robotics

MDPI

Article

Constraint-Aware Policy for Compliant Manipulation

Daichi Saito [1,*], Kazuhiro Sasabuchi [2], Naoki Wake [2], Atsushi Kanehira [2], Jun Takamatsu [2], Hideki Koike [1] and Katsushi Ikeuchi [2]

[1] School of Computing, Tokyo Institute of Technology, Tokyo 152-8550, Japan; koike@c.titech.ac.jp
[2] Applied Robotics Research, Microsoft, Redmond, WA 98052, USA; kazuhiro.sasabuchi@microsoft.com (K.S.); naoki.wake@microsoft.com (N.W.); atsushi.kanehira@microsoft.com (A.K.); jun.takamatsu@microsoft.com (J.T.); katsuike@microsoft.com (K.I.)
* Correspondence: saito.d.ah@m.titech.ac.jp

Abstract: Robot manipulation in a physically constrained environment requires compliant manipulation. Compliant manipulation is a manipulation skill to adjust hand motion based on the force imposed by the environment. Recently, reinforcement learning (RL) has been applied to solve household operations involving compliant manipulation. However, previous RL methods have primarily focused on designing a policy for a specific operation that limits their applicability and requires separate training for every new operation. We propose a constraint-aware policy that is applicable to various unseen manipulations by grouping several manipulations together based on the type of physical constraint involved. The type of physical constraint determines the characteristic of the imposed force direction; thus, a generalized policy is trained in the environment and reward designed on the basis of this characteristic. This paper focuses on two types of physical constraints: prismatic and revolute joints. Experiments demonstrated that the same policy could successfully execute various compliant manipulation operations, both in the simulation and reality. We believe this study is the first step toward realizing a generalized household robot.

Keywords: compliant manipulation; reinforcement learning; Learning-from-Observation

Citation: Saito, D.; Sasabuchi, K.; Wake, N.; Kanehira, A.; Takamatsu, J.; Koike, H.; Ikeuchi, K. Constraint-Aware Policy for Compliant Manipulation. *Robotics* **2024**, *13*, 8. https://doi.org/10.3390/robotics13010008

Academic Editors: Roman Mykhailyshyn and Ann Majewicz Fey

Received: 18 November 2023
Revised: 19 December 2023
Accepted: 24 December 2023
Published: 27 December 2023

1. Introduction

Many household operations require manipulating an object under a physically constrained environment, such as opening drawers and doors. A robotic system performing such household operations must be guaranteed not to damage the object or environment. Therefore, the robot needs to adjust its hand motion during the execution based on the force imposed by the environment, i.e., constraint force. This manipulation is called *compliant manipulation* [1]. There are an unpredictable amount of manipulations in the household environment; thus, the generalized controller to such manipulations is expected to realize a household robot.

This study investigates the generalization capability of a policy trained with a single environment and reward using reinforcement learning (RL) to various unseen manipulations. Although the RL-based approach [2–6] is more robust to the uncertainty associated with recognition of object information, such as pose, articulation, and shape than classical controllers [7], this requires a manual design of the training environment and reward specific to each manipulation. Thus, it is not scalable to the number of target manipulations. This issue is caused by the lack of the generalization of the policy to the unseen manipulations because this approach handles each manipulation independently.

Manipulations can be classified based on a physical constraint. In the previous study [8], a manipulation group is defined to have a common admissible/inadmissible direction, along which the object can/cannot move. For example, several manipulations, such as drawer opening, plate sliding, and pole pulling, belong to the same group because the object's admissible motion directions are constrained under a linear guide. If an object

tries to move in the inadmissible direction, the constraint exerts the force on the object. Thus, we notice that the manipulations grouped based on the constraint also have a common characteristic of the constraint force. Since compliant manipulation operations are executed leveraging the force, we design a single policy generalized to various unseen manipulations on the basis of the characteristic of the constraint force.

We propose the *constraint-aware policy*, which estimates the object's admissible direction using the constraint force. We train the policy to be generalized to unseen manipulations in the constraint group with a single environment and reward (Figure 1 Right). This environment and reward are designed assuming the *single-system condition* (Figure 1 Left) that the robot hand and the object move in unison and can be regarded as the composite body, where the internal forces, such as frictional forces, are canceled out. Thus, the policy can obtain the constraint force exerted on the object. The environment is designed as a simplification of the real-world manipulations by extracting the common characteristic of the physical constraint critical to compliant manipulation operations, which is the key to the generalization. The assumption is practically realistic because it can be easily satisfied by an execution design, such as moving the hand slowly. Under the single-system condition, the estimation error of the admissible direction decreases in accordance with the reduction in the magnitude of the constraint force; thus, the reward is calculated only utilizing the magnitude.

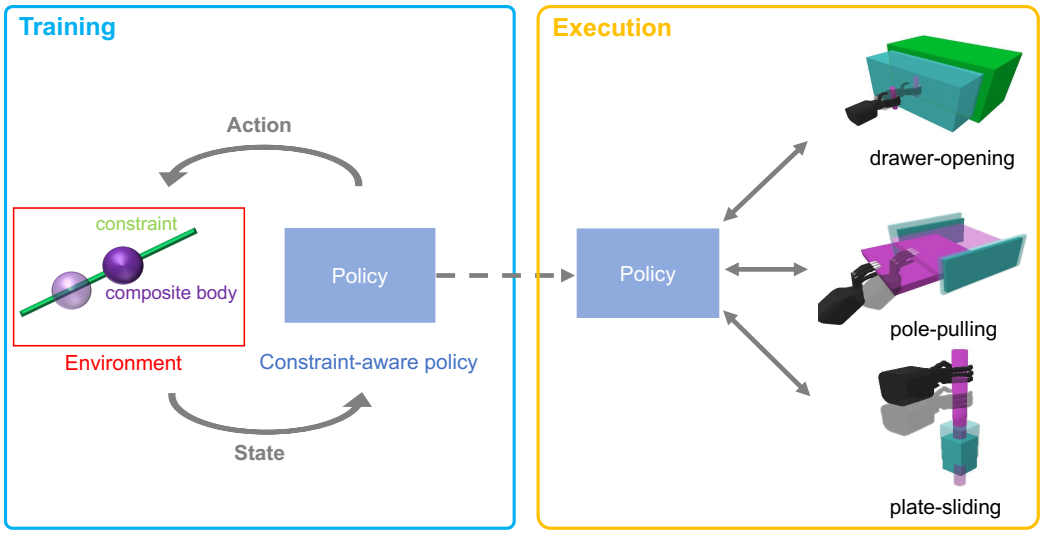

Figure 1. Concept behind constraint-aware policy. Various manipulations with a common physical constraint can be simplified as just a composite body and constraint, enabling the robot to obtain the constraint force, which we call the single-system condition. The constraint-aware policy is trained in the environment, which consists of the body and its constraint. The policy can be applied to various manipulations with the same physical constraint under the single-system condition.

In this study, we design the policy for the manipulation group with either a prismatic or revolute joint, which are representative constraints in the household environment. Under the constraint of either a prismatic or revolute joint, the object has one-degree-of-freedom translation and rotation, respectively. In addition to the generalization within a group, we investigate the transferability to a different group. Specifically, we consider transferring the policy for a prismatic joint to the manipulations with a revolute joint. To reuse the policy, we discuss the common and uncommon aspect of a revolute joint compared to a prismatic joint.

- The common aspect: Circular motion can be considered as a series of infinitesimal linear motions.
- The uncommon aspect: The hand must rotate in conjunction with the object to achieve the single-system condition.

From the common aspect, we can apply the same constraint-aware policy so that the policy estimates the admissible direction in both groups, whereas, owing to the uncommon aspect, the hand should rotate at execution only in manipulations with a revolute joint. To decide whether the hand needs to rotate or not, the type of the physical constraint, such as a prismatic or revolute joint, should be known.

To identify the constraint type, we leverage Learning-from-Observation (LfO) [9,10]. LfO provides the robot with hints for a manipulation through a multimodal one-shot human demonstration which includes a verbal instruction and hand movement. The instruction contains semantic information that enables a robot to infer the constraint type of the manipulated object. For example, the verbal instruction of "open the refrigerator door" is associated with a revolute joint. At execution, the robot selects the policy corresponding to the obtained constraint type from the preliminary prepared policies. In this study, we determine whether the physical constraint is a prismatic or revolute joint using the LfO system, and find out the necessity of the rotation of the hand.

We conducted experiments to investigate the generalization capability of the trained constraint-aware policy to various unseen household manipulations, such as a drawer opening, plate sliding, pole pulling, door opening, and handle rotating, in the simulator. We also compared the generalization with the classical controller [7], which is designed for the group with a prismatic joint. As a result of this experiment, unlike the classical controller, the constraint-aware policy can be executed in various manipulations. In addition, we evaluated the performance in the real-world using the policy and the LfO system, and demonstrated that the policy can be applied on a physical robot without additional training.

Toward a robot system capable of performing a wide range of manipulations, it is important to design the generalized policy for each manipulation group. Given that household manipulations can be classified based on their common constraints [8], the key to the generalized policies is to design an environment and reward focusing on a common characteristic within each constraint group. This study validated the concept of the constraint-aware policy for two fundamental physical constraints, those being a prismatic and revolute joint. We believe this study is the first step towards realizing the generalized household robot.

The contributions of this study are as follows:

- We proposed a constraint-aware policy which is trained using a single environment and reward and generalized to various unseen manipulations with a common physical constraint.
- We designed a simple training environment and reward function based on the constraint for the training of the constraint-aware policy.
- We demonstrated that unseen compliant manipulation operations can be executed on a physical robot using the constraint-aware policy and the LfO system.

The remainder of this paper is organized as follows. Section 2 reviews related work and states the focus of this paper. Section 3 introduces the constraint-aware policy. Section 4 describes the details of LfO to apply the constraint-aware policy in practice. Section 5 presents experiments for compliant manipulation using the constraint-aware policy in the simulation and reality. Section 6 discusses the result of our experiment and an extension of our method to hardware-level reusability and other constraints. Section 7 concludes this paper.

2. Related Work

In this study, we focus on a design of a policy which is robust to uncertainty associated with recognition, such as object pose and articulation, and object shape. In addition, we aim to train the policy, which is generalized to various unseen manipulations with a single environment and reward using RL. The representative approaches of compliant manipulation are the planning-based approach, classical closed-loop controller, and RL. In this section, we briefly review these approaches for compliant manipulation.

Previous research has focused on designing policies for opening drawers and doors. The pioneering work on door opening is [11], where robot motion is planned based on a known door model. In an unstructured environment, the model is unknown, and two methods can be used: geometry estimation and a closed-loop online controller to minimize force and torque. Several studies have been conducted on geometry estimation [12–20], where articulation pose is estimated from visual input, and a motion trajectory can be planned from this estimation result. However, the estimation accuracy is insufficient for compliant manipulation (e.g., $\sim 20^{\circ}$ estimation error in a rotation axis orientation on real-world data [18]), and causes the planning-based approach to fail. To deal with such estimation errors, other studies [21,22] have devised a robot mechanism for compliance.

Closed-loop controllers have been proposed in several studies, which can deal with uncertainty in geometry estimation [7,23–26]. In [23], an online controller was designed on the basis of a simple strategy in which the end-effector follows the path of the least force. Several studies have proposed online controllers based on this strategy [7,24–26]. These online controllers use the magnitude of force, which differs due to the change in the environment, and are not robust to the environmental change. To address this issue, we propose a constraint-aware policy using RL that can deal with uncertainty. Classical controllers also have a problem that requires manual parameter tuning. An adaptive controller is the solution to tune the parameters for a specific manipulation [27–29]. This adaptive tuning requires a real-world interaction between the robot and environment. In the case of our study, in which the object is constrained to the environment, a large force is directly applied to the robot and the object under an estimation error. Thus, it is dangerous to determine the parameters through the real-world interaction, and the controller is not appropriate for this study. Using the learning-based approach for compliant manipulation mitigates the issue on the parameter tuning.

Several studies have applied RL to train a policy for compliant manipulation [2–6,30]. These studies focused on the design of policies by preparing the environment and reward for only a specific manipulation. For example, these studies prepare a door-opening environment and calculate an angle of the door as the reward. For example, Urakami et al. proposed DoorGym, which is a training environment for generalizing the door-opening policy [5]. This trained policy can be generalized to doors with various doorknobs, lighting conditions, and environmental settings, but has focused only on door opening. Therefore, the trained policy is unable to be applied to other manipulations with the same constraint. There are several studies on RL which focus on designing a generalized policy for many varieties of manipulations [31–35]. However, this approach requires time and effort to prepare environments for all target manipulations to collect a large amount of data. In addition, this approach achieves an insufficient success rate on real-world application and needs to fine-tune the policy for a specific manipulation. In this study, we propose a policy generalized to manipulations with a common physical constraint, using a single environment and reward based on the common characteristic among these manipulations.

3. Method

In this study, we aim to train the policy generalized to various compliant manipulation operations, which is required in many household manipulations. Toward this policy, we design a single environment and reward based on the common characteristic of the physical constraint within a manipulation group. In this section, we explain an approach to the learning of this constraint-aware policy.

This section is organized as follows. Section 3.1 explains the target manipulation group in this study. Section 3.2 states assumptions for executing the constraint-aware policy. Section 3.3 introduces the training method of the policy for the target manipulation group in Section 3.1. Section 3.4 describes the technical details of satisfying the single-system condition, which is one of the assumptions explained in Section 3.2. These details are essential for an appropriate execution of the policy trained under the environment and reward in Section 3.3.

3.1. Target Manipulation Group

In this study, we focus on manipulation groups with the physical constraints, which are one-degree-of-freedom translation (prismatic joints) or rotation (revolute joints). These physical constraints are representative of the household environment. In the manipulations with a prismatic joint, such as drawer opening, plate sliding, and pole pulling, the object's admissible motion directions are constrained under a linear guide. As for the manipulations with a revolute joint, including door opening and handle rotating, the admissible directions are constrained under a rotational axis.

Compliant manipulations of the same group have a common characteristic of the constraint force. A large force is exerted on an object when the object tries to move along the inadmissible direction. Since compliant manipulation operations can be achieved using the force, we achieve various unseen manipulations within the same group by a single policy based on such a characteristic of the force.

3.2. Assumptions

The constraint-aware policy in this study is executed on the following assumptions.

Assumption 1: Single-system condition: The robot hand and object move in unison, where the internal forces between them are canceled out.

Assumption 2: The inertial force on the manipulated object is negligible.

Assumption 3: Friction in the joint mechanism is sufficiently weak such that the manipulated object can move smoothly along the desired trajectory.

Assumption 4: The workplane of the robot hand and direction of the rotation axis are known; thus, the robot hand and manipulated object move on a known plane.

These assumptions can be fulfilled in the manipulations we are focusing on. Assumptions 1 and 2 can be satisfied through the design of the manipulation, with Assumption 2 being satisfied by moving the manipulated object slowly. Assumption 1 is satisfied by a grasp mechanism and an additional policy to decrease torque exerted on the object. For more details of Assumption 1, see Section 3.4. Assumption 3 is satisfied by many household objects, as they are designed for easy handling by humans. Finally, this study focuses on objects with only one prismatic or revolute joint, which are representative of household environments; thus, regarding Assumption 4, the workplane can easily be obtained. These can be obtained using Learning-from-Observation (LfO), where a human provides manipulation instructions to a robot through a one-shot demonstration [9,10]. We can calculate the workplane from human hand trajectories. For more details on Assumption 4, see Section 4.

3.3. Training Design under Single-System Condition

Deep RL is employed to design the control policy, as it mitigates the requirement of manual parameter tuning and is robust to uncertainties, such as recognition error and sensor noise, unlike classical controllers [7,23,24].

To design the control policy, we assume compliant manipulation as a Markov decision process and apply deep RL to train a constraint-aware policy. The Markov decision process has a state space $\mathcal{S}$, action space $\mathcal{A}$, state transition $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, initial state distribution ρ_0, and reward $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. At each timestep t, an agent interacts with an environment with an action a_t determined from state s_t, resulting in s_{t+1} and r_{t+1}. The goal of RL is finding the optimal policy $\pi(a|s)$ that maximizes the cumulative reward $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)]$, where γ is the discount factor, $\gamma \in [0, 1)$, and T is the episode length.

In this study, the robot hand moves along a motion direction $\boldsymbol{d} \in \mathbb{R}^3$ and observes a force $\boldsymbol{F} \in \mathbb{R}^3$. We train the policy π to estimate an optimal motion direction while the hand moves along the estimated direction.

3.3.1. Training Environment

The training environment is designed based on the single-system condition. This environment consists of a single composite body and a prismatic joint (Figure 2). This

composite body represents the robot hand and manipulated object under the single-system condition. At each timestep, a force exerted on the body $\boldsymbol{F}$ is obtained as a result of interaction between the body and constraint. The constraint is represented as a constraint equation, and the force is calculated by solving the equation of motion, which includes the constraint force [36]. The single-system condition guarantees that $\boldsymbol{F}$, measured at the robot wrist, is identical to the constraint force on the body, as any internal forces between the hand and the object can be ignored.

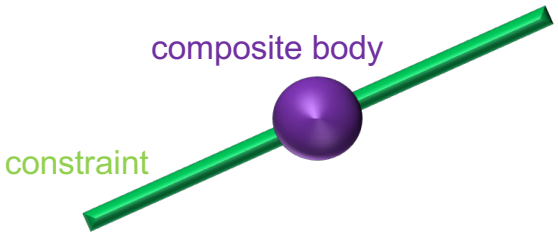

Figure 2. Training environment concept, consisting of the single composite body (purple sphere) and prismatic joint (green line).

This environmental design offers the advantage of a low simulation cost, as it is unnecessary to consider unstable factors, such as contact simulations between objects. This improves simulation speed and leads to faster training. Furthermore, the policy trained in this environment can be easily adapted to different robot hands because it is independent of the specific characteristics of the robot hand itself.

3.3.2. State and Action

At timestep t, the state $\boldsymbol{s}_t \in \mathbb{R}^6$ consists of the normalized force obtained from a sensor $\bar{\boldsymbol{F}}_t \in \mathbb{R}^3$ ($\bar{\boldsymbol{F}}_t = \frac{\boldsymbol{F}_t}{\|\boldsymbol{F}_t\|_2}$) and the motion direction of the robot hand $\boldsymbol{d}_t \in \mathbb{R}^3$. Utilizing the normalized force vector is important because the normalization makes the policy robust to a change in the magnitude of force, which is caused by an environmental change. Note that if the constraint force is so small that they are negligible, various noises such as sensing errors and joint bending are amplified. In this study, we assume that the constraint force is constantly large enough to ignore these factors. In the case that these factors are negligible, we should calculate a magnitude of the force smaller than a predefined threshold as zero value. The action $\boldsymbol{a}_t \in \mathbb{R}^3$ is defined as an operation that modifies the direction of motion. Given $\boldsymbol{s}_t$ and $\boldsymbol{a}_t$, the motion direction is updated using the following equation:

$$\boldsymbol{d}_{t+1} = \frac{\boldsymbol{d}_t + \boldsymbol{a}_t}{\|\boldsymbol{d}_t + \boldsymbol{a}_t\|_2} \tag{1}$$

When the object tries to move in the inadmissible direction, the constraint force is exerted on the object. The policy should modify the motion direction toward this force direction such that the force is reduced. As shown in Figure 3, the update of the motion direction by the optimal policy guarantees the adjustment of $\|\boldsymbol{F}\|_2$ resulting from the interaction between the object and the constraint. Thus, the motion direction can be appropriately modified using the force direction. Note that the direction of the constraint force can be obtained under Assumption 3, where a friction in the joint mechanism is sufficiently weaker than the constraint force.

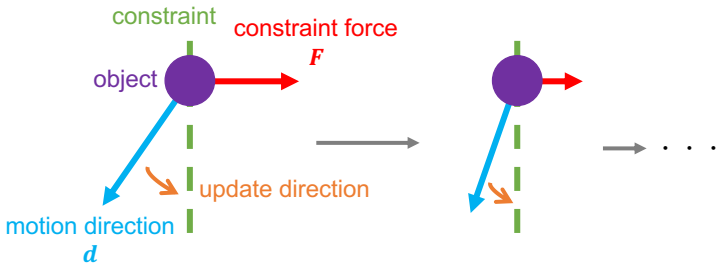

Figure 3. Updating motion direction using the constraint-aware policy. The purple circle and green line represent an object and its constraint, respectively. When the object tries to move in the inadmissible direction, the constraint force is exerted on the object. The motion direction is modified toward this force direction such that the force is reduced.

3.3.3. Reward

We train the constraint-aware policy to estimate the motion direction of the robot hand. To train the optimal policy, we should set an appropriate reward function based on the constraint. Thus, we consider the case that the motion direction is not along the constraint (Figure 3). In compliant manipulations with both the prismatic and revolute joints, if the robot hand does not move along the constraint, the constraint force is exerted by the physical constraint on the object. This force is minimized when the motion direction is along the constraint. Thus, we propose the reward r_t represented by the constraint force $\|\boldsymbol{F}_t\|_2$:

$$r_t = -\|\boldsymbol{F}_t\|_2 \tag{2}$$

3.4. Technical Details of Satisfying the Single-System Condition When Applying the Policy to a Robot

The constraint-aware policy is trained and executed under the assumption of the single-system condition. To satisfy the single-system condition, the relative position and orientation between the robot hand and an object must be maintained. Two main challenges to satisfy this condition are identified: fingertip slipping and lack of contact between the robot and object.

3.4.1. Avoidance of Fingertip Slipping

A violation of the single-system condition can occur if a large impulse force causes the robot's fingertips to slip on the manipulated object. This large impulse force is mainly caused in case that the robot hand tries to move in the inadmissible direction by the large amount of translation. Thus, to prevent the large impulse force, we implement the robot control system so that the robot hand moves slowly. Moreover, fingertip slipping is likely to occur if the hand orientation remains constant during manipulation of a revolute joint where the orientation of the manipulated object changes. To avoid the slipping, we change the hand orientation based on the change in the motion direction, as follows. We define q_t as the quaternion representing the hand orientation in the world coordinate system at time t; then, q_{t+1} can be calculated using the following equation:

$$q_{t+1} = \Delta q_t \otimes q_t \tag{3}$$

where Δq_t represents the quaternion rotating the angle between $\boldsymbol{d}_t$ and $\boldsymbol{d}_{t+1}$ around the outer product of $\boldsymbol{d}_t$ and $\boldsymbol{d}_{t+1}$.

This strategy does not necessarily guarantee a change in the orientation of the hand completely in conjunction with the orientation of the object, and can be adopted only in case the relative orientation between the hand and manipulated object is not strictly fixed. An example case is door opening with a lazy closure, which is one of the grasps [37], as shown in Figure 4. Using the lazy closure, the contact regions remain constant and stable

manipulation is ensured while opening the door, even though the relative orientation between the hand and manipulated object is not strictly fixed. However, when the relative orientation between the hand and manipulated object is strictly fixed, such as handle rotating, a more precise method to change the hand orientation is required. Thus, we prepare an additional policy to maintain the single-system condition for this case. Further details are provided in Appendix A.

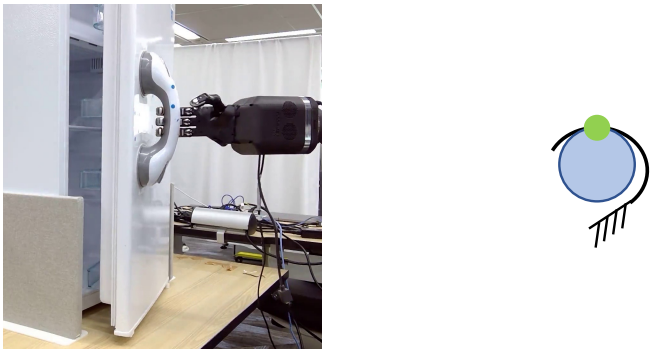

Figure 4. Door opening with "Lazy-closure". A photograph of the actual manipulation is shown on the left. The right of the figure shows a diagrammatic representation of a robot grasping a handle with a lazy closure, where the blue and green circles indicate the handle and contact points, respectively, and the black arc is the gripper.

3.4.2. Guarantee of Hand–Object Contact

The manipulated object and robot hand must be in contact throughout the manipulation to maintain the single-system condition. Contact is guaranteed if a non-zero constraint force is measured by a sensor on the wrist of the robot. Thus, the contact condition is ensured by applying a constraint force at the beginning of the manipulation and maintaining it throughout the manipulation. Specifically, the constraint force $\boldsymbol{F}$ fed into the policy is defined as the raw force value $\boldsymbol{F}_s$ offset by the force $\boldsymbol{F}_d$ (i.e., $\boldsymbol{F} = \boldsymbol{F}_s - \boldsymbol{F}_d$).

The displacement of the hand is classified into admissible or inadmissible directions between the robot hand and object. If the hand moves along the inadmissible direction, the hand collides with the object. In this case, the single-system condition is kept. If the estimated displacement $\boldsymbol{d}$ is out of inadmissible directions between the hand and object, the hand goes away from the object and the single-system condition is broken. In this study, we assume that the estimated displacement is always within the inadmissible directions between the robot hand and object.

4. Learning-from-Observation System

Compliant manipulation is executed by combining our constraint-aware policy with the Learning-from-Observation (LfO), a system in which a human provides manipulation instructions to a robot through a one-shot demonstration [9,10]. In this study, the physical constraint, workplane, and initial motion direction are obtained from a human demonstration for compliant manipulation. Using this system, we can satisfy Assumption 4, i.e., the workplane can be determined by leveraging the demonstration. This section describes the details of the LfO system applied in this study.

As shown in Figure 5, the LfO system consists of two phases: the demonstration phase and execution phase. The demonstration phase involves the LfO system obtaining a sequence of tasks from a human demonstration and assigning skill parameters to each task. During the execution phase, the system decodes the skill parameters into the execution commands.

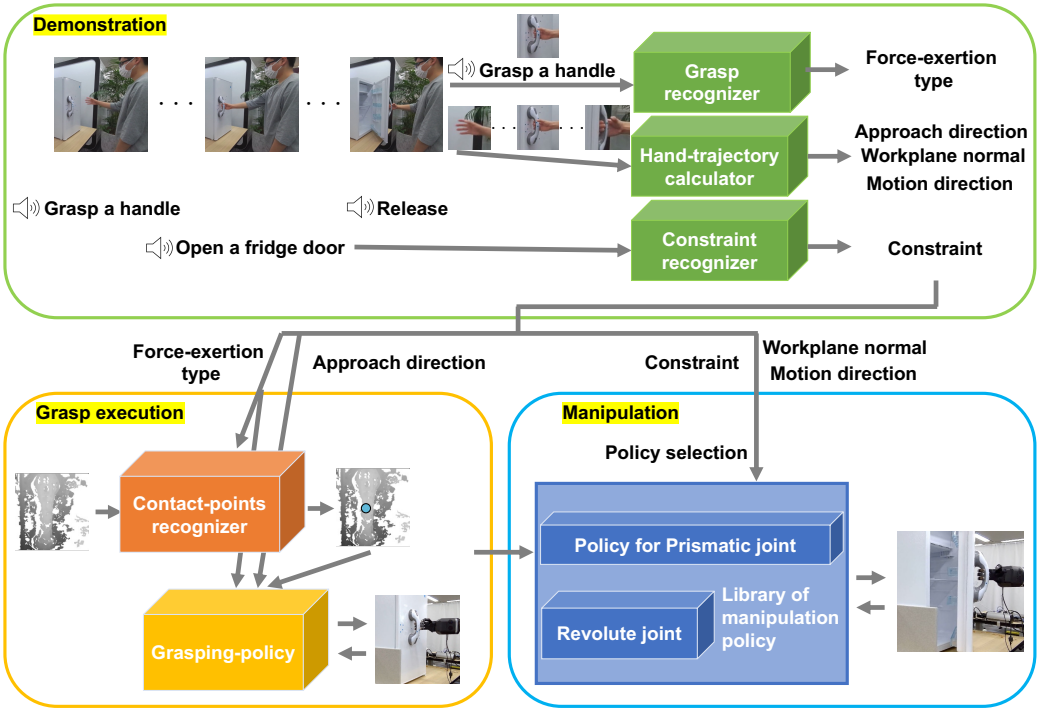

Figure 5. Flow of the LfO system combined with constraint-aware policy.

In the demonstration phase, a human demonstration is encoded into a sequence of tasks using skill parameters [10]. The demonstration consists of an RGBD image sequence of a one-shot human demonstration and verbal instructions. In this study, the human demonstration is decomposed into several tasks, including the grasping and compliant manipulation within physical constraints (prismatic or revolute joint). The skill parameters of grasp and manipulation are also determined from the image sequence and verbal instructions.

For grasping, the skill parameters include the force exertion type and approach direction appropriate for the task situation [37]. A convolutional neural network (CNN)-based classifier (grasp recognizer in Figure 5) recognizes one of the four force exertion types based on the human hand image at the moment of grasp and the name of the object [38]. Similar hand shapes can be recognized as different force exertion types using the name of the object. The approach direction is calculated from the trajectory of the human hand in the demonstration (hand trajectory calculator in Figure 5).

The physical constraint is determined from the verbal instruction (constraint recognizer in Figure 5). For example, the verbal instruction of "open a fridge door" is associated with a revolute joint. For compliant manipulation of a prismatic joint, the skill parameters include the workplane normal and initial motion direction. Meanwhile, the skill parameters of compliant manipulation for a revolute joint include the rotation radius, in addition to the workplane normal and initial motion direction. These parameters are calculated by the hand trajectory calculator. The workplane normal and rotation radii are calculated using plane fitting and circular fitting, respectively.

In the execution phase, the robot executes the target task sequence by first grasping an object and then manipulating it. In the grasping, a contact point recognizer and grasping policy are selected based on the force exertion type obtained in the demonstration phase [37]. The recognizer and policy are previously trained for each force exertion type. The contact point recognizer has a simple CNN structure, where the input is the depth image of the target object and the output is the contact points to be grasped. The detected contact points are passed on to the grasping policy, and the grasp is executed.

In the manipulation, a manipulation policy is executed. The manipulation policy is selected based on the constraints obtained in the demonstration phase. In the task

involving the prismatic or revolute joints, the constraint-aware policy is applied. Note that, as described in Section 3.4, in a task with a revolute joint, the hand orientation is changed to maintain the single-system condition because the orientation of the manipulated object changes during the manipulation. Therefore, the constraint type (prismatic or revolute) must be determined prior to manipulation.

5. Experiment

We evaluated the performance of the proposed constraint-aware policy in the presence of errors in motion direction. We also confirmed the generalization capability of our policy for manipulations with a common constraint. In addition, we evaluated the feasibility of executing our policy and the LfO system on a physical robot. These evaluation processes are described in more detail below.

5.1. Setup

The training environment was implemented using PyBullet simulator [36] and the policy was trained using Microsoft Bonsai, a framework for RL (https://www.microsoft.com/en-us/ai/autonomous-systems-project-bonsai, accessed on 26 December 2023). The episode length of the training environment was set to five timesteps ($T = 5$). To simulate the uncertainty in the sensors, Gaussian noise was added to the observed force and motion direction at the first timestep. The proximal policy optimization (PPO) algorithm [39] was used to train the policy. Batch size and learning rate were set to 6000 and 5×10^{-5}, respectively. The policy π_θ is parameterized by a multilayer perceptron with two 256-dimensional hidden layers. A hyperbolic tangent (*tanh*) was used as the activation function as in [39].

The learned policy was tested using PyBullet simulator. The motion direction was updated every 100 ms in the control loop, and the robot hand was moved by 1 cm along the motion direction in each timestep. Each test started with the robot hand grasping the object, which was achieved using another RL policy [37].

For the physical robot experiments, we utilized a Nextage (https://nextage.kawadarobot.co.jp/, accessed on 26 December 2023) robot with six degrees of freedom in its arms. A four-fingered robot hand, the Shadow Dexterous Hand Lite (https://www.shadowrobot.com/dexterous-hand-series/, accessed on 26 December 2023), was attached to the robot. The Leptrino FFS series (https://www.leptrino.co.jp/product/6axis-force-sensor, accessed on 26 December 2023) was utilized as the force–torque sensor and attached between the manipulator and robot hand, as shown in Figure 6.

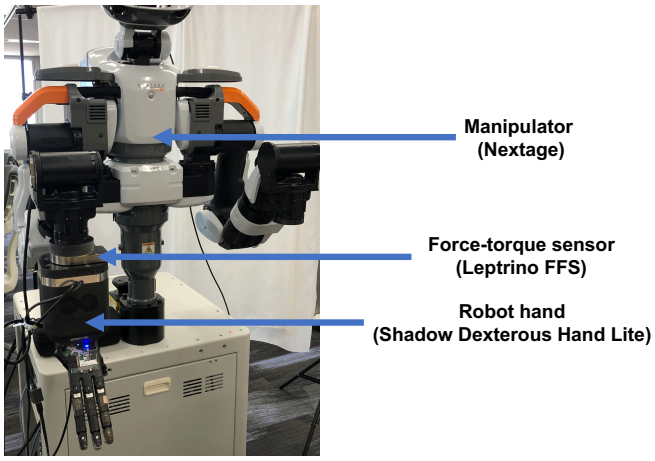

Figure 6. Robot setup, with force–torque sensor attached between the manipulator and robot hand.

5.2. Training in Simulation

The policy was trained in the simulation environment consisting of the object and prismatic joint. The episode reward obtained by the RL agent increased as the training progressed, and the training was completed when the rewards converged (Figure 7).

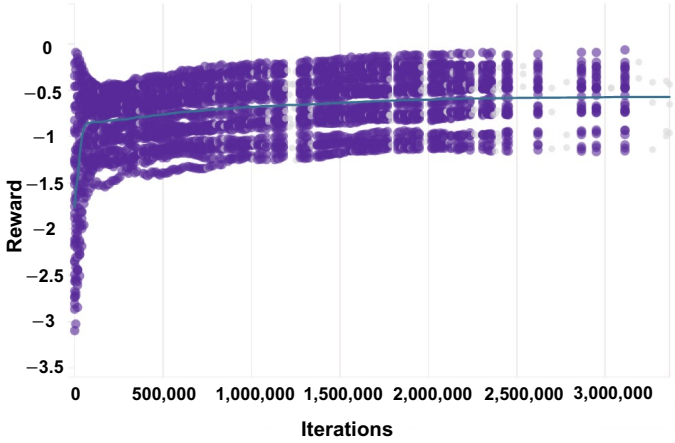

Figure 7. Learning curve of the constraint-aware policy. The blue line and dots are the mean reward of multiple episodes and reward of each episode, respectively. The purple dots represent the reward when the policy is saved.

5.3. Policy Performance in Presence of Motion Direction Errors

A simulated drawer-opening environment was used to evaluate the performance of the proposed policy when the policy faced an error in the motion direction. The drawer was constrained by a prismatic joint, and the episode was considered completed when the drawer had been moved by 25 cm. The handle of the drawer was grasped using a lazy closure.

The results are presented in Figure 8, where the initial motion direction was set with a 30° (Figure 8A) or −30° (Figure 8B) offset from the admissible constraint direction. In both cases, the drawer opening was successfully executed. The curves represent the change in the relative angle between the admissible constraint direction and the current motion direction. The angles converged to near 0°. This result indicates that the proposed constraint-aware policy could estimate the motion direction from the direction of the constraint force.

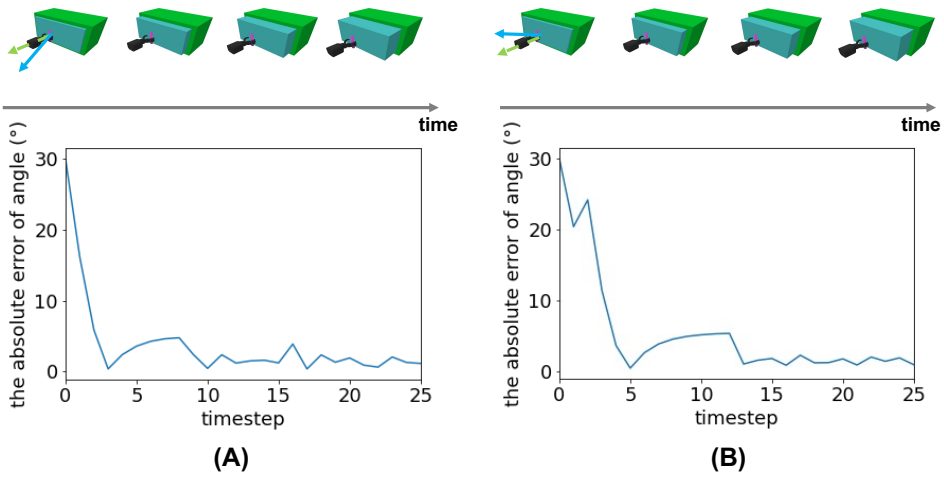

Figure 8. Policy performance in the presence of motion direction error. (**A**) The initial motion direction was set with a 30° offset from the constraint direction. (**B**) Initial motion direction was set with a −30° offset from the constraint direction. The upper panel shows the resulting simulated drawer opening. The lower panel shows the change in the relative angle between the admissible constraint direction (green arrow) and the current motion direction (blue arrow).

5.4. Comparison of Proposed and Classical Controller for Various Manipulations

To evaluate the generalization capability of the proposed constraint-aware policy for various manipulations, we compared it with a state-of-the-art classical controller [7]. Our constraint-aware policy and classical controller were executed on three manipulations with a prismatic joint: (A) drawer opening, (B) plate sliding, and (C) pole pulling. These manipulations were selected because they require different force exertion types for grasp, such as active force, passive force, and lazy closure. These force exertion types cover the types that need no regrasping [37]. In this experiment, the initial motion direction was set with an offset ranging from -30° to 30° in increments of 5° from the constraint direction. We manually tuned the control parameters for drawer opening and used the same parameters for plate sliding and pole pulling.

The results are shown in Table 1. Ours could be successfully executed in all trials for three manipulations. The classical controller could be successfully executed in all trials for drawer opening, while the controller failed the execution for plate sliding and pole pulling, which are not used for the parameter tuning. This result shows that the constraint-aware policy is more generalized for the three manipulations than the classical controller.

Table 1. The comparison of the number of successful trials using our constraint-aware policy (Ours) and the classical controller [7] (Classical) for the three manipulations: drawer opening, plate sliding, pole pulling.

	Drawer Opening	**Plate Sliding**	**Pole Pulling**
Classical	13/13	0/13	0/13
Ours	13/13	13/13	13/13

The example results of the classical controller are shown in Figure 9 (Classical-A, Classical-B, and Classical-C). The initial motion direction was set with a -30° offset from the constraint direction, similar to the conditions reported in Section 5.3. The controller succeeded in drawer opening but not in plate sliding or pole pulling. Since the estimated motion direction overshot in plate sliding and pole pulling, the large force was exerted on the robot finger. As a result, the robot hand could not maintain its grasp and failed to manipulate the objects. This is because the parameters were tuned for the magnitude of the sensed force, which differs according to the grasp. For example, the magnitude changes depending on the degree of joint flexion caused by the object collision with the finger. One of the factors that affects the degree is the values of joint commands, which differ according to the grasp. In practice, the magnitude varies depending not only on the grasp but also on the friction coefficient between the hand and object, object weight, damping coefficient of the finger joints, and sensor noise. Although we could tune the parameters for three manipulations, expert knowledge is necessary for the tuning. In addition, we should obtain multiple environments in advance, whereas the network parameters of our policy could be learned and the single environment is prepared for the training.

The example results of our constraint-aware policy are shown in Figure 10 (Ours-A, Ours-B, and Ours-C). Unlike the classical controller, our constraint-aware policy succeeded in all three manipulations. This is because the utilized state includes the normalized force instead of the raw force, which is not robust to the change in the environment and force exertion type. Using the normalized force makes the policy robust to changes in grasp.

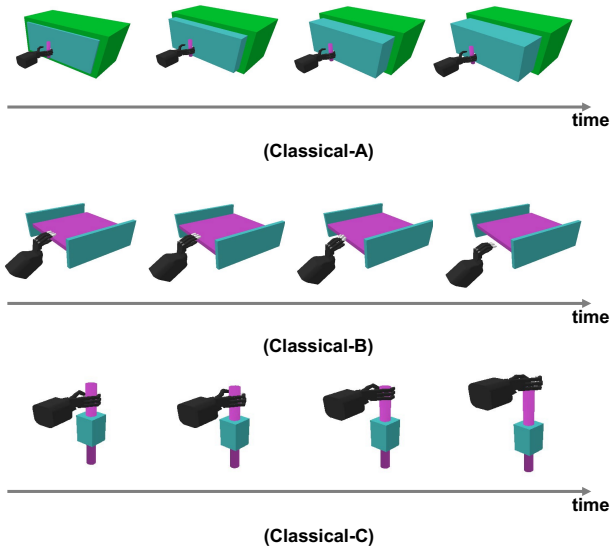

Figure 9. Execution of three manipulations using the classical controller [7]. (**A**): Drawer opening, (**B**): plate sliding, (**C**): pole pulling.

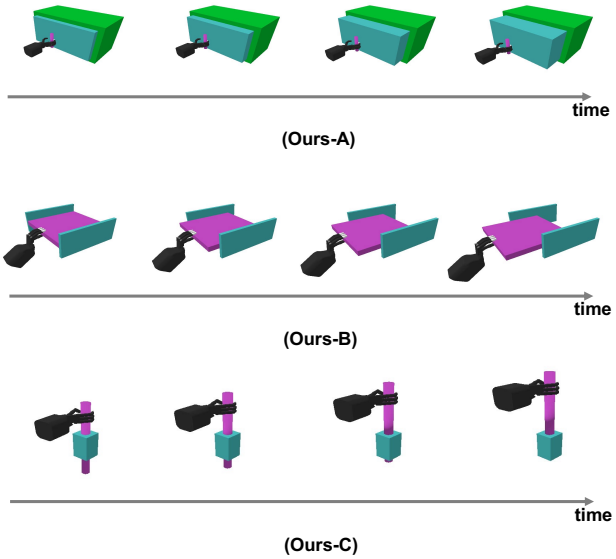

Figure 10. Execution of three manipulations using our constraint-aware policy. (**A**): Drawer opening, (**B**): plate sliding, (**C**): pole pulling.

5.5. Policy Performance for Manipulations with a Revolute Joint

The proposed constraint-aware policy was executed in two different manipulations involving a revolute joint: door opening and handle rotating. The initial motion direction was set with a 15° offset from the constraint direction. Door opening and handle rotating were executed with a lazy and passive force closure, respectively.

The results are shown in Figure 11, demonstrating that the proposed policy could appropriately change the motion direction. Thus, our policy can be executed for manipulations with both prismatic and revolute joints under the single-system condition. In addition, the constraint force is directed from the handle to the rotation center, even when the rotation radius differs; thus, our policy can be adopted for manipulations with varying rotation radii.

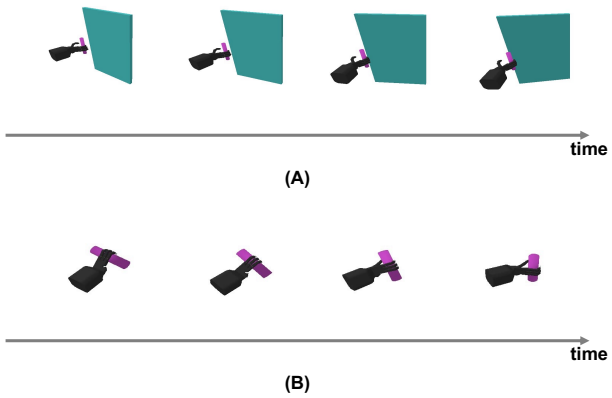

Figure 11. Execution of two manipulations with our constraint-aware policy. (**A**): Door opening, (**B**): handle rotating.

5.6. Compliant Manipulation on a Physical Robot

As mentioned prior, we combined our constraint-aware policy with the LfO system and executed it on a physical robot. In this method, the constraint was recognized from verbal instructions. It is important to identify the constraint because this is utilized to determine whether the hand rotates in conjunction with the manipulated object. In addition, the workplane and initial motion direction were determined. The grasp–manipulation–release sequence could be executed by incorporating the constraint-aware policy into such an LfO system in the real world.

Figure 12 shows the successful execution of the three manipulations, (A) drawer opening, (B) door opening, and (C) handle rotating, in the real world using our constraint-aware policy and the LfO system. Our policy uses a normalized force rather than a raw force, thereby reducing the gap between simulation and reality. Consequently, our policy can be applied to the real world without additional training.

The left side of Figure 13 shows the coordinate system used during the manipulation, while the upper-right side illustrates the change in the relative angle between the admissible motion direction $(-1, 0, 0)$ and the estimated motion direction. The lower-right chart in Figure 13 illustrates the change in the magnitude of the force obtained by the wrist force–torque sensor. These results indicate that the angle and the magnitude of the force were being reduced during execution of the drawer opening.

In Figure 14, the upper-right chart illustrates the transition of the index fingertip position, motion direction, and force direction during the door opening, while the upper-left panel shows the coordinate system used during execution, where the origin was the fingertip position of the robot's index finger at the beginning of the manipulation. The lower part of Figure 14 shows the relative angle between the motion direction and initial motion direction $(-1, 0, 0)$. It is evident that the motion direction changed based on the observed force direction, resulting in the successful execution of the door opening, as shown in the upper-right panel of Figure 14. It was observed that the angle between the initial and actual motion directions gradually increased from the lower part of Figure 14, as expected.

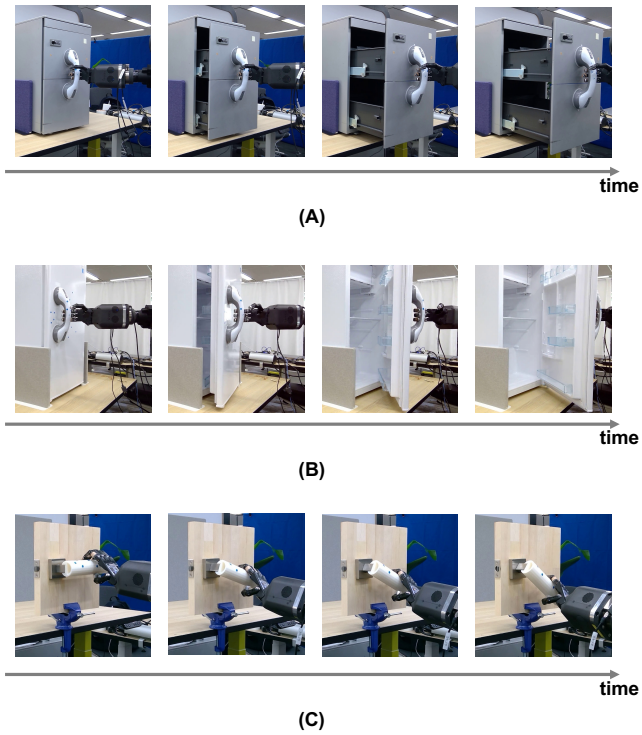

Figure 12. Applying the proposed constraint-aware policy for three manipulations using a physical robot: (**A**) drawer opening, (**B**) door opening, (**C**) handle rotating.

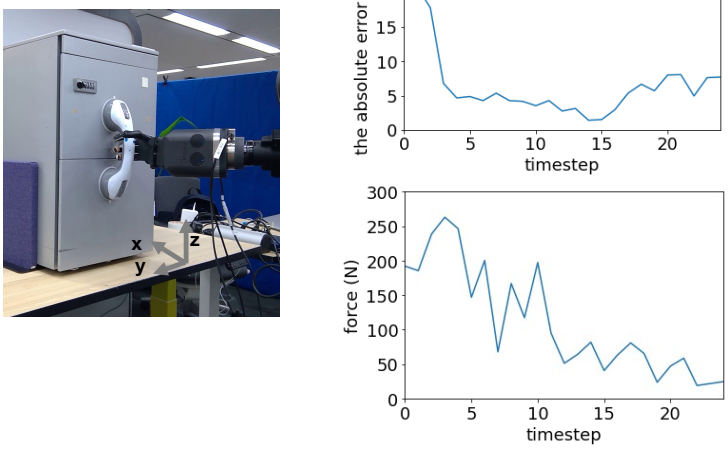

Figure 13. Execution of drawer opening using proposed constraint-aware policy. **Upper left**: coordinate system. **Upper right**: change in relative angle between the estimated motion direction and admissible motion direction $(-1, 0, 0)$. **Lower right**: the change in the magnitude of force recorded by the wrist force–torque sensor.

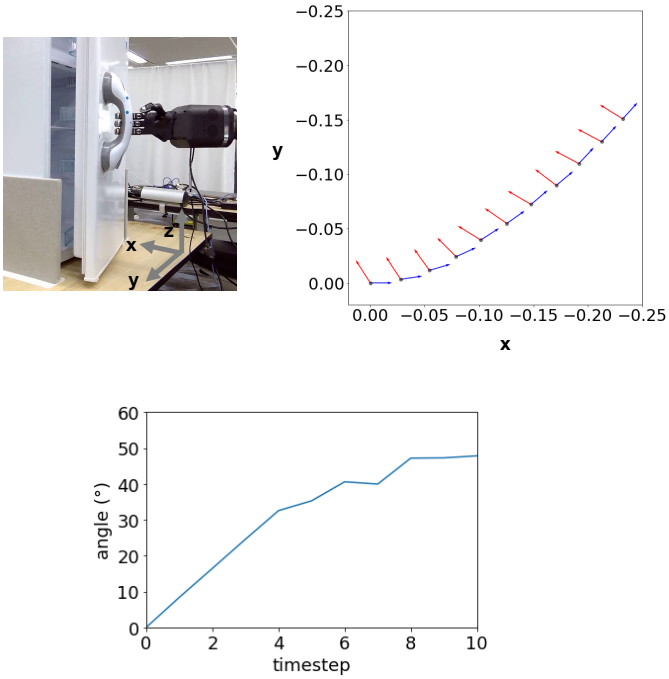

Figure 14. Execution of door opening using proposed constraint-aware policy. **Upper-left**: coordinate system with origin on the index fingertip. **Upper right**: transition of the index fingertip position (black circle), motion direction (blue arrow), and force direction (red arrow) in meters. **Lower**: relative angle between initial motion direction $(-1, 0, 0)$ and the motion direction.

6. Discussion

6.1. Summary of the Experiments

We propose the constraint-aware policy, which is trained using the direction of the constraint force exerted on the object and generalized to various unseen manipulations. In this experiment, we investigated the effectiveness of our policy for manipulations with a prismatic or revolute joint. The results revealed that our policy succeeded in the execution of various manipulations: drawer opening, plate sliding, and pole pulling, whereas the classical controller [7] failed. Although the environment and reward are simple, the policy is generalized. In addition, our policy succeeded in door opening and handle rotating. Finally, our policy could be executed on a physical robot without additional training. These results suggest that our policy is generalized to manipulations with either a prismatic or revolute joint. There is a possibility that our policy can be applied to more manipulations with these joints.

In terms of parameter-tuning cost, our method has an advantage compared to the classical controllers [7,24]. The classical controllers require manual tuning of control parameters, whereas the parameters of our policy can be tuned through the training. In terms of training cost, our method needs a single environment compared to other policies trained by reinforcement learning [5,32,33]. These methods prepare environments of target manipulation for the training (in this study, the number of environments is five), whereas our policy can be trained under the one simple single environment, which includes only a constraint and composite body. This is a benefit of the policy design based on the common characteristic of the constraint force within a manipulation group.

6.2. Limitations

6.2.1. Violation of Single-System Condition

The proposed constraint-aware policy could be implemented under the single-system condition. One violation example of the single-system condition is slipping between the fingertip and manipulated object. This slipping can occur owing to the large estimation

noise of the motion direction, rotation axis, and rotation radius. These issues cannot be addressed by our policy alone. A possible solution is to design an additional policy for maintaining contact positions by utilizing dexterous finger motions that depend on the force exerted on the fingertip. To implement this additional policy, tactile sensors are required. This will be a subject of future research.

6.2.2. Normalized Force

We assume that the inertial force and friction in the joint mechanism are weaker than the constraint force and negligible. In our experiment, this assumption was satisfied and our method could be adopted. However, there is a case that this assumption is not met. For example, the case is that the estimation error of the motion direction is near 0°. In this case, the weak inertial force and friction are amplified when normalizing them. This causes a system instability. To avoid the instability, we should calculate a magnitude of the force smaller than a predefined threshold as zero values. The way to define the threshold will be a subject of future research.

6.3. Future Directions

6.3.1. Hardware-Level Reusability

In this study, we designed a constraint-aware policy that can be applied to robot hands without considering hardware specifications assuming the single-system condition. In contrast to conventional strategies, our policy was designed to be both manipulation-agnostic and hardware-independent. When using new hardware, robot programmers typically must modify software, which can be time-consuming. To address this issue, some software programs enabling reusability have already been developed [40]. The work in this study represents another contribution to this field; using our constraint-aware policy, hardware-level reusability can be achieved. To demonstrate reusability, future studies will validate the hardware-level reusability of the proposed policy.

6.3.2. Constraint-Aware Policy for other Constraints

Many manipulations in a household environment can be grouped based on constraints [8]. This taxonomy includes manipulation groups with prismatic and revolute joints as well as those with other constraints. One solution for achieving various manipulations in a household environment is to design a policy for each manipulation group. For achieving various household manipulations, our concept of the constraint-aware policy can be applied to other constraints. It should be effective to consider various manipulations with the same constraint as one manipulation group and design a policy with an awareness of the constraint.

7. Conclusions

In this study, we proposed a constraint-aware policy that can be applied to various manipulations with either a prismatic or revolute joint. We designed a training environment and a reward function to train the policy based on these constraints. The experimental results showed that the single policy could be executed on three manipulations with a prismatic joint (drawer opening, plate sliding, and pole pulling), even when an estimation error in the motion direction was applied in the simulation. Unlike the classical controller, our policy achieved robust execution against environmental changes. In addition, we could execute our policy on two manipulations with a revolute joint (door opening and handle rotating). Furthermore, three manipulations, drawer opening, door opening, and handle rotating, were successfully executed on an actual robot without additional training.

Although our policy was trained in the simple environment, our policy could be executed successfully on different manipulations. Previous reinforcement learning (RL) methods specially designed the environment and reward for each target manipulation, whereas our policy was widely applicable to various assumed situations. Thus, we successfully designed a policy generalized to manipulations constrained by either a prismatic

or revolute joint based on the constraint force, which is a common characteristic between such manipulations.

Toward a robot system capable of executing a wide range of manipulations, it is crucial to design a generalized policy for each manipulation group. Household manipulations can be categorized according to their physical constraints [8]. The key to the generalized policies is to design an environment and reward focusing on a common characteristic within each group. This study validated the concept of a constraint-aware policy for either a prismatic or revolute joint, which are fundamental in considering physical constraints. We believe this study is the first step towards realizing the generalized household robot.

Author Contributions: All authors contributed to the study conception and design. Methodology: D.S.; Software: D.S.; Validation: D.S.; Formal analysis: D.S.; Writing—Original Draft: D.S.; Writing—Review and Editing: All authors; Visualization: D.S.; Supervision: H.K. and K.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to other ongoing research.

Conflicts of Interest: Kazuhiro Sasabuchi, Naoki Wake, Atsushi Kanehira, Jun Takamatsu, Katsushi Ikeuchi are employed by the company Microsoft. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Additional Policy

We observed that the constraint-aware policy alone was unable to conduct handle rotating with passive force closure (Figure A1). This failure occurred because it was impossible to maintain the single-system condition. As described in the main text, when there is a change in the relative orientation between the robot hand and the manipulated object, torque is generated, causing slippage. The relative orientation between the hand and the manipulated object is strictly fixed, so the torque was generated due to a change in the relative orientation. Thus, an additional policy was required that would enable the hand to rotate in conjunction with the manipulated object.

Figure A1. The failure result using the constraint-aware policy.

An additional policy was developed to appropriately rotate the hand around the center of the contact points to the handle. The rotation axis corresponds to the normal of the workplane. This additional policy estimated the suitable amount of rotation w at each time step by the following process using the torque around the rotation axis τ.

1. Rotate the hand by the current estimation of w.
2. Decide the adjustment Δw as follows ($\beta > 0$):

$$\begin{cases} \Delta w = 0 & (\|\tau\| \leq \alpha) \\ \Delta w = \beta & (\tau > \alpha) \\ \Delta w = -\beta & (\tau < \alpha) \end{cases}$$

3. Update w to $w + \Delta w$.

The initial value of w is calculated using $w = \frac{v}{r}$, where r is the rotation radius obtained from human demonstration and v is the amount of translation in each time step. The policy can calculate the excess or deficiency between w and the suitable amount of rotation for

one-step translation. The constraint-aware and additional policies are combined to execute the handle rotating (Figure A2). If τ is greater than α after the robot hand is translated and rotated simultaneously, the additional policy is implemented until τ is smaller than α to minimize forces other than the constraint force. Otherwise, the constraint-aware policy is implemented solely.

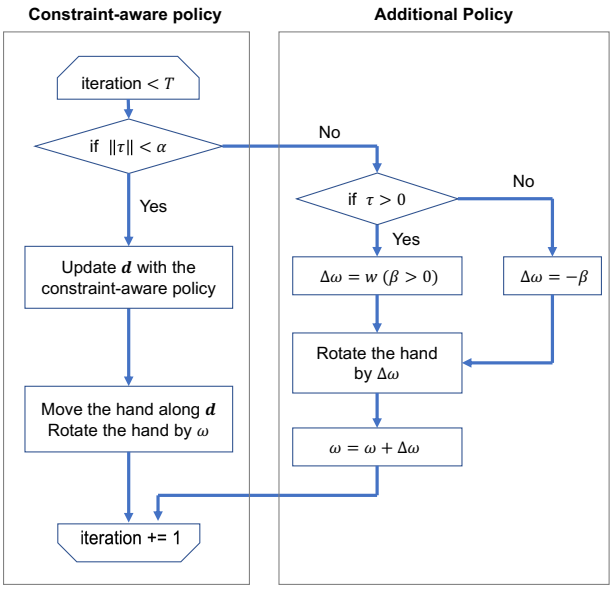

Figure A2. Combined policy for the case in which the hand cannot rotate freely around the rotation axis. The generalized policy is executed if the torque around the rotation axis $\|\tau\|$ is smaller than the threshold β. Otherwise, an additional policy is executed. T is the episode length.

Figure 11B shows the successful result of handle rotating using the combined policy. In the experiment, we set $\alpha = 10, \beta = 1^\circ$. The single-system condition was maintained by rotating the hand appropriately based on the torque. This result demonstrates that our constraint-aware policy, combined with the additional policy, can successfully execute handle rotating while maintaining the single-system condition. Although the additional policy requires manual tuning of the parameters to minimize the torque, these parameters have a higher interpretability for tuning than the control parameters of the classical controller.

References

1. Mason, M.T. Compliance and force control for computer controlled manipulators. *IEEE Trans. Syst. Man Cybern.* **1981**, *11*, 418–432. [CrossRef]
2. Yahya, A.; Li, A.; Kalakrishnan, M.; Chebotar, Y.; Levine, S. Collective robot reinforcement learning with distributed asynchronous guided policy search. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 79–86.
3. Gu, S.; Holly, E.; Lillicrap, T.; Levine, S. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3389–3396.
4. Rajeswaran, A.; Kumar, V.; Gupta, A.; Vezzani, G.; Schulman, J.; Todorov, E.; Levine, S. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In Proceedings of the Robotics: Science and Systems (RSS), Pittsburgh, PA, USA, 26–30 June 2018.
5. Urakami, Y.; Hodgkinson, A.; Carlin, C.; Leu, R.; Rigazio, L.; Abbeel, P. Doorgym: A scalable door opening environment and baseline agent. *arXiv* **2019**, arXiv:1908.01887.
6. Sun, Y.; Zhang, L.; Ma, O. Force-Vision Sensor Fusion Improves Learning-Based Approach for Self-Closing Door Pulling. *IEEE Access* **2021**, *9*, 137188–137197. [CrossRef]
7. Karayiannidis, Y.; Smith, C.; Barrientos, F.E.V.; Ögren, P.; Kragic, D. An adaptive control approach for opening doors and drawers under uncertainties. *IEEE Trans. Robot.* **2016**, *32*, 161–175. [CrossRef]
8. Ikeuchi, K.; Wake, N.; Arakawa, R.; Sasabuchi, K.; Takamatsu, J. Semantic constraints to represent common sense required in household actions for multi-modal learning-from-observation robot. *arXiv* **2021**, arXiv:2103.02201.

9. Ikeuchi, K.; Suehiro, T. Toward an assembly plan from observation. I. Task recognition with polyhedral objects. *IEEE Trans. Robot. Autom.* **1994**, *10*, 368–385. [CrossRef]
10. Wake, N.; Kanehira, A.; Sasabuchi, K.; Takamatsu, J.; Ikeuchi, K. Interactive Learning-from-Observation through multimodal human demonstration. *arXiv* **2022**, arXiv:2212.10787.
11. Nagatani, K.; Yuta, S. An experiment on opening-door-behavior by an autonomous mobile robot with a manipulator. In Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots, Pittsburgh, PA, USA, 5–9 August 1995; Volume 2, pp. 45–50. [CrossRef]
12. Klingbeil, E.; Saxena, A.; Ng, A.Y. Learning to open new doors. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 2751–2757. [CrossRef]
13. Abbatematteo, B.; Tellex, S.; Konidaris, G. Learning to Generalize Kinematic Models to Novel Objects. In *Proceedings of the the Conference on Robot Learning, PMLR, Virtual Event, 30 October–1 November 2020*; Kaelbling, L.P., Kragic, D., Sugiura, K., Eds.; Proceedings of Machine Learning Research; 2020; Volume 100, pp. 1289–1299.
14. Rühr, T.; Sturm, J.; Pangercic, D.; Beetz, M.; Cremers, D. A generalized framework for opening doors and drawers in kitchen environments. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 3852–3858. [CrossRef]
15. Li, X.; Wang, H.; Yi, L.; Guibas, L.J.; Abbott, A.L.; Song, S. Category-Level Articulated Object Pose Estimation. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 3703–3712. [CrossRef]
16. Arduengo, M.; Torras, C.; Sentis, L. Robust and adaptive door operation with a mobile robot. *Intell. Serv. Robot.* **2021**, *14*, 409–425. [CrossRef]
17. Liu, L.; Xue, H.; Xu, W.; Fu, H.; Lu, C. Toward Real-World Category-Level Articulation Pose Estimation. *IEEE Trans. Image Process.* **2022**, *31*, 1072–1083. [CrossRef] [PubMed]
18. Jain, A.; Lioutikov, R.; Chuck, C.; Niekum, S. ScrewNet: Category-Independent Articulation Model Estimation From Depth Images Using Screw Theory. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 13670–13677. [CrossRef]
19. Eisner, B.; Zhang, H.; Held, D. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv* **2022**, arXiv:2205.04382.
20. Wei, F.; Chabra, R.; Ma, L.; Lassner, C.; Zollhöfer, M.; Rusinkiewicz, S.; Sweeney, C.; Newcombe, R.; Slavcheva, M. Self-supervised neural articulated shape and appearance models. In Proceedings of the the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 15816–15826.
21. Kessens, C.C.; Rice, J.B.; Smith, D.C.; Biggs, S.J.; Garcia, R. Utilizing compliance to manipulate doors with unmodeled constraints. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 483–489.
22. Jain, A.; Kemp, C.C. Pulling open doors and drawers: Coordinating an omni-directional base and a compliant arm with equilibrium point control. In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 1807–1814.
23. Niemeyer, G.; Slotine, J.J. A simple strategy for opening an unknown door. In Proceedings of the International Conference on Robotics and Automation, Albuquerque, NM, USA, 25 April 1997; Volume 2, pp. 1448–1453. [CrossRef]
24. Schmid, A.J.; Gorges, N.; Goger, D.; Worn, H. Opening a door with a humanoid robot using multi-sensory tactile feedback. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 285–291.
25. Chung, W.; Rhee, C.; Shim, Y.; Lee, H.; Park, S. Door-Opening Control of a Service Robot Using the Multifingered Robot Hand. *IEEE Trans. Ind. Electron.* **2009**, *56*, 3975–3984. [CrossRef]
26. Karayiannidis, Y.; Smith, C.; Ögren, P.; Kragic, D. Adaptive Force/Velocity control for opening unknown doors1. *IFAC Proc. Vol.* **2012**, *45*, 753–758. [CrossRef]
27. Pilastro, D.; Oboe, R.; Shimono, T. A nonlinear adaptive compliance controller for rehabilitation. *IEEJ J. Ind. Appl.* **2016**, *5*, 123–131. [CrossRef]
28. Corrá, L.; Oboe, R.; Shimono, T. Adaptive optimal control for rehabilitation systems. In Proceedings of the IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October– November 2017; pp. 5197–5202.
29. Pareek, S.; NIsar, H.; Kesavadas, T. AR3n: A Reinforcement Learning-based Assist-As-Needed Controller for Robotic Rehabilitation. *arXiv* **2023**, arXiv:2303.00085.
30. Nair, A.V.; Pong, V.; Dalal, M.; Bahl, S.; Lin, S.; Levine, S. Visual reinforcement learning with imagined goals. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 9209–9220.
31. Yu, T.; Quillen, D.; He, Z.; Julian, R.; Hausman, K.; Finn, C.; Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Proceedings of the Conference on Robot Learning, PMLR, Virtual Event, 30 October–1 November 2020; pp. 1094–1100.
32. Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv* **2022**, arXiv:2212.06817.

33. Reed, S.; Zolna, K.; Parisotto, E.; Colmenarejo, S.G.; Novikov, A.; Barth-Maron, G.; Gimenez, M.; Sulsky, Y.; Kay, J.; Springenberg, J.T.; et al. A generalist agent. *arXiv* **2022**, arXiv:2205.06175.
34. Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv* **2023**, arXiv:2307.15818.
35. Shridhar, M.; Manuelli, L.; Fox, D. Perceiver-actor: A multi-task transformer for robotic manipulation. In Proceedings of the Conference on Robot Learning, PMLR, Atlanta, GA, USA, 6–9 November 2023; pp. 785–799.
36. Coumans, E.; Bai, Y. PyBullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning. 2016–2021. Available online: http://pybullet.org (accessed on 26 December 2023).
37. Saito, D.; Sasabuchi, K.; Wake, N.; Takamatsu, J.; Koike, H.; Ikeuchi, K. Task-grasping from a demonstrated human strategy. In Proceedings of the 2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids), Ginowan, Japan, 28–30 November 2022; pp. 880–887.
38. Wake, N.; Sasabuchi, K.; Ikeuchi, K. Grasp-type recognition leveraging object affordance. *arXiv* **2020**, arXiv:2009.09813.
39. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
40. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.