) \sim \mathcal{N}(\mu_t, \Sigma_t)$, where μ_t and Σ_t are the mean and the covariance of $p(x_t)$, respectively. The derivative in (34) can be written as:

$$\frac{d\mathbb{E}_{x_t}[c(x_t)]}{d\theta} = \frac{d\mathbb{E}_{x_t}[c(x_t)]}{dp(x_t)} \frac{dp(x_t)}{d\theta} = \\ = \frac{\partial \mathbb{E}_{x_t}[c(x_t)]}{\partial \mu_t} \frac{d\mu_t}{d\theta} + \frac{\partial \mathbb{E}_{x_t}[c(x_t)]}{\partial \Sigma_t} \frac{d\Sigma_t}{d\theta}. \quad (35)$$

Given $c(x_t)$, the item $\partial \mathbb{E}_{x_t}[c(x_t)]/\partial \mu_t$ and $\partial \mathbb{E}_{x_t}[c(x_t)]/\partial \Sigma_t$ could be calculated analytically. Then we will focus on the calculation of $d\mu_t/d\theta$ and $d\Sigma_t/d\theta$. According to the computation sequence of the (22), we know that the predicted mean μ_t and the covariance Σ_t are functionally dependent on $p(x_{t-1}) \sim \mathcal{N}(\mu_{t-1}, \Sigma_{t-1})$ and the strategy parameters θ through u_{t-1}. We thus obtain:

$$\frac{d\mu_t}{d\theta} = \frac{\partial \mu_t}{\partial \mu_{t-1}} \frac{d\mu_{t-1}}{d\theta} + \frac{\partial \mu_t}{\partial \Sigma_{t-1}} \frac{d\Sigma_{t-1}}{d\theta} + \frac{\partial \mu_t}{\partial \theta}, \tag{36}$$

$$\frac{d\Sigma_t}{d\theta} = \frac{\partial \Sigma_t}{\partial \mu_{t-1}} \frac{d\mu_{t-1}}{d\theta} + \frac{\partial \Sigma_t}{\partial \Sigma_{t-1}} \frac{d\Sigma_{t-1}}{d\theta} + \frac{\partial \Sigma_t}{\partial \theta}, \tag{37}$$

$$\frac{\partial \mu_t}{\partial \theta} = \frac{\partial \mu_\Delta}{\partial p(u_{t-1})} \frac{\partial p(u_{t-1})}{\partial \theta} = \frac{\partial \mu_\Delta}{\partial \mu_u} \frac{\partial \mu_u}{\partial \theta} + \frac{\partial \mu_\Delta}{\partial \Sigma_u} \frac{\partial \Sigma_u}{\partial \theta}, \tag{38}$$

$$\frac{\partial \Sigma_t}{\partial \theta} = \frac{\partial \Sigma_\Delta}{\partial p(u_{t-1})} \frac{\partial p(u_{t-1})}{\partial \theta} = \frac{\partial \Sigma_\Delta}{\partial \mu_u} \frac{\partial \mu_u}{\partial \theta} + \frac{\partial \Sigma_\Delta}{\partial \Sigma_u} \frac{\partial \Sigma_u}{\partial \theta}. \tag{39}$$

By repeated application of the chain-rule, equations (35–39) can be computed analytically. We omit further lengthy details here and refer to [32] for more information. Analytic derivatives allow for standard gradient-based non-convex optimization methods, and fast convergence could be guaranteed. Here, the Conjugate Gradient (CG) method is employed to search the (sub)optimal controller parameters θ^* that minimize $J^\pi(\theta)$. Note that the learned suboptimal controller maybe not the optimal strategy, but it is a feasible strategy that could guarantee the satisfactory control performance.

5. Experiments and results

5.1. Experiment design. In the following, experiments on the Reinovo REBo-V-6R-650 industrial manipulator are implemented to verify the proposed learning variable impedance control method. Reinovo REBo-V-6R-650 is a 6-DoFs industrial manipulator with a six-axis Bioforcen force/torque (F/T) sensor mounted at the wrist. Due to the fact that the Reinovo industrial robot only provides a position control mode for users, the joint torques cannot be controlled and sampled directly. The Reinovo industrial robot is controlled by PC using TCP/IP protocol running at 100 Hz. The motion control of the robot is executed using Visual Studio. The learning algorithm is implemented by MATLAB. MATLAB communicates with Visual Studio using UDP protocol.

The six-axis F/T sensor is used to percept the contact force of the end-effector. The sensing range of the F/T sensor is ±625 N F_x, F_y, ±1250 N F_z, ±25 Nm T_x, T_y, ±12.5 Nm T_z. The total accuracy of the F/T sensor is less than 1% F.S. The sensor communicates with PC via Ethernet interface using TCP/IP protocol and samples the data at 5 kHz.

To simulate the nonlinear variation characteristics of the circumstance during the force tracking task, the combination of spring and rope is taken as the unstructured variable stiffness contact environment. The experimental setup mainly consists of a spring dynamometer attached to the tool at the end-effector and a rope of unknown length tied to the spring with the other end fixed on the table (see Fig. 2). Here, the rope is in a natural state of relaxation. The contact force is controlled by stretching the rope and the spring.

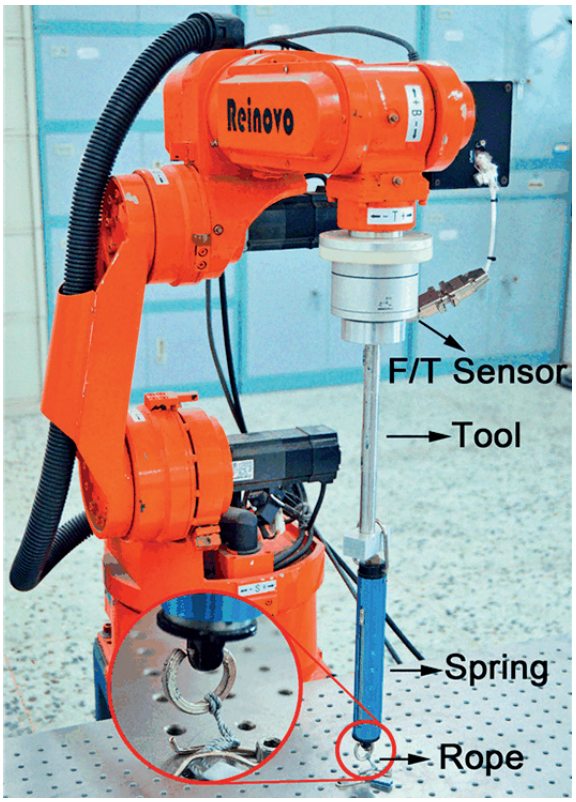

Fig. 2. Experimental setup using the Reinovo industrial robot

In the experiments, the episode length (i.e. the prediction horizon) is $T = 3$ s. The control period of the impedance controller is 0.01 s and the calculation period of the learning algorithm is 0.01 s. The number of total learning iterations, excluding the random initialization, is $N = 20$. The position and contact force of the end-effector $x = [X, Y, Z, F_x, F_y, F_z] \in \mathbb{R}^6$ are selected as the observed states. $u = [K_{dx}, K_{dy}, K_{dz}, B_{dx}, B_{dy}, B_{dz}] \in \mathbb{R}^6$ are the policy outputs (i.e. the control actions). The training target $y = [X_d, Y_d, Z_d, F_{dx}, F_{dy}, F_{dz}] = [0.41, 0, 0.2265, 0, 0, 15] \in \mathbb{R}^6$ is the desired position and the desired contact force of the end-effector, and $[0.41, 0, 0.2265]$ is the initial position of the end-effector in Cartesian space. The desired contact force in Z-axis is 15 N. If the steady state error of contact force $|F_z - F_{zd}| \leq 1$ N and the overshoot is less than 3 N, the task is successful; otherwise, it is failed. The number of the GP controller is $n = 10$. The ranges of the impedance parameters are set as $K_{dx,y,z} \in [0.1\ 25]$ and $B_{dx,y,z} \in [50\ 1000]$.

The energy penalty coefficient of the cost function is $\zeta = 0.03$. In the initial trial, the impedance parameters are initialized to stochastic variables which are subject to Gaussian-distribution $\mathcal{N}(u_0|0.7u_{\max}, u_{\max})$. Then a test is performed to acquire the initial states that are required for learning the GP model. The whole learning process is implemented automatically.

To verify that the proposed efficient learning variable impedance control method can be applied in different environments, two different spring dynamometers are used in the experiments. The exact values of the stiffness and damping of the springs are unknown to the system. The specifications of the two spring dynamometers used here are shown in Table 1.

Table 1
Specifications of the spring dynamometers

Spring	Range (kg)	Length (m)	Diameter (mm)
1	30	0.185	29
2	15	0.155	20

5.2. Experimental results. The experimental results of learning variable impedance control on the first spring are shown in Fig. 3. Figure 3a illustrates the overview of the learned cost curve. The abscissa is the learning iteration while the ordinate is the cumulative cost. The light blue dash-dotted line represents the cumulative cost during the policy search process. The red dotted line is the predicted cost mean according the control strategy. The red shade is the 95% confidence interval of the predicted cost. The block marks are the actual cumulative costs, which indicate whether the task is successful or not. The blue solid line is the actual cost curve. Figure 3b shows the learning process of variable impedance force control, where $N = 0$ is the result of the initial trial.

From the experimental results of the first spring, we can see that in the stochastic initialization trial (Fig. 3b $N = 0$), the manipulator moves slowly and the rope begins to be stretched to increase the contact force at $T = 2.5$ s while the contact force reaches 10.5 N at the end of the test, which implies that the task failed. After one learning

iteration, the updated GP controller adjusts the impedance control parameters depending on the current states. In the second trial (Fig. 3b $N = 1$), the rope and the spring can be stretched at $T = 1.5$ s, which is faster than that of the first trial, and the contact force reaches the desired values rapidly. The task failed because the overshoot is greater than 3 N. Using the historical saved data, the learning algorithm is further optimized. By adjusting the impedance control parameters dynamically, the rope is tightened quickly, while the overshoot is suppressed effectively. Only two learning iterations are needed to complete the task successfully. In order to reduce the total cumulative cost continuously, the explorations are carried out continuously to get a better control strategy. After 17 iterations (Fig. 3b $N = 17$), the cumulative cost is the smallest and the force control performance is also the best. The rope can be stretched at $T = 1$ s, and stable control of contact force is achieved by suppressing the overshoot.

Figure 4 shows the experimental results using the second spring. It can be seen that the results of the second spring are similar to that of the first scenario. Only six learning iterations are needed to learn the stable strategy to complete the task successfully. After eight learning iterations (Fig. 4b $N = 8$), the best strategy is learned. Using this strategy, the rope can be stretched at $T = 0.7$ s and the force is stabilized without overshoot.

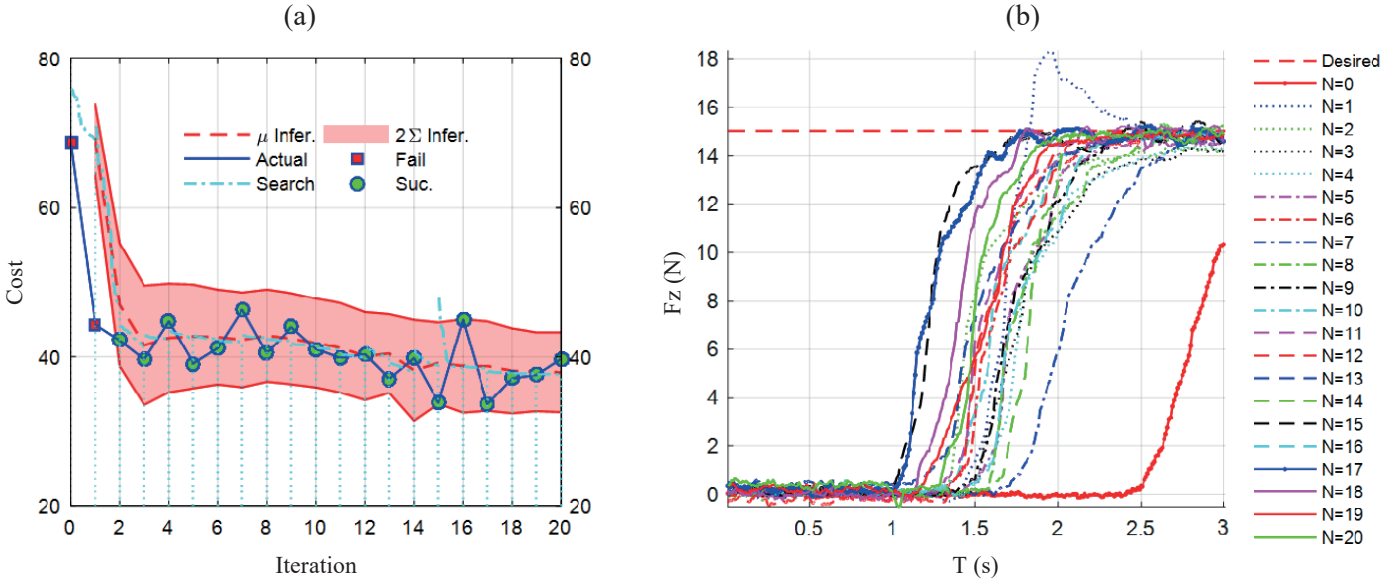

Fig. 3. Learning variable impedance control results of spring 1. a) Overview of the learned cost curve. b) Learning process of variable impedance force control

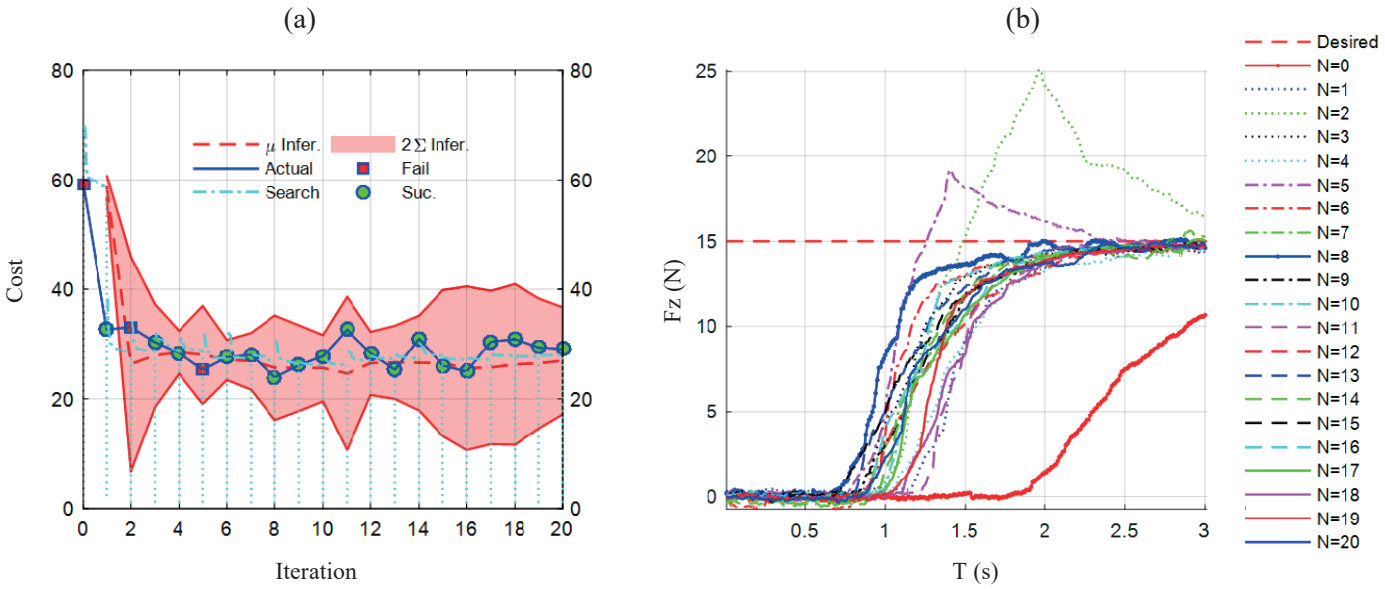

Fig. 4. Learning variable impedance control results of spring 2. a) Overview of the learned cost curve. b) Learning process of variable impedance force control

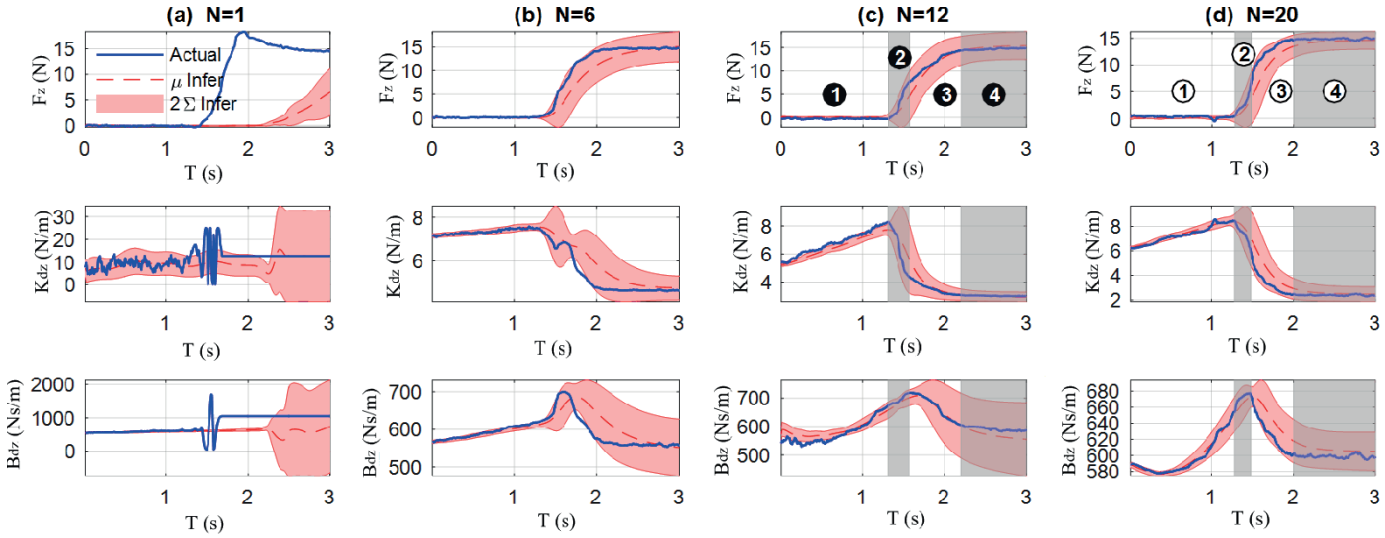

Fig. 5. States evolution and force control process using the first dynamometer

5.3. Results analysis of the learning process. Figure 5 shows the states evolution and force control process using the first dynamometer. The blue solid line is the actual state trajectory. The red dotted line and the red shade are the mean and the 95% confidence interval, respectively. The columns (a-d) in Fig. 5 are the states trajectories of the 1st, 6th, 12th, and 20th learning iteration, respectively. The top row is the change of contact force F_z, the second row is the profile of the target stiffness K_{dz}, and the third row is the profile of the target damping B_{dz}. Figure 6 shows the stretching process of the combination of the rope and the spring.

It can be seen from the results that in the early stage of learning, the uncertainties of the GP model are large due to the lack of collected data. With the increase of interaction time, the historical sampled data are constantly enriched, and the learned GP model, which can predict the states more and more

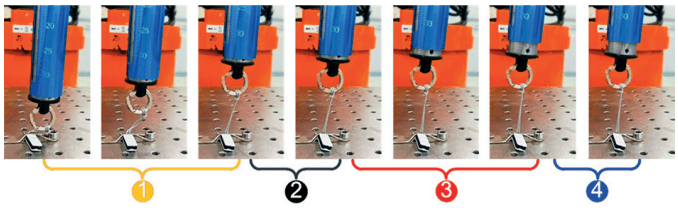

Fig. 6. Stretching process of the combination of the rope and the spring

accurately, is optimized and stabilized gradually. The control strategy is continually improved and regulates the profile of impedance control parameters to achieve better results.

Figure 7 shows the joint trajectories and Cartesian trajectories during the 20th experiment iteration. The trajectories of other iterations are similar to those of the 20th iteration. The joint positions and velocities are directly measured. The Cartesian positions and velocities of the end-effector are calculated using the inverse kinematics and Jacobian matrix, respectively. The shaded areas divide the process into four phases corresponding to Fig. 5d. Table 2 summarizes the key states of the four phases during the 20th iteration. The corresponded subscripts of F_z, K_{dz} and B_{dz} are shown in Fig. 5d while the subscripts of position (P) and velocity (V) are shown in Fig. 7c and d, respectively.

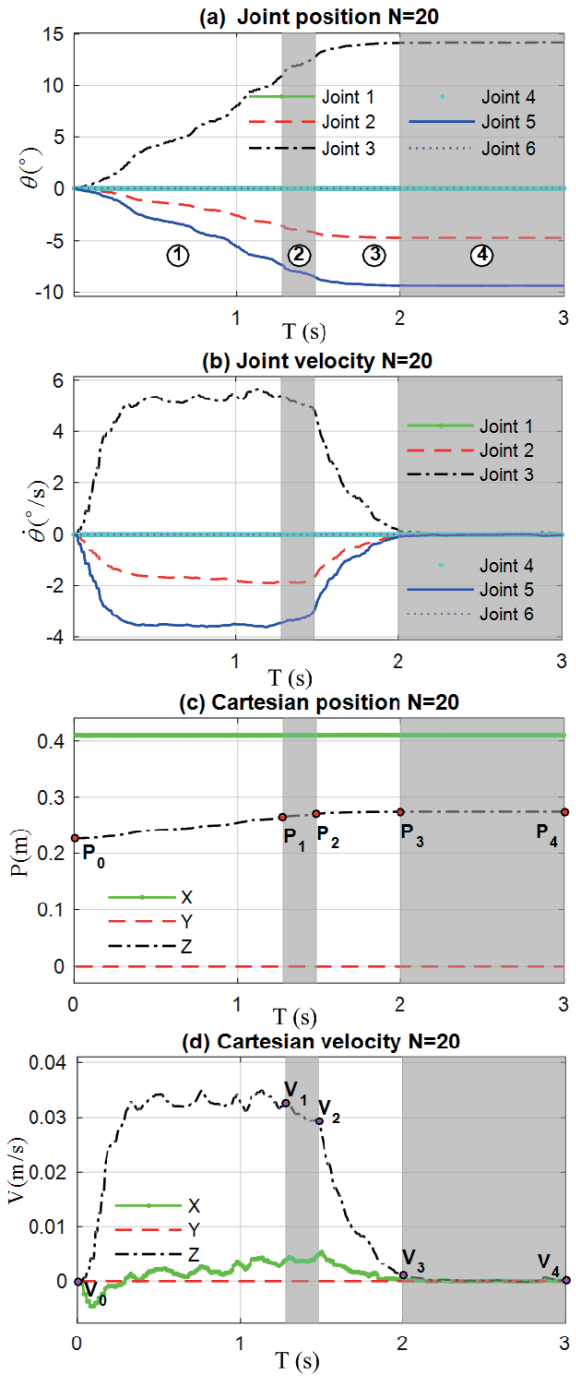

Fig. 7. Joint trajectories and Cartesian trajectories during the 20th experiment iteration

Table 2
Key states of the 4 phases during the 20th iteration

N	T (s)	P (m)	V (ms^{-1})	F_z (N)	K_{dz} (Nm^{-1})	B_{dz} (Nsm^{-1})
0	0.00	0.2265	0.000	0.39	6.24	588.9
1	1.28	0.2642	0.032	0.32	8.46	653.9
2	1.48	0.2701	0.029	6.09	5.88	673.8
3	2.00	0.2744	0.001	14.62	2.46	601.9
4	3.00	0.2745	0.000	14.87	2.37	598.5

Figure 7 shows the joint trajectories and Cartesian trajectories during the 20th experiment iteration. The trajectories of other iterations are similar to those of the 20th iteration. The joint positions and velocities are directly measured. The Cartesian positions and velocities of the end-effector are calculated using the inverse kinematics and Jacobian matrix, respectively. The shaded areas divide the process into four phases corresponding to Fig. 5d. Table 2 summarizes the key states of the four phases during the 20th iteration. The corresponded subscripts of F_z, K_{dz} and B_{dz} are shown in Fig. 5d while the subscripts of position (P) and velocity (V) are shown in Fig. 7c and d, respectively.

According to the definition of the cost function (29–31), we learn that small damping parameters will not only make the robot move quickly to contact with the environment to reduce the distance between x_t and x_{target}, but also reduce the cost caused by the energy consumption. Unfortunately, small damping parameters could reduce the positioning accuracy of the robot and thus make the system with poor ability to suppress disturbances. On the contrary, large damping parameters could improve the system's ability of suppressing disturbances and reduce the speed of motion. Hence, the learning algorithm must make a tradeoff between rapidity and stability to achieve the proper control strategy. Through continuous training and learning, the end-effector can contact with the environment more and more quickly, and the overshoot of the contact force is effectively suppressed.

As shown in Fig. 5–7 and Table 2, the process of regulating impedance parameters can be divided into four phases, which corresponds to the process of force control mentioned above:

1) $T_0 - T_1$: Phase before stretching the rope. The manipulator moves freely in the free space (Fig. 6.1). To tighten the rope quickly, the movement of the end-effector increases as the impedance parameters increase. The contact force is zero in this phase.
2) $T_1 - T_2$: Phase of stretching the rope. When the rope is stretched, the manipulator is suddenly converted from free

space motion to constrained space motion (Fig. 6.2). The stiffness of the contact environment increases suddenly, and this can be seen as a disturbance of environment. Consequently, the stiffness of the controller declines rapidly to make the system "soft" to ensure safety. Meanwhile, the damping continues to increase to make the system "stiff" to suppress the impact of environmental disturbance and avoid oscillation. On the whole, the system achieves an appropriate strategy by weighting "soft" and "stiff." In this phase, the contact force increases rapidly until the rope is tightened.

3) $T_2 - T_3$: Phase of stretching the spring. The spring begins to be stretched after the rope is tightened (Fig. 6.3). Although the environment changes suddenly, the controller dose not select the strategy as phase (2); it makes the system "soft" by gradually reducing the stiffness and damping to suppress the disturbances. In this way, the contact force increases slowly to avoid overshoot when approaching the desired value.
4) $T_3 - T_4$: Stable phase of stretching the spring. The manipulator contacts with the environment continuously and the contact force is stabilized to the desired value (Fig. 6.4). In this phase, the stiffness and damping of the controller are kept at minimum so that the system maintains the ability of compliance and the energy consumption could be reduced.

The impedance characteristics described above are similar to the strategy employed by humans for force tracking [33–35]. Reduce the impedance by muscle relaxation to make the system soft when it needs to guarantee safety, while increase the impedance by muscle contraction to makes the system stiff when it needs to guarantee fast-tracking or to suppress disturbances. When the contact environment is stable, the arm is kept in a compliant state by relaxation to achieve stable control.

There are total 20 learning iterations throughout the experiment. After six learning iterations, which means that only 18 seconds of interaction time is required, a sufficient dynamic model and controller can be learned to successfully complete the force tracking task. The experimental results above verify that the proposed bionic and efficient learning variable impedance control method is data-efficient, mainly because this method explicitly establishes the transition dynamics that are used for internal simulations and predictions of the system. In this way, more efficient information could be extracted from the sampled data.

5.4. Comparison with fixed impedance control. Variable impedance control can regulate the task-specific impedance parameters at different phases to complete the task more effectively. In this way, it can achieve a tradeoff between rapidity and stability, which is the characteristic that the impedance control with fixed parameters does not have. Using the first spring, a performance comparison between the proposed learning variable impedance control and the fixed parameter impedance control was conducted.

The comparison of force control performance is illustrated in Fig. 8. The magenta dashed line is the result of the fixed control parameter $K_d = 10$, $B_d = 450$, while the green point-solid line is the result of $K_d = 10$, $B_d = 400$. The red dotted line is the result of stochastic initialization, and the black dotted line is the result of the 20th learning iteration. The blue solid line is the result of the 17th learning iteration, which is the optimal strategy.

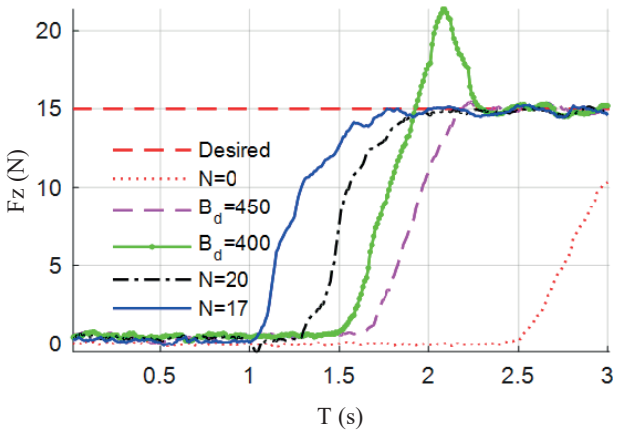

Fig. 8. Comparison between the learning variable impedance control and the fixed parameters impedance control

We can see from the results that when the fixed impedance parameters are $K_d = 10$, $B_d = 450$, the contact force began to increase at $T = 1.7$ s by stretching the rope without overshoot. Decreasing the target damping parameters could improve the response level of the manipulator to the contact force. When the damping is adjusted to $B_d = 400$, the rope is stretched at $T = 1.5$ s, but the large overshoot leads the task to failure. Using the strategy learned after 17 iterations, which is the optimal strategy, the rope is stretched at $T = 1$ s, and the contact force is controlled to reach the desired value without overshoot. It is impossible for impedance control with fixed parameters to achieve such performance.

In order to quantitatively compare the performances of learning variable impedance control and fixed impedance control, we use two indicators to quantify the performances. The first indicator is the accumulated cost $J^{\pi}(\theta)$ defined in (32). The second one is the root-mean-square error (RMSE) of the contact force:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{H} \left(F_z(t) - F_{zd}\right)^2}{H}}, \tag{40}$$

where $F_z(t)$ is the actual contact force in Z-axis, and F_{zd} is the desired contact force in Z-axis. H is the total number of the samples during the episode.

Table 3 reveals the performance comparison indicators. Here, $T(s)$ is the time that the contact force began to increase. For learning variable impedance control, the end-effector could contact with the environment quickly without force overshoot. The cost and the RMSE is smaller than those of fixed impedance control, which indicates that the proposed method is effective.

Table 3
Performance comparison between learning variable impedance control and fixed impedance control

Mode	Name	T (s)	Cost	RMSE	Overshoot
Fixed	$B_d = 450$	1.70	43.29	11.29	No
	$B_d = 400$	1.50	41.09	10.83	Yes
Variable	N = 20	1.25	39.71	10.19	No
	N = 17	1.00	33.64	9.23	No

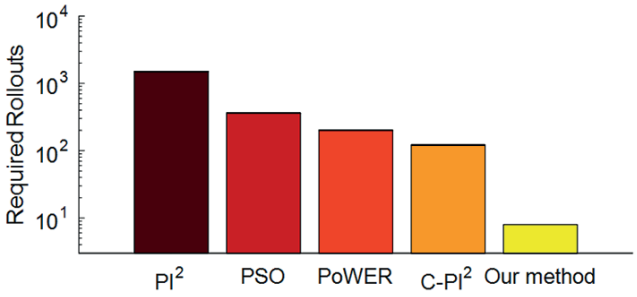

Fig. 9. Comparison of learning speed with other learning variable impedance control methods

6. Discussion

Learning variable impedance control methods can learn a satisfactory strategy through trial and error to complete force control task. Current learning variable impedance control methods usually require hundreds of rollouts to get a stable strategy. For tasks that are sensitive to the contact force, such as automatic assembly of explosive components, too many physical interactions with the environment during the learning process are often infeasible. Improving the learning method efficiency is critical. The required rollouts to get a satisfactory strategy could be used as an indicator of the learning speed.

Figure 9 shows the comparison of learning speed with other learning variable impedance control methods. From the results of [3, 28, 30, 36], we can see that, to get a stable strategy, PI^2 needs more than 1000 rollouts, whereas PSO requires 360 rollouts. The efficiency of PoWER is almost the same as that of C-PI^2, which requires 200 and 120 rollouts, respectively. The proposed method in this paper only requires fewer than 10 rollouts to obtain a satisfactory strategy that realizes fast control of contact force without overshoot. It outperforms other learning variable impedance control methods by at least one order of magnitude. The required interaction time is significantly reduced, which implies that the proposed method is more data-efficient.

7. Conclusion

In this paper, we propose an efficient learning variable impedance control method for the industrial robots to perform repetitive force-sensitive tasks. We have provided the key techniques to efficiently learn the impedance control strategy with minimal interaction time. This method was characterized by data-efficiency and no need for prior knowledge of the environment. To get the similar compliant ability of humans, we added an energy consumption item to the cost function to punish the actions. Furthermore, the probabilistic GP model was employed to pass the states uncertainties to permit long-term inference in a Bayesian manner, reducing the required interaction time and allowing for efficient policy updates. The model-based RL algorithm was used to search the optimal impedance regulation strategy, which simultaneously provides the continuous target stiffness and damping to the variable impedance control scheme. The experimental results on the Reinovo 6-DoFs industrial manipulator verify that the optimal impedance control strategy could be learned in only a few trials, and it outperforms other methods by at least one order of magnitude. In conclusion, our method could be applied to automatically learn to control the contact force efficiently.

Acknowledgements. This work is supported by the National Natural Science Foundation of China and the China Academy of Engineering Physics (NSAF, Grant No.U1530119).

References

[1] N. Hogan, "Impedance control: An approach to manipulation: Part i–theory", *Journal of Dynamic Systems, Measurement, and Control 107*, pp 1–7, 1985.
[2] C. Yang, G. Ganesh, S. Haddadin, S. Parusel, A. Albu-Schaeffer, and E. Burdet, "Human-like adaptation of force and impedance in stable and unstable interactions", *IEEE Transactions on Robotics 27*, pp 918–930, 2011.
[3] J.V.D. Kieboom and A.J. Ijspeert, "Exploiting natural dynamics in biped locomotion using variable impedance control", *IEEE-RAS International Conference on Humanoid Robots*, pp 348–353, 2013.
[4] C. Yang, J. Luo, Y. Pan, Z. Liu, and C.Y. Su, "Personalized variable gain control with tremor attenuation for robot teleoperation", *IEEE Transactions on Systems Man & Cybernetics Systems PP*, pp 1–12, 2017.
[5] G. Ganesh, N. Jarrassé, S. Haddadin, A. Albu-Schaeffer, and E. Burdet, "A versatile biomimetic controller for contact tooling and haptic exploration", *2012 IEEE International Conference on Robotics and Automation*, pp 3329–3334, 14–18 May 2012.
[6] F. Ferraguti, C. Secchi, and C. Fantuzzi, "A tank-based approach to impedance control with variable stiffness", *2013 IEEE International Conference on Robotics and Automation*, pp 4948–4953, 6–10 May 2013 2013.
[7] K. Lee and M. Buss, "Force tracking impedance control with variable target stiffness", *IFAC Proceedings Volumes 41*, pp 6751–6756, 2008.
[8] D. Braun, M. Howard, and S. Vijayakumar, "Optimal variable stiffness control: Formulation and application to explosive movement tasks", *Autonomous Robots 33*, pp 237–253, 2012.
[9] T. Tsuji and Y. Tanaka, "On-line learning of robot arm impedance using neural networks", *Robotics and Autonomous Systems 52*, pp 257–271, 2005.

[10] A.S. Polydoros and L. Nalpantidis, "Survey of model-based reinforcement learning: Applications on robotics", *Journal of Intelligent & Robotic Systems 86*, pp 153–173, 2017.
[11] K. Arulkumaran, M.P. Deisenroth, M. Brundage, and A.A. Bharath, "Deep reinforcement learning: a brief survey", *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp 26–38, 2017.
[12] M. Deisenroth, G. Neumann, and J. Peters, "A survey on policy search for robotics", *Journal of Intelligent & Robotic Systems 15*, pp 1–2, 2013.
[13] A. Nagabandi, G. Kahn, R.S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning", *arXiv preprint arXiv:1708.02596* 2017.
[14] M.P. Deisenroth and C.E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search", *International Conference on Machine Learning*, *ICML 2011*, pp 465–472, June 28 – July 2011.
[15] M.P. Deisenroth, D. Fox, and C.E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control", *IEEE Trans Pattern Anal Mach Intell 37*, pp 408–423, 2015.
[16] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, *et al.* "Human-level control through deep reinforcement learning", *Nature 518*, p 529, 2015.
[17] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.* "Mastering the game of go without human knowledge", *Nature 550*, p 354, 2017.
[18] M. Denil, P. Agrawal, T.D. Kulkarni, T. Erez, P. Battaglia, and N. de Freitas, "Learning to perform physics experiments via deep reinforcement learning", *arXiv preprint arXiv:1611.01843* 2016.
[19] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates", *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp 3389–3396, May 29 2017-June 3 2017 2017.
[20] V. Koropouli, S. Hirche, and D. Lee, "Generalization of force control policies from demonstrations for constrained robotic motion tasks", *Journal of Intelligent & Robotic Systems 80*, pp 1–16, 2015.
[21] D. Mitrovic, S. Klanke, M. Howard, and S. Vijayakumar, "Exploiting sensorimotor stochasticity for learning control of variable impedance actuators", *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pp 536–541, 6–8 Dec. 2010 2010.
[22] K. Kronander and A. Billard, "Learning compliant manipulation through kinesthetic and tactile human-robot interaction", *IEEE Trans Haptics 7*, pp 367–380, 2014.
[23] D. Mitrovic, S. Klanke, and S. Vijayakumar, "Learning impedance control of antagonistic systems based on stochastic optimization principles", *International Journal of Robotics Research 30*, pp 556–573, 2011.
[24] S.M. Khansari-Zadeh and O. Khatib, "Learning potential functions from human demonstrations with encapsulated dynamic and compliant behaviors", *Autonomous Robots 41*, pp 45–69, 2017.
[25] M. Li, H. Yin, K. Tahara, and A. Billard, "Learning object-level impedance control for robust grasping and dexterous manipulation", *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp 6784–6791, May 31–June 7 2014.
[26] Z. Du, W. Wang, Z. Yan, W. Dong, and W. Wang, "Variable admittance control based on fuzzy reinforcement learning for minimally invasive surgery manipulator", *Sensors 17*, 2017.
[27] J. Buchli, F. Stulp, E. Theodorou, and S. Schaal, "Learning variable impedance control", *International Journal of Robotics Research 30*, pp 820–833, 2011.
[28] F. Stulp, J. Buchli, A. Ellmer, M. Mistry, E.A. Theodorou, and S. Schaal, "Model-free reinforcement learning of impedance control in stochastic environments", *IEEE Transactions on Autonomous Mental Development 4*, pp 330–341, 2012.
[29] M. Kalakrishnan, L. Righetti, P. Pastor, and S. Schaal, "Learning force control policies for compliant manipulation", *Ieee/rsj International Conference on Intelligent Robots and Systems*, *IROS 2011*, pp 4639–4644, September 2011.
[30] F. Winter, M. Saveriano, and D. Lee, "The role of coupling terms in variable impedance policies learning", *International Workshop on Human-Friendly Robotics*, 2016.
[31] C.E. Rasmussen and C.K.I. Williams, "Gaussian processes for machine learning (adaptive computation and machine learning)", *MIT Press:* p 69–106, 2006.
[32] M. Deisenroth, "Efficient reinforcement learning using gaussian processes", *KIT Scientific Publishing:* 2010.
[33] C. Takahashi, R. Scheidt, and D. Reinkensmeyer, "Impedance control and internal model formation when reaching in a randomly varying dynamical environment", *Journal of Neurophysiology 86*, pp 1047–1051, 2001.
[34] D.W. Franklin, E. Burdet, K.P. Tee, R. Osu, C.-M. Chew, T.E. Milner, and M. Kawato, "Cns learns stable, accurate, and efficient movements using a simple algorithm", *J Neurosci 28*, pp 11165–11173, 2008.
[35] A. Babiarz, R. Bieda, K. Jaskot, and J. Klamka, "The dynamics of the human arm with an observer for the capture of body motion parameters", *Bull. Pol. Ac.: Tech. 61*, p 955, 2013.
[36] J. Kober and J. Peters, "Learning motor primitives for robotics", *2009 IEEE International Conference on Robotics and Automation*, pp 2112–2118, 12–17 May 2009 2009.