sensors

MDPI

Article

Efficient Force Control Learning System for Industrial Robots Based on Variable Impedance Control

Chao Li, Zhi Zhang *, Guihua Xia, Xinru Xie and Qidan Zhu

College of Automation, Harbin Engineering University, Harbin 150001, China; li_chao@hrbeu.edu.cn (C.L.); xiaguihua@hrbeu.edu.cn (G.X.); xiexinru@hrbeu.edu.cn (X.X.); zhuqidan@hrbeu.edu.cn (Q.Z.)

* Correspondence: zhangzhi1981@hrbeu.edu.cn; Tel.: +86-139-4614-2053

Received: 4 June 2018; Accepted: 26 July 2018; Published: 3 August 2018

Abstract: Learning variable impedance control is a powerful method to improve the performance of force control. However, current methods typically require too many interactions to achieve good performance. Data-inefficiency has limited these methods to learn force-sensitive tasks in real systems. In order to improve the sampling efficiency and decrease the required interactions during the learning process, this paper develops a data-efficient learning variable impedance control method that enables the industrial robots automatically learn to control the contact force in the unstructured environment. To this end, a Gaussian process model is learned as a faithful proxy of the system, which is then used to predict long-term state evolution for internal simulation, allowing for efficient strategy updates. The effects of model bias are reduced effectively by incorporating model uncertainty into long-term planning. Then the impedance profiles are regulated online according to the learned humanlike impedance strategy. In this way, the flexibility and adaptivity of the system could be enhanced. Both simulated and experimental tests have been performed on an industrial manipulator to verify the performance of the proposed method.

Keywords: force control; variable impedance control; efficient learning; Gaussian processes; industrial robot

1. Introduction

With the development of the modern robotics, compliance control is becoming an important component for industrial robots. Control of contact force is crucial for successfully executing operational tasks that involve physical contacts, such as grinding, deburring, or assembly. In the structured environment, good performances could be achieved using classical force control methods [1]. However, it is difficult to control the contact force effectively in the unstructured environment. Impedance control [2] provides a suitable control architecture for robots in both unconstrained and constrained motions by establishing a suitable mass-spring-damper system.

Neuroscience studies have demonstrated how humans perform specific tasks by adapting muscle stiffness [3]. Kieboom [4] studied the impedance regulation rule for bipedal locomotion and found that variable impedance control can improve gait quality and reduce energy expenditure. The ability to task-dependently change the impedance is one important aspect of biomechanical systems that leads to its good performance. Recently, many researchers have explored the benefits of varying the impedance during the task for robotics [5–9]. The basic idea is to adjust the impedance parameters according to the force feedback. Humanlike adaptivity was achieved in [9] by adapting force and impedance, providing an intuitive solution for human-robot interactions. Considering ensuring safe interaction, Calinon [6] proposed a learning-based control strategy with variable stiffness to reproduce the skill characteristics. Kronander [10] has demonstrated the stability of variable impedance control for the control system. Variable impedance not only enables control of the dynamic relationship between

contact forces and robot movements, but also enhances the flexibility of the control system. Generally, the methods of specifying the varying impedance can be classified into three categories.

(1) Optimal control. The basic idea is to dynamically adjust the gains by the feedback information using optimization techniques. They are usually robust to uncertain systems. Medina [11] proposed a variable compliance control approach based on risk-sensitive optimal feedback control. This approach has the benefits of high adaptability to uncertain and variable environment. The joint torque and the joint stiffness are independently and optimally modulated using the optimal variable stiffness control in [12]. Adaptive variable impedance control is proposed in [13], it stabilizes the system by adjusting the gains online according to the force feedback. To guarantee stable execution of variable impedance tasks, a tank-based approach to passive varying stiffness is proposed in [14].

(2) Imitation of human impedance. Humans have a perfect ability to complete a variety of interaction tasks in various environments by adapting their biomechanical impedance characteristics. These excellent abilities are developed over years of experience and stored in the central nervous system [15]. For the purpose of imitating the human impedance modulation manner, some methods have been proposed [6,8]. Toshi [16] discussed the impedance regulation law of the human hand during dynamic-contact tasks and proposed a bio-mimetic impedance control for robot. Lee [17] designed a variable stiffness control scheme imitating human control characteristics, and it achieved force tracking by adjusting the target stiffness without estimating the environment stiffness. Yang [18] introduced a coupling interface to naturally transfer human impedance adaptive skill to the robot by demonstration. Kronander [19] and Li [20] addressed the problem of compliance adjusting in a robot learning from demonstrations (RLfD), in which a robot could learn to adapt the stiffness based on human–robot interaction. Yang [21] proposed a framework for learning and generalizing humanlike variable impedance skills, combining the merits of the electromyographic (EMG) and dynamic movement primitives (DMP) model. To enhance the control performance, the problem of transferring human impedance behaviors to the robot has been studied in [22–24]. These methods usually use the EMG device to collect the muscle activities information, based on which variation of human limb impedance can be estimated.

(3) Reinforcement learning (RL). Reinforcement learning constitutes a significant aspect of the artificial intelligence field with numerous applications ranging from medicine to robotics [25,26]. Researchers have recently focused on learning an appropriate modulation strategy by means of RL to adjust the impedance characteristic of robot [7,27–29]. Du [30] proposed a variable admittance control method based on fuzzy RL for human–robot interaction. It improves the positioning accuracy and reduces the required energy by dynamically regulating the virtual damping. Li [31] proposed an BLF-based adaptive impedance control framework for a human–robot cooperation task. The impedance parameters were learned using the integral RL to get adaptive robot behaviors.

In summary, to derive an effective variable impedance controller, the first category and the second category of methods usually need advanced engineering knowledge about the robot and the task, as well as designing these parameters. Learning variable impedance control based on RL is a promising and powerful method, which can get the proper task-specific control strategy automatically through trial-and-error. RL algorithms can be broadly classified as two types: model-free and model-based. In model-free RL, policy is found without even building a model of the dynamics, and the policy parameters can be searched directly. However, for each sampled trajectory, it is necessary to interact with the robot, which is time-consuming and expensive. In model-based RL, the algorithm explicitly builds a transition model of the system, which is then used for internal simulations and predictions. The (local) optimal policy is improved based on the evaluations of these internal simulations. The model-based RL is more data-efficient than the model-free RL, but more computationally intensive [32,33]. In order to extend to high-dimensional tasks conveniently and avoid the model-bias of model-based RL algorithm, the existing learning variable impedance control methods are most commonly based on a model-free RL algorithm. Buchli [5] proposed a novel variable impedance control based on PI^2, which is a model-free RL algorithm. It realized simultaneous regulation of

motion trajectory and impedance gains using DMPs. This algorithm has been successfully extended to high-dimensional robotic tasks such as opening door, picking up pens [34], box flipping task [35] and sliding switch task for tendon-driven hand [36]. Stulp [29] further studied the applicability of PI^2 in stochastic force field, and it was able to find motor policies that qualitatively replicate human movement. Considering the coupling between degrees of freedom (DoFs), Winter [37] developed a C-PI^2 algorithm based on PI^2. Its learning speed was much higher than that of previous algorithms.

However, these methods usually require hundreds or thousands of rollouts to achieve satisfactory performance, which is unexpected for a force control system. None of these works address the issue of sampling efficiency. Improving data-efficiency is critical to learning to perform force-sensitive tasks, such as operational tasks of fragile components, because too many physical interactions with the environment during the learning process is usually infeasible. Alternatively, model-based RL is a promising way to improve the sampling efficiency. Fast convergence towards an optimal strategy could be guaranteed. Shadmehr [38] demonstrated that humans learn an internal model of the force field and compensate for external perturbations. Franklin [39] presented a model which combines three principles to learn stable, accurate, and efficient movements. It is able to accurately model empirical data gathered in force field experiments. Koropouli [27] investigated the generalization problem of force control policy. The force-motion mapping policy was learned from a set of demonstrated data to endow robots with certain human-like adroitness. Mitrovic [28] proposed to learn both the dynamics and the noise properties through supervised learning, using locally weighted projection regression. This model was then used in a model-based stochastic optimal controller to control a one-dimensional antagonistic actuator. It improved the accuracy performance significantly, but the analytical dynamic model still had accuracy limitations. These methods did not solve the key problem of model bias well.

Therefore, in order to improve the sampling efficiency, we propose a data-efficient learning variable impedance control method based on model-based RL that enables the industrial robots to automatically learn to control the contact force in the unstructured environment. A probabilistic Gaussian process (GP) model is approximated as a faithful proxy of the transition dynamics of the system. Then the probabilistic model is used for internal system simulation to improve the data-efficiency by predicting the long-term state evolution. This method reduces the effects of model-bias effectively by incorporating model uncertainty into long-term planning. The impedance profiles are regulated automatically according to the (sub)optimal impedance control strategy learned by the model-based RL algorithm to track the desired contact force. Additionally, we present a way of taking the penalty of control actions into account during planning to achieve the desired impedance characteristics. The performances of the system are verified through simulations and experiments on a six-DoF industrial manipulator. This system outperforms other learning variable impedance control methods by at least one order of magnitude in terms of learning speed.

The main contributions of this paper can be summarized as follows: (1) A data-efficient learning variable impedance control method is proposed to improve the sampling efficiency, which could significantly reduce the required physical interactions with environment during force control learning process. (2) The proposed method learns an impedance regulation strategy, based on which the impedance profiles are regulated online in a real-time manner to track the desired contact force. In this way, the flexibility and adaptivity of compliance control could be enhanced. (3) The impedance strategy with humanlike impedance characteristics is learned automatically through continuous explorations. There is no need to use additional sampling devices, such as EMG electrodes, to transfer human skills to the robot through demonstrations.

2. System Model and Contact Force Observer

2.1. Interaction Model

When the robot interacts with the rigid environment, the robot could be presented by a second order mass-spring-damper system, and the environment could be modeled as a spring-damping

model with stiffness k_e and damping b_e. The interaction model of the system is illustrated in Figure 1. Figure 1d shows a contact force diagram when robot comes in contact with the environment. m, b, and k denote the mass, damping, and stiffness of the robot end-effector, respectively. Let f be the contact force applied by the robot to the environment once a contact between both is established.

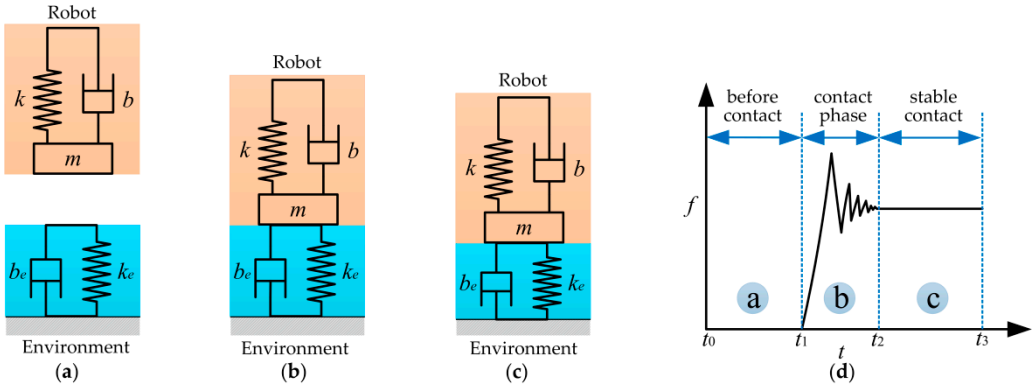

Figure 1. The interaction model of the system. (**a**) Without any contact between the robot and the environment; (**b**) critical point when contact occurs; (**c**) stable contact with the environment; (**d**) contact force diagram when robot comes in contact with the environment.

The contact process between the robot and the environment can be divided into three phases. In the first phase, the robot is approaching toward the environment (Figure 1a). There is no contact during this phase, and the contact force is zero (Figure 1d, $t_0 - t_1$). In the second phase, the robot is in contact with the environment (Figure 1b). During this phase, the contact force increases rapidly (Figure 1d, $t_1 - t_2$). Collision is inevitable, and the collision is transient and strongly nonlinear. In the third phase, the robot contacts with the environment continuously (Figure 1c) and the contact force is stabilized to the desired value (Figure 1d, $t_2 - t_3$).

High values of contact force are generally undesirable since they may stress both the robot and the manipulated object. Therefore, the overshoot and the oscillation caused by the inevitable collision should be suppressed effectively.

2.2. Contact Force Observer Based on Kalman Filter

The contact force is the quantity describing the state of interaction in the most complete fashion. To this end, the availability of force measurements is expected to provide enhanced performance for controlling interaction [1]. Recently, several methods have been proposed to make the force control possible without dedicated sensors, such as virtual force sensor [40] and contact force estimation methods based on motor signals [41,42]. However, to realize precise force control, industrial robots are typically equipped with F/T sensors at the wrist to measure the contact force. The measurements of force sensors do not correspond to the actual environmental interaction forces which usually contain inertial force and gravity. Moreover, the raw signals sampled from the force sensor may be corrupted by noise, especially in the industrial environment where equipped with large equipment. The electromagnetic noise, vibration noise, electrostatic effect, and thermal noise are very strong and complex. These disturbances seriously affect the measurement of the F/T sensor which will degrade the quality of force control. Hence, a contact force observer based on the Kalman filter is designed to estimate the actual contact force and moment applied at the contact point.

Figure 2 illustrates the contact force and moment applied at the end-effector. The center of mass of the end-effector locates at C. E is the contact point between the end-effector and the environment. The center of mass of the F/T sensor is S. As shown in Figure 2, the corresponding coordinate frames with the principal axes are denoted by Σ_C, Σ_E, and Σ_S, respectively. The world frame is denoted by Σ_W.

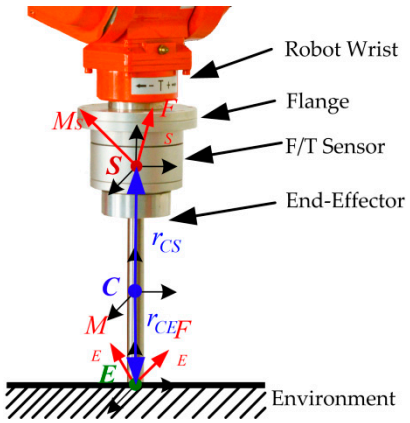

Figure 2. Contact force and moment applied at the end-effector.

Assume that the robot moves slowly during the tasks, the inertial effect could be negligible. The contact force model could be built using the Newton–Euler equations [43]

$$\begin{bmatrix} R_E^C & 0 \\ S(r_{CE}^C)R_E^C & R_E^C \end{bmatrix} \begin{bmatrix} F_E^E \\ M_E^E \end{bmatrix} = \begin{bmatrix} R_S^C & 0 \\ S(r_{CS}^C)R_S^C & R_S^C \end{bmatrix} \begin{bmatrix} F_S^S \\ M_S^S \end{bmatrix} - \begin{bmatrix} mR_W^C \\ 0 \end{bmatrix} g^W, \tag{1}$$

where the matrix R_j^i is the rotation matrix of frame Σ_j with respect to frame Σ_i. F_E^E and M_E^E are the actual contact forces and moments applied at the end-effector. F_S^S and M_S^S are the measured forces and moments by the F/T sensor. m is the mass of the end-effector. g^W is the gravitational acceleration. r_{CE}^C denotes the vector from C to E with respect to frame Σ_C. r_{CS}^C is the vector from C to S with respect to frame Σ_C. The skew-symmetry operator applied to vector b is donated as $S(b) = S_b$. To define the state vector x and the measurement vector y

$$x = \begin{bmatrix} F_S^S & M_S^S & \dot{F}_S^S & \dot{M}_S^S \end{bmatrix}^T, \tag{2}$$

$$y = \begin{bmatrix} F_E^E & M_E^E \end{bmatrix}^T, \tag{3}$$

According to (1), the system model and measurement model can be established

$$\begin{aligned} \dot{x}(t) &= A_0 x(t) + w_x \\ y(t) &= H_0 x(t) + D_0 g^W + v_y \end{aligned}, \tag{4}$$

where $A_0 = \begin{bmatrix} 0_{6\times6} & I_{6\times6} \\ 0_{6\times6} & 0_{6\times6} \end{bmatrix}$, $H_0 = \begin{bmatrix} R_S^E & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ R_C^E[S(r_{CS}^C) - S(r_{CE}^C)]R_S^C & R_S^E & 0_{3\times3} & 0_{3\times3} \end{bmatrix}$, $D_0 = m\begin{bmatrix} -R_W^E \\ R_C^E S(r_{CE}^C) R_W^C \end{bmatrix}$. ω_x and v_y are the process noise and measurement noise. They are assumed to be independent of each other with normal probability distribution $p(\omega_x) \sim \mathcal{N}(0, Q)$, $p(v_y) \sim \mathcal{N}(0, R)$. Q, R are the corresponding covariance matrices which are assumed as diagonal matrices with constant diagonal elements. The model (4) can be discretized resulting in the discrete-time linear system

$$\begin{aligned} x_k &= Ax_{k-1} + w_{k-1}, \\ y_k &= Hx_k + Dg^W + v_k, \end{aligned} \tag{5}$$

where $A = \begin{bmatrix} I_{6\times6} & I_{6\times6} \\ 0_{6\times6} & I_{6\times6} \end{bmatrix}$, $H = H_0$, $D = D_0$. The Kalman filter update consists of two steps:

1. Predict the state estimate $\hat{x}^-_{k|k-1}$ and the error covariance $P^-_{k|k-1}$ at time step k based on the results from the previous time step $k-1$:

$$\begin{aligned} \hat{x}^-{}_{k|k-1} &= A\hat{x}_{k-1}, \\ P^-_{k|k-1} &= AP_{k-1}A^T + Q. \end{aligned} \tag{6}$$

2. Correct the predictions $\hat{x}_k$ based on the measurements y_k:

$$\begin{aligned} \hat{x}_k &= \hat{x}^-_{k|k-1} + K_k(y_k - H\hat{x}^-_{k|k-1}), \\ K_k &= P^-_{k|k-1}H^T(HP^-_{k|k-1}H + R)^{-1}, \\ P_k &= (I - K_kH)P^-_{k|k-1}, \end{aligned} \tag{7}$$

where K_k is the Kalman gain, P_k is the updated error covariance. In practice, the covariance matrices Q and R can automatically be calibrated based on the offline experimental data [41]. It should be mentioned that the larger the weights of Q are chosen, the more the observer will rely on the measurements. Larger diagonal elements in Q result in faster response time of the corresponding estimates, but this also results in increased noise amplification. An implementation example of the contact force observer is shown in Figure 3.

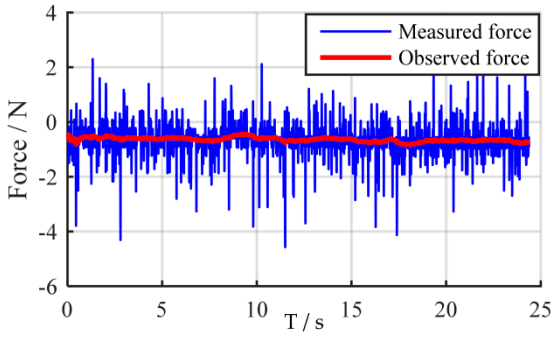

Figure 3. An implementation example of the contact force observer.

3. Position-Based Impedance Control for Force Control

One possible approach to achieve compliant behavior for robotics is the classical impedance control [2], which sometimes is also called force-based impedance control. In typical implementations, the controller has the positions as inputs and gives the motor torques as outputs. Force-based impedance control has been widely studied [44]. However, most commercial industrial robots emphasize the accuracy of trajectory following, and do not provide joint torque or motor current interfaces for users. Therefore, force-based impedance control is impossible on these robots.

Alternatively, another possible approach, which is typically suited for industrial robots, is the concept of admittance control [45], sometimes also called position-based impedance control. It maps from generalized forces to generalized positions. This control structure consists of an inner position control loop and an outer indirect force control loop. In constrained motion, the contact force measured by the F/T sensor modifies the desired trajectory in the outer impedance controller loop resulting in the compliant desired trajectory, which is to be tracked by the inner position controller. The position-based impedance control schematic for force control is shown in Figure 4. X_d, X_c, and X_m denote the reference position trajectory, the commanded position trajectory which is sent to the robot, and the measured position trajectory, respectively. Assuming good tracking performance of the inner position controller for slow motions, the commanded trajectory is equal to the measured trajectory, i.e., $X_m = X_c$.

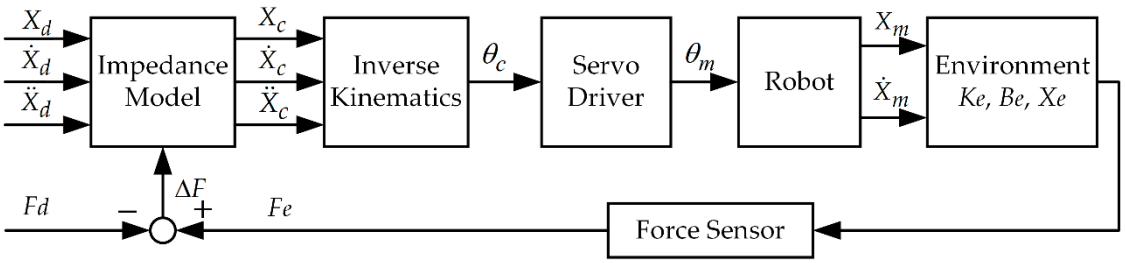

Figure 4. The position-based impedance control schematic.

Typically, the impedance model is chosen as a linear second order system

$$M_d(\ddot{X}_c - \ddot{X}_d) + B_d(\dot{X}_c - \dot{X}_d) + K_d(X_c - X_d) = F_e - F_d, \tag{8}$$

where M_d, B_d, and K_d are, respectively, the target inertial, damping, and stiffness matrices. F_d is the desired contact force. F_e is the actual contact force. The transfer-function of the impedance model is

$$H(s) = \frac{E(s)}{\Delta F(s)} = \frac{1}{M_d s^2 + B_d s + K_d}, \tag{9}$$

where $\Delta F = F_e - F_d$ denotes the force tracking error. $E = X_c - X_d$ is the desired position increment, and it is used to modify the reference position trajectory X_d to produce the commanded trajectory $X_c = X_d + E$, which is then tracked by the servo driver system.

To compute the desired position increment, discretize (9) using bilinear transformation

$$H(z) = H(s)\Big|_{s=\frac{2}{T}\frac{z-1}{z+1}} = \frac{T^2(z+1)^2}{\omega_1 z^2 + \omega_2 z + \omega_3}, \tag{10}$$

$$\begin{aligned} \omega_1 &= 4M_d + 2B_d T + K_d T^2, \\ \omega_2 &= -8M_d + 2K_d T^2, \\ \omega_3 &= 4M_d - 2B_d T + K_d T^2. \end{aligned} \tag{11}$$

Here, T is the control cycle. The desired position increment at time n is derived as

$$E(n) = \omega_1^{-1}\Big\{T^2[\Delta F(n) + 2\Delta F(n-1) + \Delta F(n-2)] - \omega_2 \delta X(n-1) - \omega_3 \delta X(n-2)\Big\}. \tag{12}$$

The environment is assumed to be a spring-damping model with stiffness K_e and damping B_e. X_e is the location of the environment. The contact force between the robot and the environment is then simplified as

$$F_e = B_e(\dot{X}_c - \dot{X}_e) + K_e(X_c - X_e). \tag{13}$$

Replacing X_d in (8) with the initial environment location X_e, and substituting (13) into (8), the new impedance equation is then converted to

$$M_d(\ddot{X}_c - \ddot{X}_e) + (B_d + B_e)(\dot{X}_c - \dot{X}_e) + (K_d + K_e)(X_c - X_e) = F_d. \tag{14}$$

Due to the fact that it is difficult to obtain accurate environment information. Therefore, if the environment stiffness K_e and damping B_e change, the dynamic characteristics of the system will change consequently. To guarantee the dynamic performance, the impedance parameters $[M_d, B_d, K_d]$ should be adjusted correspondingly.

4. Data-Efficient Learning Variable Impedance Control with Model-Based RL

Generally, too many physical interactions with the environment are infeasible for learning to execute force-sensitive tasks. In order to reduce the required interactions with the environment during force control learning process, a learning variable impedance control approach with model-based RL algorithm is proposed in the following to learn the impedance regulation strategy.

4.1. Scheme of the Data-Efficient Learning Variable Impedance Control

The scheme of the method is illustrated in Figure 5. F_d is the desired value of contact force, F is the actual contact force estimated by the force observer, and ΔF is the force tracking error. The desired position increment of the end-effector E is calculated using the variable impedance controller. X_d is the desired reference trajectory. The desired joints positions q_d are calculated by adopting inverse kinematics according to the commanded trajectory X_c. The actual Cartesian position of the end-effector X could be achieved using the measured joints positions q by means of forward kinematics. The joints are controlled by the joint motion controller of the industrial robot. K_E and B_E denote the unknown stiffness and damping of the environment, respectively.

The transition dynamics of the system is approximated by the GP model which is trained using the collected data. Then the learning algorithm is used to learn the (sub)optimal impedance control strategy π while predicting the system evolution using the GP model. Instead of the motion trajectory, the proposed method learns an impedance regulation strategy, based on which the impedance profiles are regulated online in a real-time manner to control the contact force. In this way, the dynamic relationship between contact force and robot movement could be controlled in a continuous manner. Moreover, the flexibility and adaptivity for compliance control could be enhanced. Positive definite impedance parameters could ensure the asymptotic stability of the original desired dynamics, but it is not recommended to modify the target inertia matrix because it is easy to cause the system to instability [45]. 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 parameters that should be tuned in variable impedance control. Based on the learned impedance strategy, the impedance parameters $u = [K_d\ B_d]$ are calculated according to the states of the contact force and the position of the end-effector, which are then transferred to the variable impedance controller.

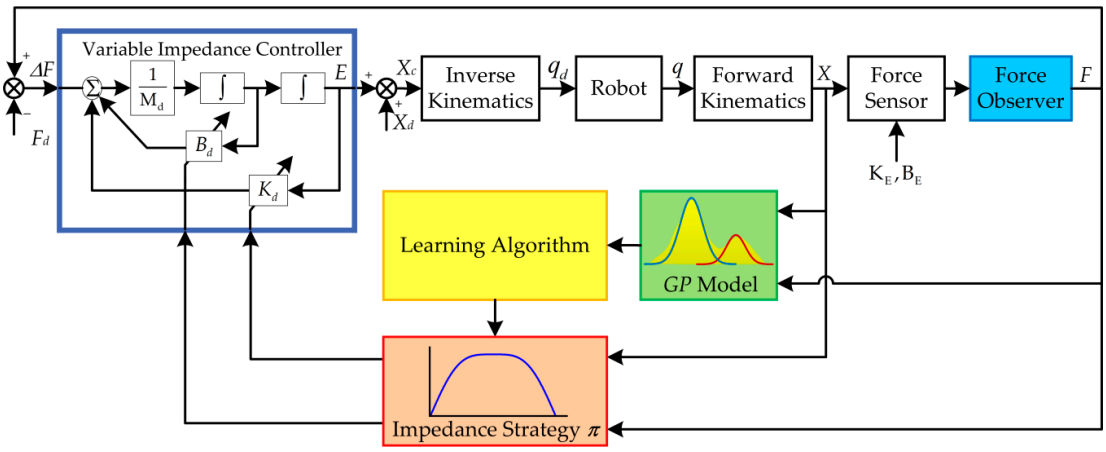

Figure 5. Scheme of the data-efficient learning variable impedance control.

The learning process of variable impedance strategy consists of seven main steps:

1. Initializing the strategy parameters stochastically, applying the random impedance parameters to the system and recording the sampled data.
2. Training the system transition dynamics, i.e., the GP model, using all historical data.

3. Inferring and predicting the long-term evolution of the states according to the GP model.
4. Evaluating the total expected cost $J^{\pi}(\theta)$ in T steps, and calculating the gradients of the cost $dJ^{\pi}(\theta)/d\theta$ with respect to the strategy parameters θ.
5. Learning the optimal strategy $\pi^* \leftarrow \pi(\theta)$ using the gradient-based policy search algorithm.
6. Applying the impedance strategy to the system, then executing a force tracking task using the learned variable impedance strategy and recording the sampled data simultaneously.
7. Repeating steps (2)–(6) until the performance of force control is satisfactory.

4.2. Variable Impedance Strategy

The impedance control strategy is defined as $\pi : x \mapsto u = \pi(x, \theta)$, where the inputs of the strategy are the observed states of the robot $x = [X\ F] \in \mathbb{R}^D$, the outputs of the 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 strategy parameters that to be learned. Here, the GP controller is chosen as the control strategy

$$\pi_t = \pi(x_t, \theta) = \sum_{i=1}^{n} \beta_{\pi,i} k(x_\pi, x_t) = \beta_\pi^T K(X_\pi, x_t), \tag{15}$$

$$\beta_\pi = \left(K_\pi(X_\pi, X_\pi) + \sigma_{\varepsilon,\pi}^2 I\right)^{-1} y_\pi, \tag{16}$$

$$k(x_\pi, x_t) = \sigma_{f,\pi}^2 \exp\left(-\frac{1}{2}(x_{\pi i} - x_t)^T \Lambda^{-1}(x_{\pi i} - x_t)\right), \tag{17}$$

where x_t is the test input. $X_\pi = [x_{\pi 1}, \ldots, x_{\pi n}]$ 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. K is the covariance matrix with entries $K_{ij} = k(x_i, x_j)$. $\Lambda = diag\left(l_1^2, \ldots, l_D^2\right)$ is the length-scale matrix where l_i is the characteristic length-scale of each input dimension, $\sigma_{f,\pi}^2$ is the signal variance, which is fixed to one here, $\sigma_{\varepsilon,\pi}^2$ is the measurement noise variance, and $\theta = \left[X_\pi, y_\pi, l_1, \ldots, l_D, \sigma_{f,\pi}^2, \sigma_{\varepsilon,\pi}^2\right]$ 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 impedance strategy π and the states x_t. The relationship between the impedance parameters and the control strategy can be written as

$$\begin{bmatrix} K_d & B_d \end{bmatrix} = u = \pi(x_t, \theta) = \beta_\pi^T K(X_\pi, x_t). \tag{18}$$

In practical systems, the physical limits of the impedance parameters should be considered. The preliminary strategy π should be squashed coherently through a bounded and differentiable saturation function. The saturation function has to be on a finite interval, such that a maximum and minimum are obtained for finite function inputs. Furthermore, the function should be monotonically increasing. The derivative and second derivative of the saturation function have to be zero at the boundary points to require stationary points at these boundaries. 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

$$S(\pi_t) = u_{\min} + u_{\max} + u_{\max}\frac{9\sin\pi_t + \sin(3\pi_t)}{8}. \tag{19}$$

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 $[\mathrm{u}_{\min}\ u_{\min} + u_{\max}]$.

4.3. Probabilistic Gaussian Process Model

Transition models have a large impact on the performance of model-based RL, since the learned strategy inherently relies on the quality of the learned forward model, which essentially serves as a simulator of the system. The transition models that have been employed for model-based RL can be classified into two main categories [25]: the deterministic models and the stochastic models. Despite the intensive computation, the state-of-the-art approach for learning the transition models is the GP model [46], because it is capable of modeling a wide spread of nonlinear systems by explicitly incorporating model uncertainty into long-term planning, which is a key problem in model-based learning. In addition, the GP model shows good convergence properties which are necessary for implementation of the algorithm. A GP model can be thought as a probabilistic distribution over possible functions, and it is completely specified by a mean function $m(\cdot)$ and a positive semi-definite covariance function $k(\cdot,\cdot)$, also called a kernel.

Here, we consider the unknown function that describes the system dynamics

$$\begin{aligned} x_t &= f(x_{t-1}, u_{t-1}), \\ y_t &= x_t + \varepsilon_t, \end{aligned} \tag{20}$$

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 proposed approach does not make a certainty equivalence assumption on the learned model. Instead, it learns a probabilistic GP model and infers the posterior distribution over plausible function f from the observations. For computation convenience, we consider a prior mean $m \equiv 0$ and the squared exponential kernel

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')), \tag{21}$$

$$k(x, x') = \alpha^2 \exp\left(-\frac{1}{2}(x - x')^T \Lambda^{-1}(x - x')\right) + \sigma_\varepsilon^2 I, \tag{22}$$

where α^2 is the variance of the latent function f, the weighting matrix $\Lambda = diag([l_1^2, \ldots l_D^2])$ depends on the different 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 $\left[\Lambda\ \alpha^2\ \sigma_\varepsilon^2\right]$ could be learned using evidence maximization algorithm [46].

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

$$p(f_*|x_*) \sim \mathcal{N}(\mu_*, \sum_*), \tag{23}$$

$$\mu_* = m(x_*) + k(x_*, X)(K + \sigma_\varepsilon^2 I)^{-1}(y - m(X)) = m(x_*) + k(x_*, X)\beta, \tag{24}$$

$$\sum_* = k(x_*, x_*) - k(x_*, X)(K + \sigma_\varepsilon^2 I)^{-1} k(X, x_*), \tag{25}$$

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

In our force control learning system, the function of the GP model is defined as $f : \mathbb{R}^{D+F} \to \mathbb{R}^E, (x_{t-1}, u_{t-1}) \longmapsto \Delta_t = x_t - x_{t-1} + \delta_t$, where $\hat{x}_{t-1} = (x_{t-1}, u_{t-1})$ is the training input tuples. Take the state increments $\Delta_t = x_t - x_{t-1} + \delta_t$ as training targets, where $\delta_t \sim \mathcal{N}(0, \Sigma_\varepsilon)$ is 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, it implies that the learning process is easier and that less data is needed to find an accurate model. Moreover, when the predictions leave the training set, the prediction will not fall back to zero but remain constant.

4.4. Approximate Prediction for Strategy Evaluation

For the sake of reducing the required physical interactions with the robots while getting an effective control strategy, the effective utilization of sampled data must be increased. To this end, the learned probabilistic GP model is used as the faithfully dynamics of the system, which is then used for internal simulations and predictions about how the real system would behave. The (sub)optimal strategy is improved based on the evaluations of these internal virtual trials. Thus, the data-efficiency is improved.

To evaluate the strategy, the long-term predictions of state $p(x_1), \ldots, p(x_T)$ should be computed iteratively from the initial state distribution $p(x_0)$ by cascading one-step predictions [47]. Since the GP model can map the Gaussian-distributed states space to the targets space, the uncertainties of the inputs can pass through the model, and the uncertainties of the model are taken into account in the long-term planning. A conceptual illustration of long-term predictions of state evolution [48] is shown in Figure 6.

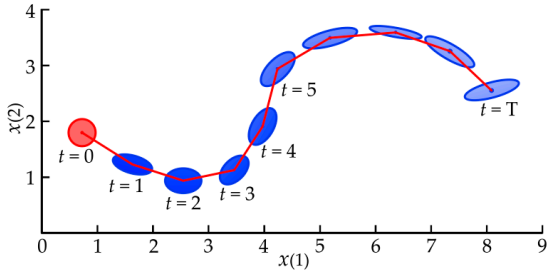

Figure 6. A conceptual illustration of long-term predictions of state evolution.

The one-step prediction of the states can be summarized as

$$p(x_{t-1}) \rightarrow p(u_{t-1}) \rightarrow p(x_{t-1}, u_{t-1}) \rightarrow p(\Delta_t) \rightarrow p(x_t). \tag{26}$$

As $u_{t-1} = \pi(x_{t-1})$ is a function of state x_{t-1} and $p(x_{t-1})$ is known, the calculation of $p(x_t)$ requires a joint distribution $p(\hat{x}_{t-1}) = p(x_{t-1}, u_{t-1})$. First, we calculate the predictive control signal $p(u_{t-1})$ 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 [47]

$$p(\hat{x}_{t-1}) = p(x_{t-1}, u_{t-1}) = \mathcal{N}(\hat{\mu}_{t-1}, \hat{\sum}_{t-1}) = \mathcal{N}\left(\begin{bmatrix} \mu_{x_{t-1}} \\ \mu_{u_{t-1}} \end{bmatrix}, \begin{bmatrix} \sum_{x_{t-1}} & \sum_{x_{t-1}, u_{t-1}} \\ \sum_{x_{t-1}, u_{t-1}}^{T} & \sum_{u_{t-1}} \end{bmatrix} \right). \tag{27}$$

The distribution of the training targets Δ_t are predicted as

$$p(\Delta_t) = \int p(f(\hat{x}_{t-1}) | \hat{x}_{t-1}) p(\hat{x}_{t-1}) d\hat{x}_{t-1}, \tag{28}$$

where the posterior predictive distribution of the transition dynamics $p(f(\hat{x}_{t-1}) | \hat{x}_{t-1})$ could be calculated using the Formulas (23)–(25). Using moment matching [49], $p(\Delta_t)$ 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

$$p(x_t | \hat{\mu}_{t-1}, \hat{\sum}_{t-1}) \sim \mathcal{N}(\mu_t, \sum_t), \tag{29}$$

$$\mu_t = \mu_{t-1} + \mu_\Delta, \tag{30}$$

$$\sum_t = \sum_{t-1} + \sum_\Delta + \mathrm{cov}[x_{t-1}, \Delta_t] + \mathrm{cov}[\Delta_t, x_{t-1}], \tag{31}$$

$$\mathrm{cov}[x_{t-1}, \Delta_t] = \mathrm{cov}[x_{t-1}, u_{t-1}] \sum_u^{-1} \mathrm{cov}[u_{t-1}, \Delta_t]. \tag{32}$$

4.5. Gradient-Based Strategy Learning

The goal of the learning algorithm is to find the strategy parameters that minimize the total expected costs $\theta^* = \arg\min J^\pi(\theta)$. The search direction can be selected using the gradient information. The total expected cost $J^\pi(\theta)$ in T steps is calculated according to the state evolution

$$J^\pi(\theta) = \sum_{t=0}^{T} \mathbb{E}[c(x_t)], x_0 \sim \mathcal{N}(\mu_0, \sum_0), \tag{33}$$

$$\mathbb{E}[c_t] = \int c_t \mathcal{N}(x_t|\mu_t, \sum_t) \, dx_t. \tag{34}$$

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

The cost function in RL usually penalizes the Euclidean distance from the current state to the target state, without considering other prior knowledge. However, in order to make robots with the ability of compliance, the control gains should not be high for practical interaction tasks. Generally, high gains will result in instability in stiff contact situations due to the inherent manipulator compliance, especially for an admittance-type force controller. In addition, low gains lead to several desirable properties of the system, such as compliant behavior (safety and/or robustness), lowered energy consumption, and less wear and tear. This is similar to the impedance regulation rules of humans. Humans learn a task-specific impedance regulation strategy that combines the advantages of high stiffness and compliance. The general rule of thumb thus seems to be "be compliant when possible; stiffen up only when the task requires it". In other words, impedance increasing ensures tracking accuracy while impedance decreasing ensures safety.

To make the robots with these impedance characteristics, we present a way of taking the penalty of control actions into account during planning. The instantaneous cost function is defined

$$c_t = c_b(x_t) + c_e(u_t), \tag{35}$$

$$c_b(x_t) = 1 - \exp(-\frac{1}{2\sigma_c^2} d(x_t, x_{\text{target}})^2) \in [0, 1], \tag{36}$$

$$c_e(u_t) = c_e(\pi(x_t)) = \zeta \cdot (u_t/u_{\max})^2. \tag{37}$$

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 between the current state x_t to the target state x_{target} and σ_c is the width of the cost function. $c_e(u_t)$ is the cost caused by the control actions, i.e., the mean squared penalty of impedance gains. The suitable impedance gains could be reduced by punishing the control actions. ζ is the action penalty coefficient. u_t is the current control signal, and u_{max} is the maximum control signal amplitude.

The gradients of $J^\pi(\theta)$ with respect to the strategy parameters θ are given by

$$\frac{dJ^\pi(\theta)}{d\theta} = \frac{d\sum_{t=0}^{T} \mathbb{E}[c(x_t)]}{d\theta} = \sum_{t=0}^{T} \frac{d\mathbb{E}[c(x_t)]}{d\theta}. \tag{38}$$

The expected immediate cost $\mathbb{E}[c(x_t)]$ 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 Equation (38) can be written as

$$\frac{d\mathbb{E}[c(x_t)]}{d\theta} = \frac{d\mathbb{E}[c(x_t)]}{dp(x_t)} \frac{dp(x_t)}{d\theta} = \frac{\partial\mathbb{E}[c(x_t)]}{\partial\mu_t} \frac{d\mu_t}{d\theta} + \frac{\partial\mathbb{E}[c(x_t)]}{\partial\sum_t} \frac{d\sum_t}{d\theta}. \tag{39}$$

Given $c(x_t)$, the item $\partial\mathbb{E}[c(x_t)]/\partial\mu_t$ and $\partial\mathbb{E}[c(x_t)]/\partial\Sigma_t$ could be calculated analytically. Then we will focus on the calculation of $\mathrm{d}\mu_t/d\theta$ and $\mathrm{d}\Sigma_t/d\theta$. Due to the computation sequence

of (26), 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 μ_{t-1}. We thus obtain

$$\frac{d\mu_t}{d\theta} = \frac{\partial \mu_t}{\partial p(x_{t-1})} \frac{dp(x_{t-1})}{d\theta} + \frac{\partial \mu_t}{\partial \theta} = \frac{\partial \mu_t}{\partial \mu_{t-1}} \frac{d\mu_{t-1}}{d\theta} + \frac{\partial \mu_t}{\partial \sum_{t-1}} \frac{d\sum_{t-1}}{d\theta} + \frac{\partial \mu_t}{\partial \theta}, \tag{40}$$

$$\frac{d\sum_t}{d\theta} = \frac{\partial \sum_t}{\partial p(x_{t-1})} \frac{dp(x_{t-1})}{d\theta} + \frac{\partial \sum_t}{\partial \theta} = \frac{\partial \sum_t}{\partial \mu_{t-1}} \frac{d\mu_{t-1}}{d\theta} + \frac{\partial \sum_t}{\partial \sum_{t-1}} \frac{d\sum_{t-1}}{d\theta} + \frac{\partial \sum_t}{\partial \theta}, \tag{41}$$

$$\frac{\partial \mu_t}{\partial \theta} = \frac{\partial \mu_\Delta}{\partial p(u_{t-1})} \frac{\partial p(u_{t-1})}{\partial \theta} = \frac{\partial \mu_\Delta}{\partial \mu_u} \frac{\partial \mu_u}{\partial \theta} + \frac{\partial \mu_\Delta}{\partial \sum_u} \frac{\partial \sum_u}{\partial \theta}, \tag{42}$$

$$\frac{\partial \sum_t}{\partial \theta} = \frac{\partial \sum_\Delta}{\partial p(u_{t-1})} \frac{\partial p(u_{t-1})}{\partial \theta} = \frac{\partial \sum_\Delta}{\partial \mu_u} \frac{\partial \mu_u}{\partial \theta} + \frac{\partial \sum_\Delta}{\partial \sum_u} \frac{\partial \sum_u}{\partial \theta}. \tag{43}$$

By repeated application of the chain-rule, the Equations (39)–(43) can be computed analytically. We omit further lengthy details here and refer to [47] for more information. Then the non-convex gradient-based optimization algorithm—e.g., conjugate gradient—can be applied to find the strategy parameters θ^* that minimize $J^\pi(\theta)$.

5. Simulations and Experiments

To verify the proposed force control learning system based on variable impedance control, a series of simulation and experiment studies on the Reinovo REBot-V-6R-650 industrial robot (Shenzhen Reinovo Technology CO., LTD, Shenzhen, China) are conducted and presented in this section. Reinovo REBot-V-6R-650 is a six-DoF industrial manipulator with a six-axis Bioforcen F/T sensor (Anhui Biofrcen Intelligent Technology CO., LTD, Hefei, China) mounted at the wrist. The 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, and ± 12.5 Nm T_z with the total accuracy less than $\leq 1\%$ F.S.

5.1. Simulation Study

We first evaluated our system through a simulation of force control using MATLAB (R2015b Version 8.6) Simulink. The block diagram of simulation is shown in Figure 7. In the simulation setup, a stiff contact plane is placed under the end-effector of the robot. The stiffness and damping of the plane are set 5000 N/m and 1 Ns/m, respectively. The robot's base is located at $[0,\ 0,\ 0]$ m while the original position of the plane O_P is located at $[0.2, -0.5, 0.15]$ m. The plane's length, width, and height are 0.9, 1, and 0.2 m, respectively. The robot should automatically learn to control the contact force to the desired value.

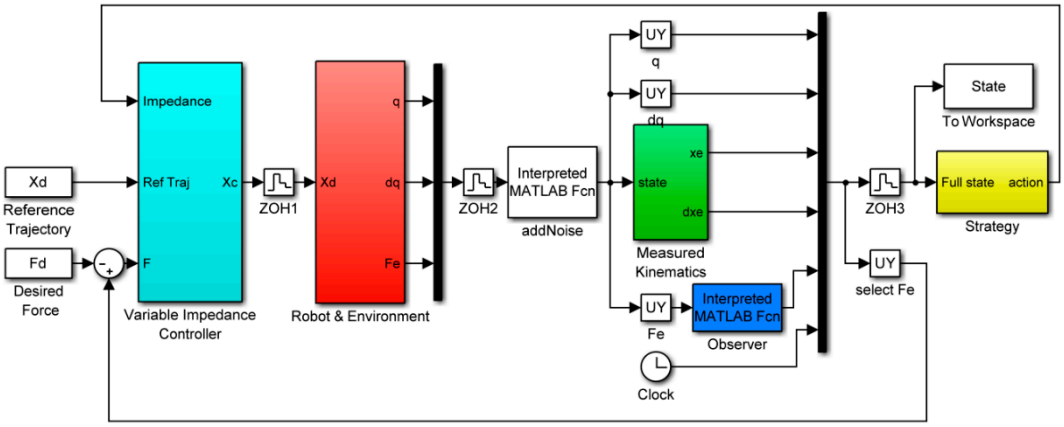

Figure 7. The block diagram of simulation in MATLAB Simulink.

In the simulation, the episode length is set as $T = 1$ 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 training inputs of the learning algorithm are the position and contact force of the end-effector $x = \left[X, Y, Z, F_x, F_y, F_z\right] \in \mathbb{R}^6$. The training targets of the learning algorithm are the desired position and the desired contact force $y = \left[X_d, Y_d, Z_d, F_{dx}, F_{dy}, F_{dz}\right] = [0.41,\ 0,\ 0.2265,\ 0,\ 0,\ 15] \in \mathbb{R}^6$. The desired position is the initial position of the end-effector in Cartesian space. The desired contact force in z-axis direction 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 target state in cost function is $x_{target} = [0.41,\ 0,\ 0.2265,\ 0,\ 0,\ 15]$. The strategy outputs are the impedance parameters $u = [K_d\ B_d]$. The number of the GP controller is $n = 10$. The ranges of the impedance parameters are set as $K_d \in [0.1\ 25]$ and $B_d \in [50\ 1000]$. The action penalty coefficient of the cost function is $\zeta = 0.03$. The measurement noise of the joint position, joint velocities and the contact force are set as $\delta \sim \mathcal{N}(0, 0.01^2)$ respectively. In the initial trial, the impedance parameters are initialized to stochastic variables that are subject to $\mathcal{N}(u_0|0.1u_{max}, 0.1u_{max})$.

5.2. Simulation Results

Figure 8 shows the simulation results of the force control learning system. The block marks in Figure 8a indicate whether the task is successful or not. The light blue dash-dotted line represents the cumulative cost during the explorations. Figure 9 details the learning process of the total 20 learning iterations. Figure 10 shows the joint trajectories after 4, 5, 10, 15, and 20 updates. Note that $N = 0$ denotes the initial trial, and the whole learning process is implemented automatically.

In the initial trial, the robot moves down slowly to search the contact plane and the end-effector begins to contact the plane at $T = 0.7$ s. Due to the large overshoot of the contact force, the task is failed until the fifth learning iteration. After the fifth learning iteration, the end-effector contacts the plane at $T = 0.34$ s, which is faster than the initial trial. The contact force reaches to the desired value rapidly with a little overshoot. The learning algorithm optimizes the control strategy continuously to reduce the cumulative cost. After seven iterations, the strategy's performance is stable, and the contact force reaches the desired value quickly without overshoot by adjusting the impedance parameters dynamically. The optimal strategy is learned after 20 iterations, and its cumulative cost is the smallest. The end-effector can contact the plane at $T = 0.3$ s.

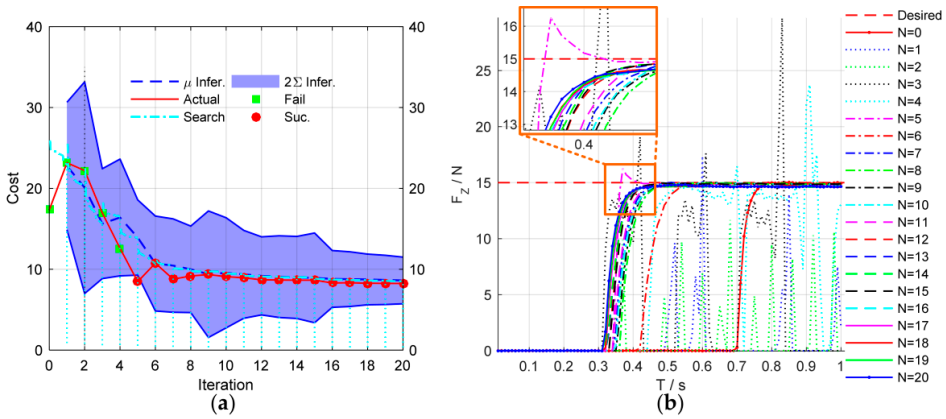

Figure 8. Simulation results of force control learning system. (**a**) The cost curve of learning process; the blue dotted line and the blue shade are the predicted cost mean and the 95% confidence interval. (**b**) The performances of force control during learning process.

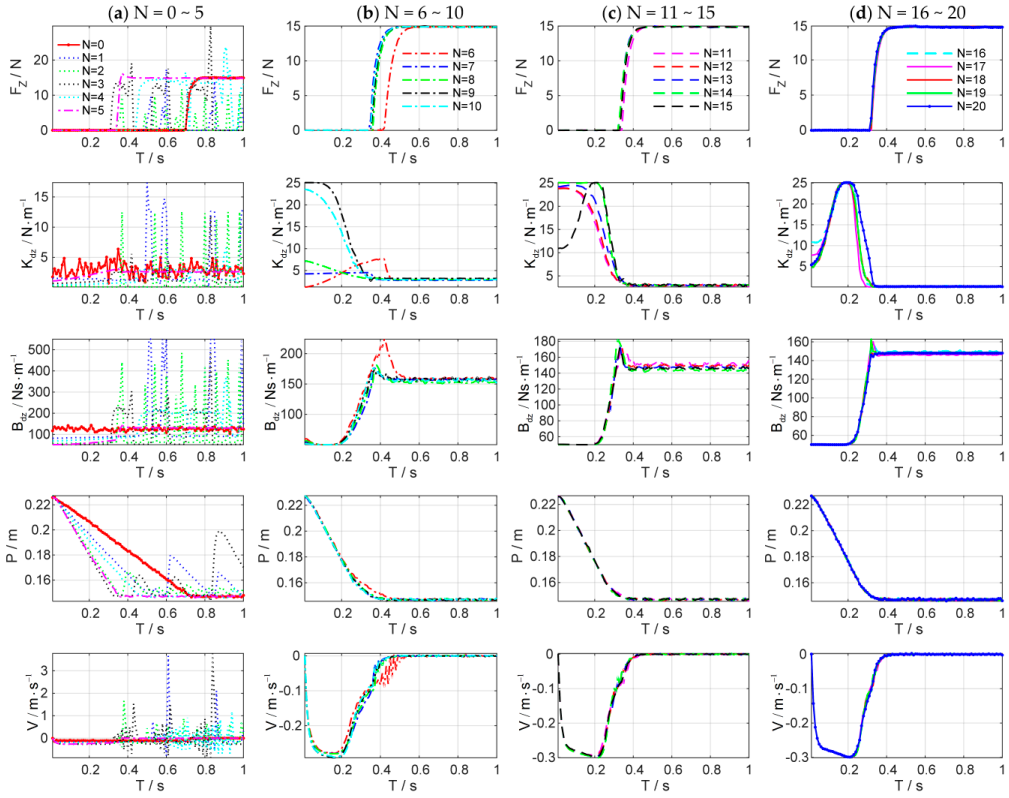

Figure 9. Learning process of the total 20 learning iterations. The first row: contact force in z-axis direction; the second row: Cartesian stiffness schedules; the third row: Cartesian damping schedules; the fourth row: position of the end-effector in z-axis direction; the fifth row: velocity of the end-effector in z-axis direction.

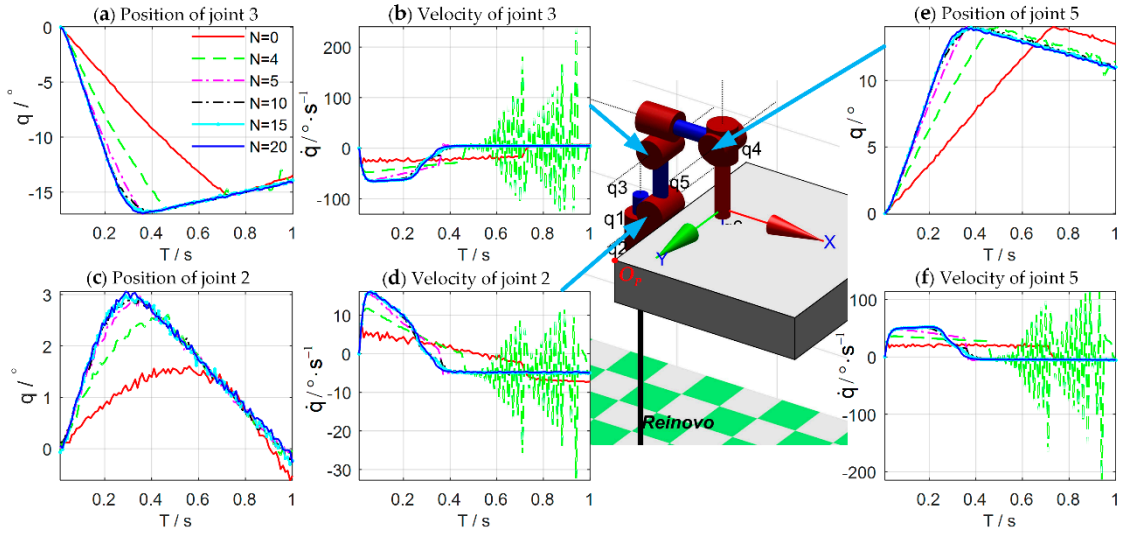

Figure 10. Joint trajectories after 4, 5, 10, 15, and 20 updates for the second, third, and fifth joint of the Reinovo robot.

Figure 11 shows the evolutions of force control and impedance profiles. The state evolutions predicted by the GP model can be represented by the blue dotted line and the blue shade, which denote the mean and the 95% confidence interval of the state prediction respectively. The historical sampled data are constantly enriched with the increase of interaction time, and the learned GP model is

optimized and stabilized gradually. When a good GP model is learned, it can be used as a faithful proxy of the real system (Figure 11b–d).

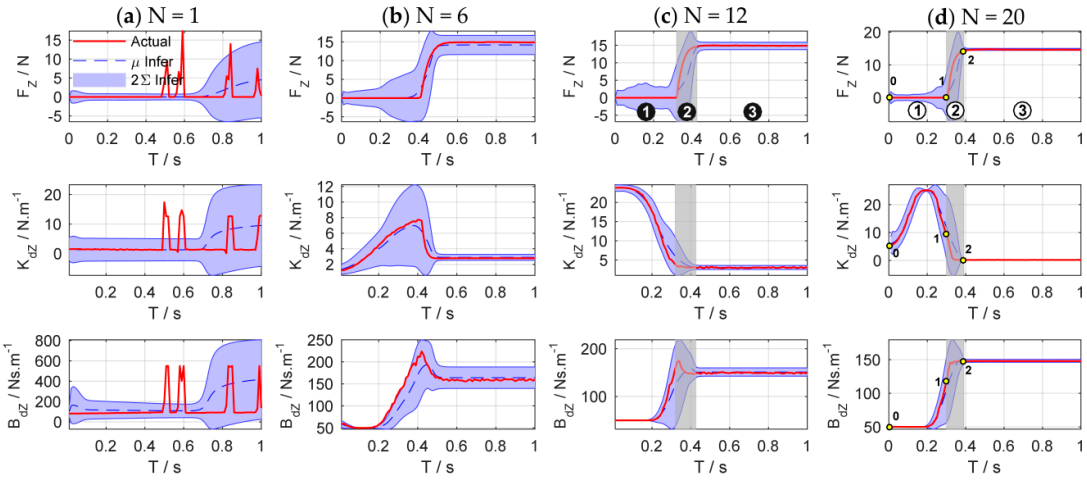

Figure 11. States evolution of force control. Columns (**a**–**d**) are the state evolutions of the 1st, 6th, 12th, and 20th learning iteration, respectively. The top row is the change of contact force F_z, the second row is the target stiffness K_{dz}, and the third row is the target damping B_{dz}.

As shown in Figure 11d, the process of impedance regulation can be divided into three phases: (1) the phase before contacting the plane; (2) the phase of coming into contact with the plane; and (3) the stable phase of contact with the plane. When the end-effector contact the environment, the manipulator is suddenly converted from free space motion to constrained space motion, and the collision is inevitable. The stiffness of the environment increases suddenly. 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.

Throughout the simulation, only five interactions (5 s of interaction time) are required to learn to complete the force tracking task successfully. Besides, the impedance characteristics of the learned strategy are similar to that of humans for force tracking. The simulation results verify the effectiveness and the efficiency of the proposed system.

5.3. Experimental Study

The hardware architecture of the system is shown in Figure 12. The test platform consists of six parts, an industrial manipulator, servo driver, Galil-DMC-2163 motion controller, Bioforcen F/T sensor, an industrial PC, and a workstation. All parts communicate with others through the switch. The motion controller communicates with the industrial PC through TCP/IP protocol running at 100 Hz. The sensor communicates with the industrial PC via Ethernet interface through TCP/IP and samples the data at 1 kHz. The specific implementation diagram of the algorithm is shown in Figure 13. The motion control of the robot is executed using Visual Studio 2010 on the industrial PC. The learning algorithm is implemented by MATLAB on the workstation. The workstation communicates with the industrial PC through UDP protocol.

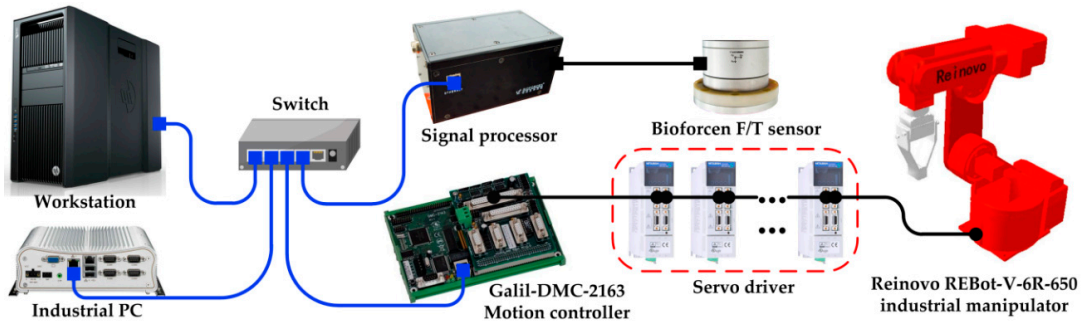

Figure 12. Hardware architecture of the system.

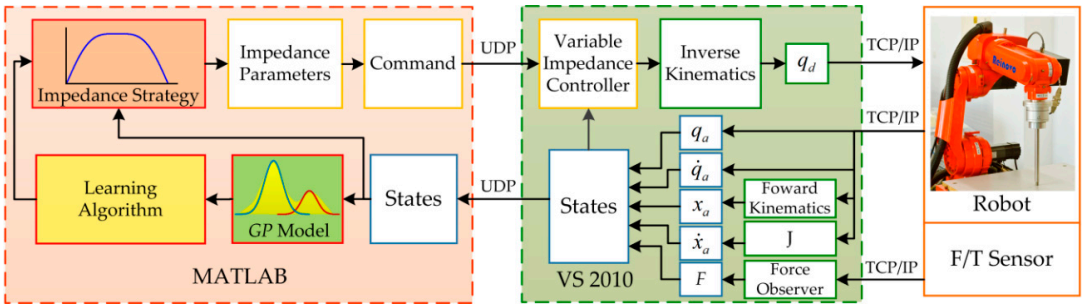

Figure 13. Implementation diagram of the algorithm.

To imitate the nonlinear variable characteristics of the circumstance during the force tracking task, a combination of spring and rope is taken as the unstructured contact environment. As shown in Figure 14, 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. The contact force is controlled by stretching the rope and the spring. Here, the specifications of the rope are unknown, and the rope is in a natural state of relaxation. The measurement range, length, and diameter of the spring dynamometer are 30 Kg, 0.185 m, and 29 mm, respectively. The exact values of the stiffness and damping of the springs are unknown to the system. In the experiments, the episode length (i.e., the prediction horizon) is $T = 3$ s. Other settings are consistent with those of simulations in Section 5.1.

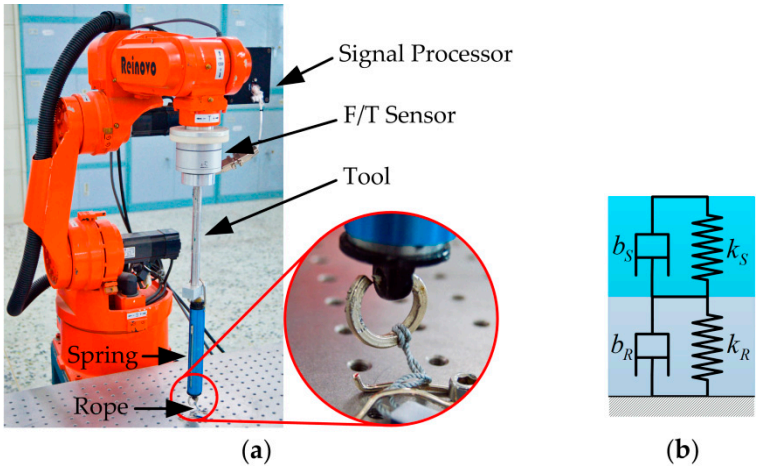

Figure 14. (**a**) Experimental setup; (**b**) simplified model of the contact environment.

5.4. Experimental Results

The experimental results are shown in Figure 15. Figure 16 details main iterations of the learning process. From the experimental results, we can see that in the initial trial, the manipulator moves slowly and the rope begins to be stretched at $T = 2.5$ s to increase the contact force. The contact force reaches 10.5 N at the end of the test, which implies that the task failed. In the second trial, 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, but the task failed because the overshoot is greater than 3 N. Only two learning iterations are needed to complete the task successfully. After 17 iterations, 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 the overshoot is suppressed effectively.

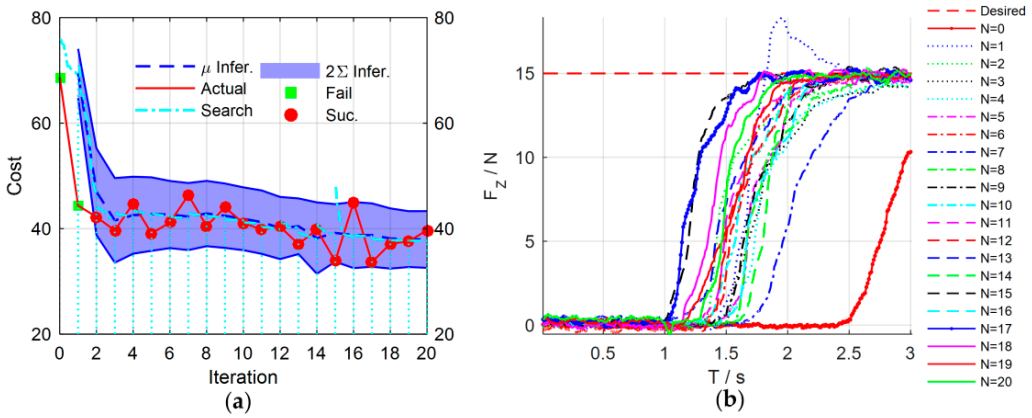

Figure 15. Experimental results of force control learning system. (**a**) The cost curve of learning process. (**b**) The performances of force control during learning process, including a total 20 learning iterations throughout the experiment.

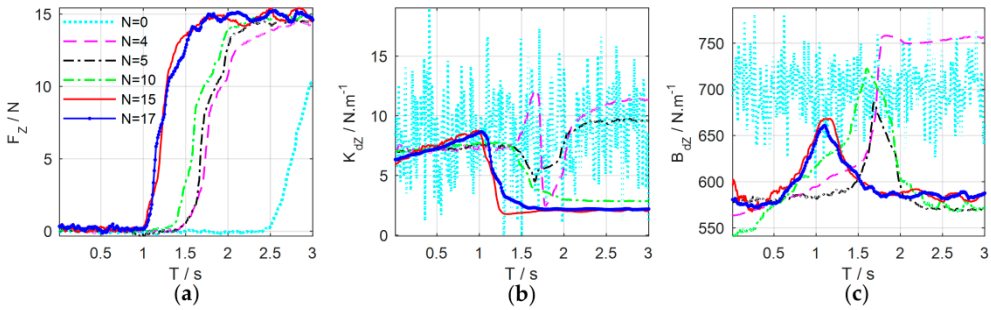

Figure 16. Main iterations of the learning process. (**a**) Contact force; (**b**) Cartesian stiffness schedules; (**c**) Cartesian damping schedules.

Figure 17 shows the evolutions of force control and impedance profiles. The bottom row shows the stretching process of the combination of the rope and the spring. The joint trajectories and Cartesian trajectories during the 20th experiment iteration are shown in Figure 18. The trajectories of other iterations are similar to those of the 20th iteration. The stretching process of the combination can be divided into four phases, just as the shaded areas shown. Table 1 summarizes the key states of the four phases. The corresponding subscripts of F_z, K_{dz}, and B_{dz} are shown in Figure 17d while the subscripts of position (P) and velocity (V) are shown in Figure 18c,d.

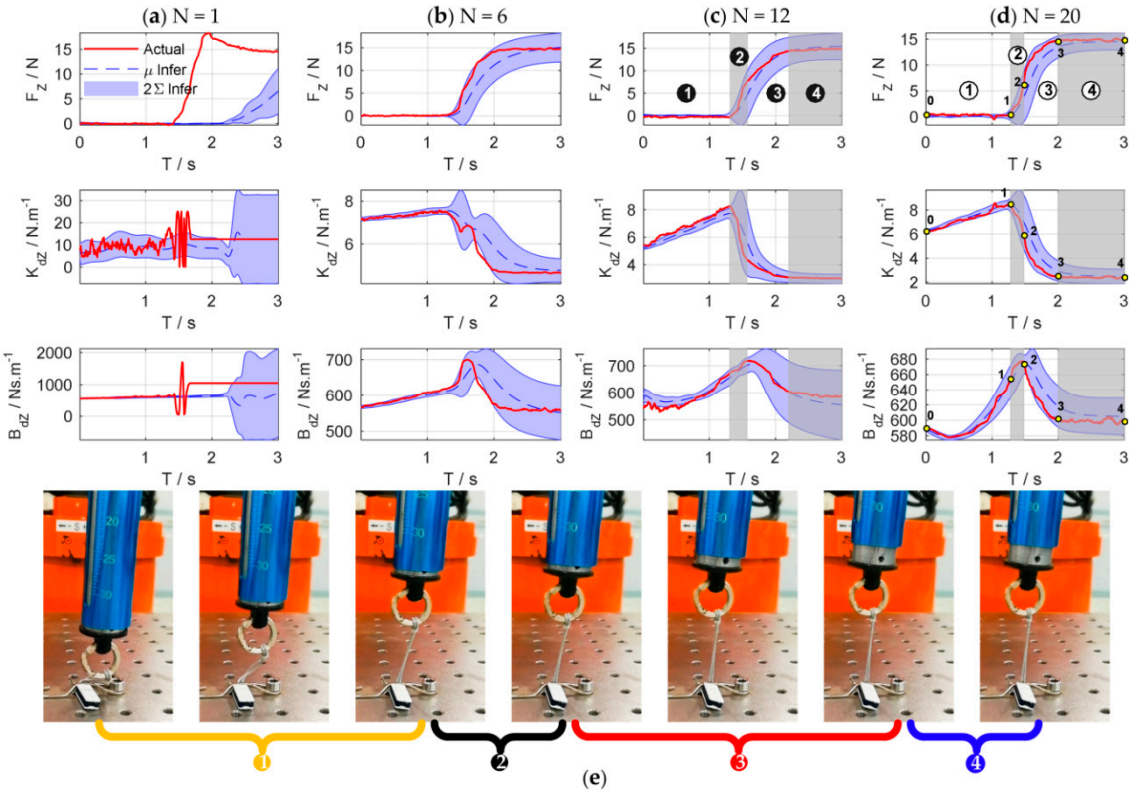

Figure 17. States evolution of force control. Columns (**a**–**d**) are the state evolutions of the 1st, 6th, 12th, and 20th learning iteration, respectively. The top row shows the contact force F_z, the second row shows the profile of stiffness K_{dz}, and the third row shows the profile of damping B_{dz}.

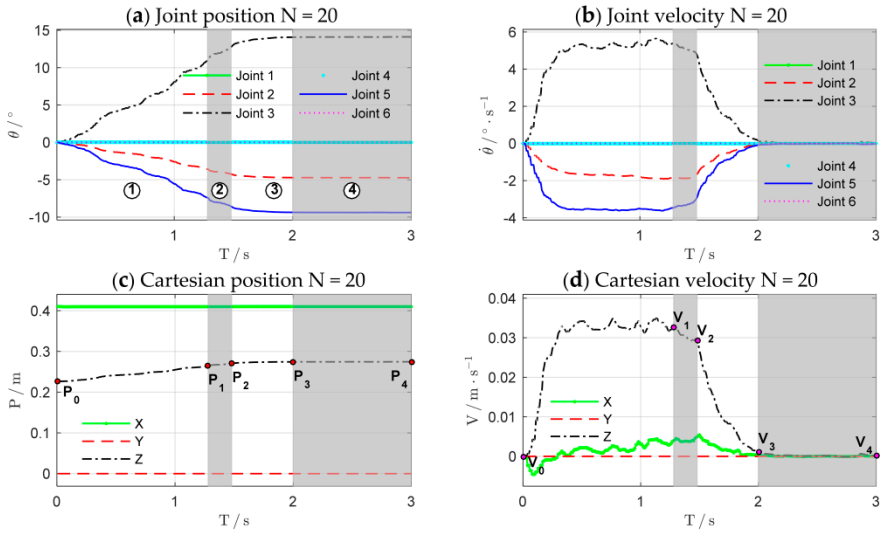

Figure 18. Trajectories during the 20th experiment iteration. (**a**) Joint position; (**b**) Joint velocity; (**c**) Cartesian position of the end-effector; (**d**) Cartesian velocity of the end-effector.

Table 1. Key states of the four phases during the 20th iteration.

Subscript	T/s	P/m	$V/m \cdot s^{-1}$	F_z/N	$K_{dz}/N \cdot m^{-1}$	$B_{dz}/Ns \cdot m^{-1}$
0	0.00	0.2265	0.0000	0.393	6.246	588.97
1	1.28	0.2642	0.0327	0.326	8.468	653.99
2	1.48	0.2701	0.0292	6.095	5.888	673.83
3	2.00	0.2744	0.0010	14.625	2.469	601.93
4	3.00	0.2745	0.0000	14.874	2.379	598.53

The four phases of impedance regulation are corresponded to the force control process mentioned above:

1. $T_0 - T_1$: Phase before stretching the rope. The manipulator moves freely in the free space. To tighten the rope quickly, the movement of the end-effector increases as the impedance 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. The stiffness of the 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. Although the environment changes suddenly, the controller does 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. In this phase, the stiffness and damping of the controller are kept at minimum so that the system maintains the ability of compliance.

There are total 20 learning iterations throughout the experiment. 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 learned GP model can be improved gradually. After two learning iterations, which means that only 6 s of interaction time is required, a sufficient dynamic model and strategy can be learned to complete the force tracking task successfully. The experimental results above verify that the proposed force control learning system is data-efficient. It is mainly because the system explicitly establishes the transition dynamics that are used for internal virtual simulations and predictions, and the optimal strategy is improved by evaluations. In this way, more efficient information could be extracted from the sampled data.

5.5. Comparative Experiments

5.5.1. Environmental Adaptability

The results above have verified that the learned strategy could adjust the impedance profiles to adapt to the environmental change in the episodic case. However, what happens if the whole contact environment changes? In order to verify the environmental adaptability of the proposed learning variable impedance control method, we use another different spring dynamometer to build the unstructured contact environment. The measurement range, length, and diameter of the second spring dynamometer are 15 Kg, 0.155 m, and 20 mm, respectively. It implies that the stiffness and

location of the environment are all changed. Other experimental setups are consistent with the Section 5.3. The initial strategy for the second environment is the learned (sub)optimal strategy for the first spring (N = 17 in Section 5.4). The results are illustrated in Figure 19.

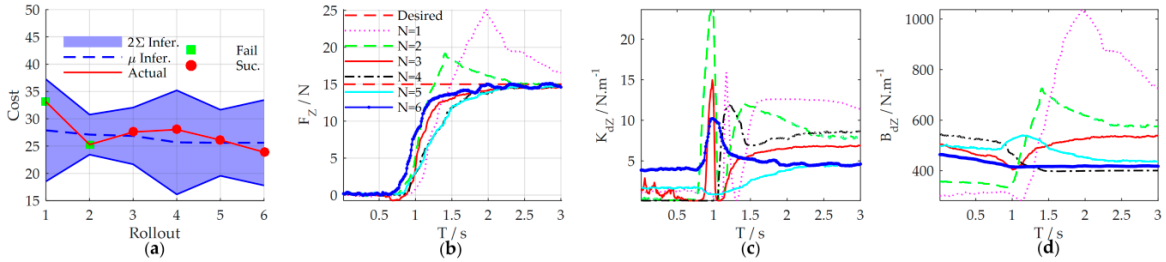

Figure 19. Experimental results of environmental adaptability. (**a**) Cost curve; (**b**) contact force; (**c**) Cartesian stiffness schedules; (**d**) Cartesian damping schedules.

From the experimental results, we can see that in the initial application of the control strategy, even though the impedance profile is regulated online according to the learned strategy, the task is failed. This is mainly because the whole environment is changed a lot, the strategy learned for the previous environment is not suitable for current environment. However, as the learning process continues to optimize, the learned (sub)optimal strategy could adapt to the new environment and successfully complete the task after two learning iterations. Therefore, the proposed method could adapt to new environments, taking advantage of its learning ability.

5.5.2. Comparison of Force Control Performance

Variable impedance control can regulate the task-specific impedance parameters at different phases to complete the task more effectively, which is the characteristic that the constant impedance control is not equipped. Next, the proposed learning variable impedance control is compared with the constant impedance control and the adaptive impedance control [13] to verify the effectiveness of force control. The stiffness of the constant impedance controller is set as $K_d = 10$ while the stiffness of the adaptive impedance controller is set as $K_d = 0$ [13]. The damping of the controller could be adjusted manually or automatically. Experimental comparison results are illustrated in Figure 20.

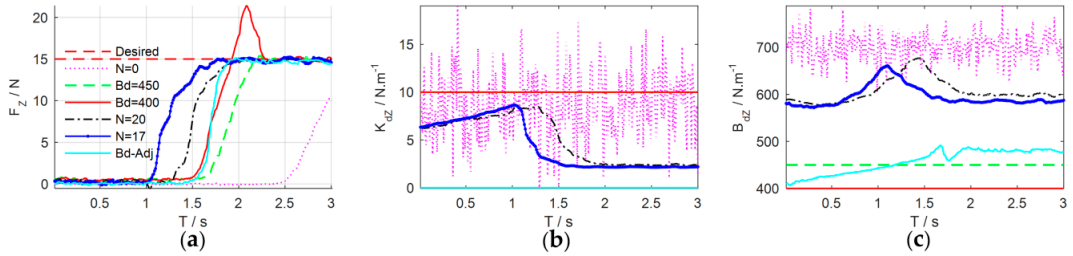

Figure 20. Experimental comparison results. (**a**) Force control performance; (**b**) target stiffness; (**c**) target damping.

In order to quantitatively compare the performances, we use four indicators to quantify the performances. The first indictor is the time T_{free} when the force begins to increase, i.e., the movement time of the robot in free space. The second indicator is the accumulated cost $J^{\pi}(\theta)$ which is defined in Equation (33). Note that the accumulated cost includes two parts: the cost caused by the error to the target state (Equation (36)) and the cost caused by the control gains (Equation (37)). The third one is the root-mean-square error ($RMSE$) of the contact force

$$RMSE = \sqrt{\frac{\sum_{t=1}^{H} (F_z(t) - F_{zd})^2}{H}}, \tag{44}$$

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

An energy indicator is defined to indicate the accumulated consumed energy during the control process

$$Ey = \sum_{t=1}^{H} \sum_{i=1}^{6} \frac{1}{2} m_i \dot{\theta}_i^2(t), \tag{45}$$

where m_i is the mass of the i^{th} joint and $\dot{\theta}_i$ is the angular velocity of the i^{th} joint. Actually, the masses of the joints are unknown accurately. Without loss of generality, the masses can be approximately set as $[m_1,\ m_2,\ m_3,\ m_4,\ m_5,\ m_6] = [1,\ 1,\ 1,\ 0.5,\ 0.3,\ 0.1]$.

Table 2 reveals the performance indicators of the three impedance controllers. Compared with the constant/adaptive impedance control, the learning variable impedance control has the minimum indicator values, which indicates that the proposed system is effective. Obviously, the performance of the adaptive impedance control is better than the constant one, but still worse than the optimal impedance strategy learned by the proposed learning variable impedance control.

Table 2. Performance indicators.

Mode	Name	T_{free}/s	Cost	RMSE	Ey ($\times 10^3$)	Overshoot
Constant Impedance control	Bd = 450	1.70	43.29	11.29	5.51	No
	Bd = 400	1.50	41.09	10.83	5.85	Yes
Adaptive Impedance control	Bd-Adj	1.50	37.23	11.09	5.38	No
Learning variable Impedance control	N = 20	1.25	39.71	10.19	2.33	No
	N = 17	1.00	33.64	9.23	2.08	No

5.5.3. Learning Speed

In order to further illustrate the efficiency of the proposed method, we compared the learning speed with state-of-the-art learning variable impedance control method, C-PI^2, through the via-gain task [37]. The cost function of C-PI^2 is chosen as $r_t = w_1 \delta(t - 0.4) \| K_1 - K_t^P \|$ with $w_1 = 1 \times 10^8$ $K_1 = 15$. The cost function of our method is chosen as the Equation (35) with $x_{target} = 15$, $\zeta = 0.03$. The cost curve of learning process is shown in Figure 21a. Similar to other studies [37], to indicate the data-efficiency of the method, we take the required rollouts to get the satisfactory strategy as the indicator of the learning speed. The results show that C-PI^2 converges after about 92 rollouts, whereas our method needs only 4 rollouts.

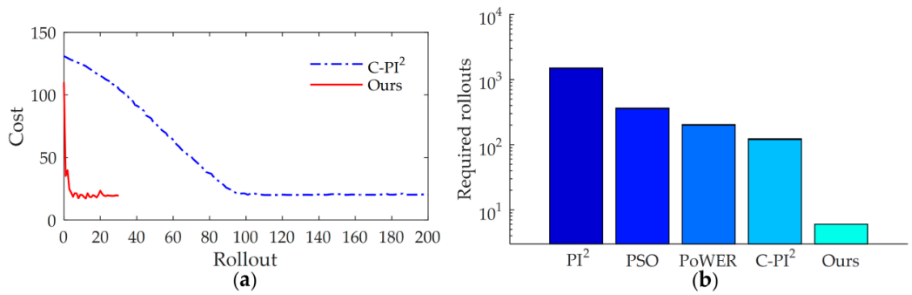

Figure 21. (**a**) The cost curve of the learning process. (**b**) Comparison of learning speed with other learning variable impedance control methods.

Current learning variable impedance control methods usually require hundreds or thousands of rollouts to get a stable strategy. For tasks that are sensitive to the contact force, too many physical interactions with the environment during the learning process are often infeasible. Improving the efficiency of learning method is critical. Figure 21b shows the comparison of learning speed with other learning variable impedance control methods. From the results of [4,37,50], 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 learning speed of C-PI^2 is much higher than that of previous methods but is still slower than our method. Our method outperforms other learning variable impedance control methods by at least one order of magnitude.

6. Discussion

According to the definition of the cost function (35)–(37), we can learn that decreasing the distance between x_t and x_{target} and keeping the impedance parameters u_t at a low level will be beneficial for minimizing the accumulated cost. Consequently, small damping parameters will make the robot move quickly to contact with the environment to reduce the distance between x_t and x_{target}. Unfortunately, small damping could reduce the positioning accuracy of the robot and thus make the system with poor ability to suppress disturbances. On the contrary, large damping could improve the system's ability of suppressing disturbances and reduce the speed of motion. It will lead to task failure if the impedance parameters cannot be regulated to suppress the overshoot. Hence, the learning algorithm must make a tradeoff between rapidity and stability to achieve the proper control strategy. By regulating the target stiffness and damping independently at different phases, the robot achieves rapid contact with the environment while the overshoot is effectively suppressed.

The impedance characteristics of the learned strategy are similar to the strategy that employed by humans for force tracking. 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 muscle relaxation to minimize the energy consumption. Our system learns the optimal impedance strategy automatically through continuous explorations. This is different from the methods of imitating human impedance behavior, such as [18] and [24], which usually need additional device—e.g., EMG electrodes—to transfer human skills to the robots by demonstration.

The proposed learning variable impedance control method emphasizes the data-efficiency, i.e., sample efficiency, by learning a GP model for the system. This is critical for learning to perform force-sensitive tasks. Note that only the application of the strategy requires physical interacting with the robot; internal simulations and strategy learning only use the learned GP model. Although a fast converging controller is found, the proposed method still has the computation intensive limitation. The computational time for each rollout on a workstation computer with an Intel(R) Core(TM) i7-6700 K CPU@4.00 GHz is detailed in Figure 22.

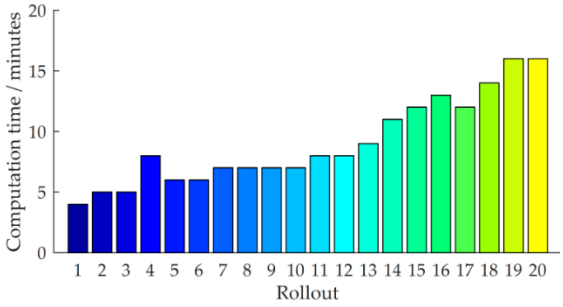

Figure 22. Computational time for each rollout.

Obviously, between the trials the method requires approximately 9 min to find the (sub)optimal strategy. With the increase of sample data set, the computational time is increasing gradually, because the kernel matrices need to be stored and inverted repeatedly. The most demanding computations are the predictive distribution and the derivatives for prediction using the GP model.

7. Conclusions and Future Works

In this paper, we presented a data-efficient learning variable impedance control method that enables the industrial robots automatically learn to control the contact force in the unstructured environment. The goal was to improve the sampling efficiency and reduce the required physical interactions during learning process. To do so, a GP model was learned as the faithful dynamics of the system, which is then used for internal simulations to improve the data-efficiency by predicting the long-term state evolution. This method learned an impedance regulation strategy, based on which the impedance profiles were regulated online to track the desired contact force. In this way, the flexibility and adaptivity of the system were enhanced. It is worth noting that the optimal impedance control strategy, which is equipped with the similar impedance characteristics of humans, is automatically learned through several iterations. There is no need to transfer human skills to the robot with additional sampling devices. The effectiveness and data-efficiency of the system were verified through simulations and experiments on the six-DoF Reinovo industrial robot. The learning speed of this system outperforms other learning variable impedance control methods by at least one order of magnitude.

Currently, the described work only focuses on the efficient learning of force control. In the future work, we will extend this system to learn to complete more complex tasks that are sensitive to contact force, such as assembly tasks of fragile components. Furthermore, parallel and online implementation of this method to improve computational efficiency would be a meaningful and interesting research direction.

Author Contributions: G.X. conceived the main idea; C.L. designed the algorithm and system; Z.Z. designed the simulation; C.L. and X.X. designed and performed the experiments; Z.Z. and Q.Z. analyzed the data; C.L. and Z.Z. wrote the paper; Q.Z. discussed and revised the paper; G.X. supervised the research work.

Funding: This research was supported by the National Natural Science Foundation of China (grant no. U1530119) and China Academy of Engineering Physics.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. Robotics: Modelling, planning and control. *Adv. Textb. Control Signal Process.* **2009**, *4*, 76–82.
2. Hogan, N. Impedance control: An approach to manipulation: Part I—Theory. *J. Dyn. Syst. Meas. Control* **1985**, *107*, 1–7. [CrossRef]
3. Burdet, E.; Osu, R.; Franklin, D.W.; Milner, T.E.; Kawato, M. The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature* **2001**, *414*, 446–449. [CrossRef] [PubMed]
4. Kieboom, J.V.D.; Ijspeert, A.J. Exploiting natural dynamics in biped locomotion using variable impedance control. In Proceedings of the 13th IEEE-RAS International Conference on Humanoid Robots, Atlanta, GA, USA, 15–17 October 2013; pp. 348–353. [CrossRef]
5. Buchli, J.; Stulp, F.; Theodorou, E.; Schaal, S. Learning variable impedance control. *Int. J. Robot. Res.* **2011**, *30*, 820–833. [CrossRef]
6. Calinon, S.; Sardellitti, I.; Caldwell, D.G. Learning-based control strategy for safe human-robot interaction exploiting task and robot redundancies. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 249–254. [CrossRef]
7. Kormushev, P.; Calinon, S.; Caldwell, D.G. Robot motor skill coordination with em-based reinforcement learning. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 3232–3237. [CrossRef]

8. Kronander, K.; Billard, A. Online learning of varying stiffness through physical human-robot interaction. In Proceedings of the IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 1842–1849. [CrossRef]
9. Yang, C.; Ganesh, G.; Haddadin, S.; Parusel, S.; Albu-Schaeffer, A.; Burdet, E. Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE Trans. Robot.* **2011**, *27*, 918–930. [CrossRef]
10. Kronander, K.; Billard, A. Stability considerations for variable impedance control. *IEEE Trans. Robot.* **2016**, *32*, 1298–1305. [CrossRef]
11. Medina, J.R.; Sieber, D.; Hirche, S. Risk-sensitive interaction control in uncertain manipulation tasks. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 502–507. [CrossRef]
12. Braun, D.; Howard, M.; Vijayakumar, S. Optimal variable stiffness control: Formulation and application to explosive movement tasks. *Auton. Robots* **2012**, *33*, 237–253. [CrossRef]
13. Duan, J.; Gan, Y.; Chen, M.; Dai, X. Adaptive variable impedance control for dynamic contact force tracking in uncertain environment. *Robot. Auton. Syst.* **2018**, *102*, 54–65. [CrossRef]
14. Ferraguti, F.; Secchi, C.; Fantuzzi, C. A tank-based approach to impedance control with variable stiffness. In Proceedings of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 4948–4953. [CrossRef]
15. Takahashi, C.; Scheidt, R.; Reinkensmeyer, D. Impedance control and internal model formation when reaching in a randomly varying dynamical environment. *J. Neurophysiol.* **2001**, *86*, 1047–1051. [CrossRef] [PubMed]
16. Tsuji, T.; Tanaka, Y. Bio-mimetic impedance control of robotic manipulator for dynamic contact tasks. *Robot. Auton. Syst.* **2008**, *56*, 306–316. [CrossRef]
17. Lee, K.; Buss, M. Force tracking impedance control with variable target stiffness. *IFAC Proc. Volumes* **2008**, *41*, 6751–6756. [CrossRef]
18. Yang, C.; Zeng, C.; Liang, P.; Li, Z.; Li, R.; Su, C.Y. Interface design of a physical human-robot interaction system for human impedance adaptive skill transfer. *IEEE Trans. Autom. Sci. Eng.* **2017**, *15*, 329–340. [CrossRef]
19. Kronander, K.; Billard, A. Learning compliant manipulation through kinesthetic and tactile human-robot interaction. *IEEE Trans. Haptics* **2014**, *7*, 367–380. [CrossRef] [PubMed]
20. Li, M.; Yin, H.; Tahara, K.; Billard, A. Learning object-level impedance control for robust grasping and dexterous manipulation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6784–6791. [CrossRef]
21. Yang, C.; Zeng, C.; Fang, C.; He, W.; Li, Z. A dmps-based framework for robot learning and generalization of humanlike variable impedance skills. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 1193–1203. [CrossRef]
22. Yang, C.; Luo, J.; Pan, Y.; Liu, Z.; Su, C.Y. Personalized variable gain control with tremor attenuation for robot teleoperation. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**, *PP*, 1–12. [CrossRef]
23. Ajoudani, A.; Tsagarakis, N.; Bicchi, A. Tele-impedance: Teleoperation with impedance regulation using a body–machine interface. *Int. J. Robot. Res.* **2012**, *31*, 1642–1656. [CrossRef]
24. Howard, M.; Braun, D.J.; Vijayakumar, S. Transferring human impedance behavior to heterogeneous variable impedance actuators. *IEEE Trans. Robot.* **2013**, *29*, 847–862. [CrossRef]
25. Polydoros, A.S.; Nalpantidis, L. Survey of model-based reinforcement learning: Applications on robotics. *J. Intell. Robot. Syst.* **2017**, *86*, 153–173. [CrossRef]
26. Kober, J.; Peters, J. Reinforcement learning in robotics: A survey. *Int. J. Robot. Res.* **2013**, *32*, 1238–1274. [CrossRef]
27. Koropouli, V.; Hirche, S.; Lee, D. Generalization of force control policies from demonstrations for constrained robotic motion tasks. *J. Intell. Robot. Syst.* **2015**, *80*, 1–16. [CrossRef]
28. Mitrovic, D.; Klanke, S.; Howard, M.; Vijayakumar, S. Exploiting sensorimotor stochasticity for learning control of variable impedance actuators. In Proceedings of the 10th IEEE-RAS International Conference on Humanoid Robots, Nashville, TN, USA, 6–8 December 2010; pp. 536–541. [CrossRef]
29. Stulp, F.; Buchli, J.; Ellmer, A.; Mistry, M.; Theodorou, E.A.; Schaal, S. Model-free reinforcement learning of impedance control in stochastic environments. *IEEE Trans. Auton. Ment. Dev.* **2012**, *4*, 330–341. [CrossRef]
30. Du, Z.; Wang, W.; Yan, Z.; Dong, W.; Wang, W. Variable admittance control based on fuzzy reinforcement learning for minimally invasive surgery manipulator. *Sensors* **2017**, *17*, 844. [CrossRef] [PubMed]

31. Li, Z.; Liu, J.; Huang, Z.; Peng, Y.; Pu, H.; Ding, L. Adaptive impedance control of human-robot cooperation using reinforcement learning. *IEEE Trans. Ind. Electron.* **2017**, *64*, 8013–8022. [CrossRef]
32. Deisenroth, M.; Neumann, G.; Peters, J. A survey on policy search for robotics. *J. Intell. Robot. Syst.* **2013**, *2*, 1–142. [CrossRef]
33. Nagabandi, A.; Kahn, G.; Fearing, R.S.; Levine, S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. *arXiv*, 2017.
34. Kalakrishnan, M.; Righetti, L.; Pastor, P.; Schaal, S. Learning force control policies for compliant manipulation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 4639–4644. [CrossRef]
35. Pastor, P.; Kalakrishnan, M.; Chitta, S.; Theodorou, E. Skill learning and task outcome prediction for manipulation. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3828–3834. [CrossRef]
36. Rombokas, E.; Malhotra, M.; Theodorou, E.; Matsuoka, Y.; Todorov, E. Tendon-driven variable impedance control using reinforcement learning. In *Robotics: Science and Systems VIII*; MIT Press: Cambridge, MA, USA, 2012.
37. Winter, F.; Saveriano, M.; Lee, D. The role of coupling terms in variable impedance policies learning. In Proceedings of the 9th International Workshop on Human-Friendly Robotics, Genova, Italy, 29–30 September 2016.
38. Shadmehr, R.; Mussa-Ivaldi, F.A. Adaptive representation of dynamics during learning of a motor task. *J. Neurosci.* **1994**, *14*, 3208–3224. [CrossRef] [PubMed]
39. Franklin, D.W.; Burdet, E.; Tee, K.P.; Osu, R.; Chew, C.-M.; Milner, T.E.; Kawato, M. Cns learns stable, accurate, and efficient movements using a simple algorithm. *J. Neurosci.* **2008**, *28*, 11165–11173. [CrossRef] [PubMed]
40. Villagrossi, E.; Simoni, L.; Beschi, M.; Pedrocchi, N.; Marini, A.; Molinari Tosatti, L.; Visioli, A. A virtual force sensor for interaction tasks with conventional industrial robots. *Mechatronics* **2018**, *50*, 78–86. [CrossRef]
41. Wahrburg, A.; Bös, J.; Listmann, K.D.; Dai, F.; Matthias, B.; Ding, H. Motor-current-based estimation of cartesian contact forces and torques for robotic manipulators and its application to force control. *IEEE Trans. Autom. Sci. Eng.* **2018**, *15*, 879–886. [CrossRef]
42. Wahrburg, A.; Morara, E.; Cesari, G.; Matthias, B.; Ding, H. Cartesian contact force estimation for robotic manipulators using Kalman filters and the generalized momentum. In Proceedings of the IEEE International Conference on Automation Science and Engineering (CASE), Gothenburg, Sweden, 24–28 August 2015; pp. 1230–1235. [CrossRef]
43. Wittenburg, J. *Dynamics of Multibody Systems*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007; ISBN 9783540739142.
44. Floreano, D.; Husbands, P.; Nolfi, S. *Springer Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2008.
45. Ott, C. *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 49, ISBN 978-3-540-69253-9.
46. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*; MIT Press: Cambridge, MA, USA, 2006; pp. 69–106. ISBN 026218253X.
47. Deisenroth, M.P.; Fox, D.; Rasmussen, C.E. Gaussian processes for data-efficient learning in robotics and control. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 408–423. [CrossRef] [PubMed]
48. Diesenroth, M.P. Highlight Talk: Gaussian Processes for Data Efficient Learning in Robotics and Control. AISTATS 2014. Available online: https://www.youtube.com/watch?v=dWsjjszwfi0 (accessed on 12 November 2014).
49. Deisenroth, M.P.; Rasmussen, C.E. Pilco: A model-based and data-efficient approach to policy search. In Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011; pp. 465–472.
50. Kober, J.; Peters, J. Learning motor primitives for robotics. In Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 2112–2118. [CrossRef]