

Contact-rich $SE(3)$ -Equivariant Robot Manipulation Task Learning via Geometric Impedance Control

JooHwan Seo¹, Nikhil P. S. Prakash¹, Xiang Zhang¹, Changhao Wang¹,
Jongeun Choi^{1,2}, Masayoshi Tomizuka¹, and Roberto Horowitz¹

Abstract—This paper presents a differential geometric control approach that leverages $SE(3)$ group invariance and equivariance to increase transferability in learning robot manipulation tasks that involve interaction with the environment. Specifically, we employ a control law and a learning representation framework that remain invariant under arbitrary $SE(3)$ transformations of the manipulation task definition. Furthermore, the control law and learning representation framework are shown to be $SE(3)$ equivariant when represented relative to the spatial frame. The proposed approach is based on utilizing a recently presented geometric impedance control (GIC) combined with a learning variable impedance control framework, where the gain scheduling policy is trained in a supervised learning fashion from expert demonstrations. A geometrically consistent error vector (GCEV) is fed to a neural network to achieve a gain scheduling policy that remains invariant to arbitrary translation and rotations. A comparison of our proposed control and learning framework with a well-known Cartesian space learning impedance control, equipped with a Cartesian error vector-based gain scheduling policy, confirms the significantly superior learning transferability of our proposed approach. A hardware implementation on a peg-in-hole task is conducted to validate the learning transferability and feasibility of the proposed approach. The simulation and hardware experiment video is posted: <https://sites.google.com/berkeley.edu/equivariant-task-learning/home>

Index Terms—Machine Learning for Robot Control, Compliance and Impedance Control, Learning from Demonstration

I. INTRODUCTION

Learning has become a prevalent method for robots to acquire skills in automated manipulation tasks [1]. Most learning-based approaches are formulated using a Cartesian frame to represent the end-effector workspace. However, Cartesian-based learning formulations lack learning transferability in that trained policies cannot be directly transferred to arbitrarily translated/rotated task descriptions unless extensive additional training is conducted [2], [3].

Manuscript received: August, 26, 2023; Revised November, 8, 2023; Accepted December, 7, 2023.

This paper was recommended for publication by Editor Jens Kober upon evaluation of the Associate Editor and Reviewer's comments.

This research is partially funded by (1) the Tsinghua-Berkeley Shenzhen Institute (TBSI) phase II and (2) the Hong Kong Center for Construction Robotics Limited (HKCRC). Jongeun Choi was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT). (No.RS-2023-00221762 and No.2021R1A2B5B01002620)

¹JooHwan Seo, Nikhil P.S. Prakash, Xiang Zhang, Changhao Wang, Jongeun Choi, Masayoshi Tomizuka, and Roberto Horowitz are with UC Berkeley, Department of Mechanical Engineering. {jooHwan_seo, nikhilps, xiang_zhang_98, changhaowang, tomizuka, horowitz}@berkeley.edu

²Jongeun Choi is with Yonsei University, School of Mechanical Engineering. joungeunchoi@yonsei.ac.kr

Digital Object Identifier (DOI): see top of this page.

Copyright ©2024 IEEE

From the differential geometric perspective, the poor transferability of trained policies can be directly attributed to the absence of $SE(3)$ group equivariance in the Cartesian-based learning framework. For instance, a group transformation (e.g. translation/rotation) of the task definition does not result in a corresponding transformation of trained policy and task execution. As a consequence, group-transformed tasks have to be relearned, requiring a significant increase in training episodes and learning resources.

Recently, in the field of geometric deep learning (GDL) in computer vision applications [4], [5], the symmetry inherent to a group structure within a domain has been exploited and integrated into neural network models to enhance learning transferability and robustness to untrained data, thereby improving sample efficiency. The key properties exploited by GDL are group invariance and equivariance. Invariance refers to a property of a map whose output remains unchanged when a group action transforms its input, while equivariance implies that a map's output is transformed by the representation of the same group action when its input is transformed by the group action.

Robotic manipulator workspaces have a group structure, which is frequently represented by the Special Euclidean group $SE(3)$ [6]. In our previous work [7], we introduced a geometric impedance control (GIC) for robot manipulators that fully incorporates the geometric structure of $SE(3)$. GIC leverages the left-invariance of the distance metric and potential function in $SE(3)$ to design a control law expressed in the body-frame coordinate system.

In this paper, we leverage $SE(3)$ group invariance and equivariance, extensively studied in GDL, to enhance learning transferability and sampling efficiency in contact-rich robotic manipulation tasks. A Peg-in-Hole (PiH) task is used as a testbed for evaluation since it is sensitive to $SE(3)$ transformations (i.e. translation/rotation) and involves contact-rich robot interaction with the environment. A geometric learning variable impedance control is presented that utilizes the GIC in [7] and incorporates a learning variable impedance control framework, where the gain scheduling policy is trained in a supervised learning fashion from expert demonstrations. A geometrically consistent error vector (GCEV) is fed to a neural network to achieve a gain scheduling policy that remains invariant under arbitrary $SE(3)$ transformations. The main contributions of our paper are:

- 1) We propose key components for learning transferability in robotic manipulation tasks under dynamic feedback

control law: left-invariance and a control law formulated in the body-frame coordinate system.

- 2) We provide a theoretical justification for our proposed approach and demonstrate how learning transferability can be achieved through these components.
- 3) We show that the proposed approach is equivariant when described in the spatial frame, as the trained policy and control law are invariant in the body frame.
- 4) We validate the feasibility of the proposed approach through a hardware experiment, including a workflow for data collection and policy training.

II. RELATED WORKS

A. Geometric Deep Learning for Robotics Problem

Geometric Deep Learning (GDL) has found success in image-based data, particularly in medical fields [5], and has also been applied in robotics [8], [9], [10]. These geometric approaches leverage group structures' invariance and equivariance. For instance, SE(2) equivariance in [8] for image-based inputs, while SE(3) equivariance is explored in [9], [10] for point cloud input data. [9] introduces a descriptor field with SE(3) equivariance, improving performance for out-of-distribution inputs, with further enhancement in [10], [11], [12].

Equivariant structures have been pursued in other robotics applications as well [13], [14], [15]. In [13], the translational equivariance is achieved via soft correspondence. [14] attains equivariance on the Euclidean group E(2) through data augmentation, which is data-inefficient. [15] leverages SE(2) equivariance for learning tabletop object manipulation but only within the SE(2) space, without extending equivariance to control inputs.

Equivariant RL approaches have also been proposed in [16], [17], [18] to enhance RL's sample efficiency by leveraging symmetries in the Markov Decision Process (MDP). However, these equivariant RL approaches are generally applicable to simple scenarios without consideration of dynamics or force interaction [16] or are limited to SE(2) or SO(2) scenarios with actions restricted to displacements [17], [18].

To date, the application of GDL in robotics has mainly focused on point cloud or image inputs and simpler task types without force interactions. Additionally, while control methods remain prevalent in robotics research, the equivariance properties of dynamic controls have yet to be fully explored, emphasizing the growing need to integrate GDL concepts with dynamic robot controllers.

B. Peg-in-Hole task and Learning Variable Impedance Control

The Peg-in-Hole (PiH) task is a benchmark problem for force-controlled robotic manipulation tasks [19]. A widely utilized approach to solve a PiH task is the variable impedance control [3], [20], [2], where the controller's impedance gains change depending on the states. These approaches, however, do not take into account the geometric structures of the manipulator. As a result, it is reported in [2] that the success rate of a trained policy dropped significantly under SE(3) transformations, i.e., when the peg position is tilted relative to the orientation used for training. To deal with this issue,

a typical approach is to adopt the domain randomization technique as in [3], where both the initial and goal poses of the end-effector are randomized during the training. In contrast, our proposed approach achieves robustness to out-of-distribution (OOD) goal poses and learning transferability without randomizing goal poses during the training stage. We note that the proposed approach is not constrained to the PiH task, but it can be extended to other manipulation tasks with force interaction, such as surface wiping or pick-and-place.

III. PRELIMINARIES

A. Lie Groups and Manipulator Dynamics

The configuration of the manipulator's end-effector can be defined by its position and orientation, and the configuration manifold lies in the Special Euclidean group $SE(3)$. We can represent the end-effector's configuration frame $\{e\}$ using the following homogeneous matrix g_{se} , to fixed a (inertial) spatial frame $\{s\}$, as follows:

$$g_{se} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \in SE(3), \quad (1)$$

where R is a rotation matrix and $R \in SO(3)$, and $p \in \mathbb{R}^3$. We will drop the subscript s since the spatial coordinate frame can be considered as an identity without loss of generality. In addition, we also drop the subscript e for the current configuration of the end-effector for notational compactness unless specified, i.e., $g_{se} = g$. We also use $g = (R, p)$ for notational compactness.

The Lie algebra of $SE(3)$, $se(3)$ can be represented by

$$\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} \in se(3), \quad \forall \xi = \begin{bmatrix} v \\ \omega \end{bmatrix} \in \mathbb{R}^6, \quad v, \omega \in \mathbb{R}^3, \quad \hat{\omega} \in so(3).$$

For the details of the Lie group for robotic manipulators, we refer to [6], [21]. Note also that we utilize the standard hat-map and vee-map notations as defined in [7].

The velocity of the end-effector relative to its body frame, $V^b \in \mathbb{R}^6$, can be calculated by:

$$V^b = \begin{bmatrix} v^b \\ \omega^b \end{bmatrix} = (g^{-1}\dot{g})^\vee, \quad (2)$$

i.e., $\dot{g} = g\hat{V}^b$. The velocity V^b can also be computed using the body Jacobian matrix $J_b(q)$ via $V^b = J_b(q)\dot{q}$. For the details about $J_b(q)$, we refer to Chap 5.1 of [21].

The dynamic equations of motion for rigid-link robotic manipulators are given by:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = T + T_e, \quad (3)$$

where $M(q) \in \mathbb{R}^{n \times n}$ is the symmetric positive definite inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ is a Coriolis matrix, $G(q) \in \mathbb{R}^n$ is a moment term due to gravity, $T \in \mathbb{R}^n$ is a control input, and $T_e \in \mathbb{R}^n$ is an external disturbance. As in [7], the manipulator dynamics in operational space formulation [22], using the body-frame velocity, is represented as follows:

$$\tilde{M}(q)\dot{V}^b + \tilde{C}(q, \dot{q})V^b + \tilde{G}(q) = \tilde{T} + \tilde{T}_e, \quad \text{where} \quad (4)$$

$$\begin{aligned} \tilde{M}(q) &= J_b(q)^{-T} M(q) J_b(q)^{-1}, \\ \tilde{C}(q, \dot{q}) &= J_b(q)^{-T} (C(q, \dot{q}) - M(q) J_b(q)^{-1} \dot{J}_b(q)) J_b(q)^{-1}, \\ \tilde{G}(q) &= J_b(q)^{-T} G(q), \quad \tilde{T} = J_b(q)^{-T} T, \quad \tilde{T}_e = J_b(q)^{-T} T_e, \end{aligned}$$

where $A^{-T} = (A^{-1})^T$. We will denote $\tilde{M}(q)$ as \tilde{M} , $\tilde{C}(q, \dot{q})$ as \tilde{C} and $\tilde{G}(q)$ as \tilde{G} for the rest of the paper.

B. Geometric Impedance Control Law

We employ the geometric impedance control (GIC) law proposed in [7]. We first note that g denotes the current configuration matrix, with p and R in (1) representing the current position and rotation matrix, i.e., $g = (R, p)$. In a similar way, $g_d = (R_d, p_d)$ denotes the desired current configuration matrix. In a nutshell, a GIC control law $\tilde{T} \in se^*(3)$ in the wrench is given by

$$\tilde{T} = \tilde{M}\dot{V}_d^* + \tilde{C}V_d^* + \tilde{G} - f_G - K_d e_V. \quad (5)$$

where \tilde{C} , and \tilde{G} are matrices in the operational space formulation (4), K_d is symmetric positive definite damping matrix, f_G is the elastic generalized force in $se^*(3)$ and e_V is a velocity error vector, both of them described on the body-frame, which will be described subsequently. The elastic geometric wrench (force) $f_G(g, g_d)$ is given by

$$f_G(g, g_d) = \begin{bmatrix} f_p \\ f_R \end{bmatrix} = \begin{bmatrix} R^T R_d K_p R_d^T (p - p_d) \\ (K_R R_d^T R - R^T R_d K_R)^\vee \end{bmatrix}, \quad (6)$$

where K_p and K_R denote symmetric positive definite stiffness matrices in translation and rotation, respectively.

The current and desired velocity vectors cannot be directly compared, as they lie on different tangent spaces. Therefore, we utilize a vector translation map (Adjoint map) to first translate the desired velocity to the tangent space of the current velocity [23]. The error vector e_V is defined by

$$e_V = V^b - V_d^* = [e_v^T, e_\omega^T]^T$$

$$V_d^* = \text{Ad}_{g_{ed}} V_d^b, \text{ with } \text{Ad}_{g_{ed}} = \begin{bmatrix} R_{ed} & \hat{p}_{ed} R_{ed} \\ 0 & R_{ed} \end{bmatrix}, \quad (7)$$

where V_d^b is a desired velocity in the desired body frame, $\text{Ad}_{g_{ed}} : \mathbb{R}^6 \rightarrow \mathbb{R}^6$ is an Adjoint map, $R_{ed} = R^T R_d$, $p_{ed} = -R^T (p - p_d)$, and V_d^* is a translated desired velocity on the configuration body-frame.

For the PiH task, we assumed that the final desired configuration g_d is given and is time-invariant, i.e., $\dot{g}_d(t) = 0$. Thus, the controller law (5) is modified as

$$\tilde{T} = -f_G - K_d V^b + \tilde{G}, \quad (8)$$

which can be interpreted as a PD control together with gravity compensation. For more details on the GIC, such as derivation and stability properties, we refer to [7] and its references [24], [23]. Here, we define a geometrically consistent error vector (GCEV) e_G , which will be utilized in the learning impedance gains later in this paper. The GCEV $e_G(g, g_d)$ is defined as follows:

$$e_G(g, g_d) = \begin{bmatrix} e_p \\ e_R \end{bmatrix} = \begin{bmatrix} R^T (p - p_d) \\ (R_d^T R - R^T R_d)^\vee \end{bmatrix} \in \mathbb{R}^6. \quad (9)$$

Finally, we note that all the vectors/wrench, such as e_G , e_V , and f_G , are described in the body-frame coordinate unless otherwise specified.

C. Cartesian space Impedance Control

As a benchmark approach for the proposed GIC, we also briefly introduce a Cartesian space Impedance Controller (CIC), which is a currently standard method for impedance

control. In the operational space formulation, correctly representing the rotational dynamics has always received significant interest [25]. In particular, a positional Cartesian error vector e_C widely utilized in CIC can be defined in the following way [26], [27], [28].

$$e_C = [(e_{C,p})^T, (e_{C,R})^T]^T, \quad \text{where} \quad (10)$$

$$e_{C,p} = p - p_d, \quad e_{C,R} = (r_{d_1} \times r_1 + r_{d_2} \times r_2 + r_{d_3} \times r_3),$$

with $R = [r_1, r_2, r_3]$ and $R_d = [r_{d_1}, r_{d_2}, r_{d_3}]$. Utilizing the positional error vector in Cartesian space and considering a fair comparison with the GIC formulation, we will utilize the following CIC formulation for the PiH task.

$$\tilde{T}_C = -K_C e_C - K_{d_C} V^s + \tilde{G}_C, \quad (11)$$

where \tilde{G}_C can be obtained by replacing J_b by J_s in (4), K_{d_C} is a damping matrix for the CIC, and $K_C = \text{blkdiag}(K_{C,p}, K_{C,R})$ with $K_{C,p}$ and $K_{C,R}$ are translational and rotational stiffness matrices, respectively. In addition, J_s denotes a spatial frame Jacobian matrix and $V^s = J_s \dot{q}$. To implement control law (8) and (11), the wrenches should first be converted to joint torque T in (3) by multiplying corresponding Jacobian matrices, i.e., $T = J_b^T \tilde{T}$ and $T = J_s^T \tilde{T}_C$, respectively.

We highlight the differences between the GIC and the CIC in the following remark:

Remark 1. Differences between GIC and CIC

The main differences between the GIC and the CIC are twofold:

1. CIC deals with translational and rotational dynamics separately, but GIC deals with translational and rotational dynamics as a unified entity, using the geometric elastic wrench $f_G(g, g_d)$ in (6).
2. GIC utilizes a body-frame Jacobian J_b while CIC utilizes a spatial-frame Jacobian J_s . Therefore, we can interpret that GIC is formulated on the body frame coordinate attached to the end-effector, while CIC is formulated relative to the spatial frame.

IV. PROBLEM DEFINITION AND SOLUTION APPROACH

A. Problem Setup

1) *Overview*: Our ultimate goal is to provide a learning-based solution to solve a peg-in-hole task, a classic representative of contact-rich force-based robotic manipulation tasks. We will address this problem in the framework of learning variable impedance control, where the gain scheduling policy of the impedance control laws is trained using learning algorithms. In particular, behavior cloning (BC) from expert demonstrations is utilized in a supervised learning fashion to obtain the gain scheduling policy.

To achieve this, we introduce a gain scheduling policy parameterized by a simple neural network. This neural network takes as input the positional signals representing the current end-effector pose and the desired goal pose, such as e_G and e_C . We consider the gain-scheduling neural network policy output as an action (a_t) and input positional signals as the states (s_t). Formally,

$$a_t = \mu_\theta(s_t), \quad (K_p, K_R)_t = h(a_t) = \pi_\theta(s_t), \quad (12)$$

where h denotes a mapping from the action signal to the gains, K_p and K_R are impedance gains defined in (6) or (11), and θ denotes parameters of the neural network μ_θ . We employ a standard multi-layer perceptron (MLP) as a neural network. For the rest of the paper, we will call π_θ as a gain-scheduling policy and drop the subscript t for the compactness of notation.

To show the effectiveness of the geometric formulation, we will propose and compare two different approaches to learning variable impedance control: 1. Selection of the control rule (GIC vs CIC), 2. Selection of the states s_t (Geometric error vs Cartesian error).

The performances are evaluated on four main scenarios in Fig. 1. The gain scheduling policy μ_θ is trained only in the default scenario (Fig. 1(a)). The trained policy is then tested in the other scenarios (Fig. 1s(b)-(d)) to evaluate its zero-shot transferability and robustness to OOD data. All simulations are conducted in the Mujoco simulation environment [29] for environment setup and the Berkeley RL kit [30] for the RL training. The GitHub repository of this project is published in https://github.com/Joohwan-Seo/GIC_Learning_public.

2) *Action Mapping to Gains*: The mapping from the actions to the impedance gains $h(a)$ is defined here. To map the actions from policy to impedance gains, we first consider diagonal components of the matrix gains K_p and K_R as follows:

$$K_p = \text{diag}([k_{p_1}, k_{p_2}, k_{p_3}]), \quad K_R = \text{diag}([k_{r_1}, k_{r_2}, k_{r_3}]).$$

The damping matrix K_d is fixed as

$$K_d = 8 \cdot \text{diag}([k_{p_1}, k_{p_2}, k_{p_3}, k_{r_1}, k_{r_2}, k_{r_3}])^{0.5} \quad (13)$$

For the task shown in Fig. 1(a), a symmetric structure is considered in the action mapping. Specifically, we have the same action mapping in the x and y directions (k_{p_1} and k_{p_2}) and a different mapping in the z direction. In the rotational part, the action mapping for k_{r_1} , k_{r_2} , and k_{r_3} is the same. The selected action mapping is as follows:

$$\begin{aligned} k_{p_i} &= 10^{a_i+2.5}, \quad \text{for } i = 1, 2, & k_{p_3} &= 10^{1.5 \cdot a_3+2.0}, \\ k_{r_j} &= 10^{0.6 \cdot a_j+2.0}, \quad \text{for } j = 1, 2, 3, \end{aligned}$$

where a denotes an action, $a = [a_1, a_2, \dots, a_6]^T \in [-1, 1] \times \dots \times [-1, 1] \subset \mathbb{R}^6$.

B. Solution Approach

In this subsection, we will first introduce our proposed approach. In what follows, the behavior cloning (BC) to obtain the gain-scheduling policy is introduced.

1) *Proposed Approach*: Our proposed approach utilizes GIC with a learning impedance gain scheduling policy, where the input to the neural network s is the GCEV $e_g(g, g_d)$ defined in (9). The action from the gain scheduling policy then becomes $(K_p, K_R) = \pi_\theta(e_g)$. The GIC control law (6) equipped with gain scheduling policy $\pi_\theta(e_g)$ has a crucial property for learning transferability as shown in the following lemma.

Lemma 1. Left-invariance of the GCEV (9) and the elastic wrench (6)

$e_g(g, g_d)$ and $f_g(g, g_d)$ are left-invariant to the arbitrary left-transformation g_l in $SE(3)$.

Proof. Let $g_l = (R_l, p_l)$. Then, the left-transformed homogeneous matrix $g_l g$ is calculated in the following way:

$$g_l g = \begin{bmatrix} R_l & p_l \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} R_l R & R_l p + p_l \\ 0 & 1 \end{bmatrix}$$

Similarly, $g_l g_d = (R_l R_d, R_l p_d + p_l)$. The left-transformed GCEV is then

$$\begin{aligned} e_g(g_l g, g_l g_d) &= \begin{bmatrix} R^T R_l^T (R_l p + p_l - R_l p_d - p_l) \\ ((R_l R_d)^T R_l R - (R_l R)^T R_l R_d)^\vee \end{bmatrix} \\ &= \begin{bmatrix} R^T (p - p_d) \\ (R_d^T R - R^T R_d)^\vee \end{bmatrix} = e_g(g, g_d) \end{aligned} \quad (14)$$

As a result, the gain scheduling policy $\pi_\theta(e_g)$ is also left-invariant, i.e., let $(K_p, K_R)(g, g_d) = (\pi_\theta \circ e_g)(g, g_d)$, then $(K_p, K_R)(g_l g, g_l g_d) = (K_p, K_R)(g, g_d)$. Similarly, the left-transformed elastic wrench is

$$\begin{aligned} f_g(g_l g, g_l g_d) &= \begin{bmatrix} (R_l R)^T R_l R_d K_p^l (R_l R_d)^T (R_l p + p_l - R_l p_d - p_l) \\ (K_R^l (R_l R_d)^T R_l R - (R_l R)^T R_l R_d K_R^l)^\vee \end{bmatrix} \\ &= \begin{bmatrix} R^T R_d K_p R_d (p - p_d) \\ (K_R R_d R - R^T R_d K_R)^\vee \end{bmatrix} = f_g(g, g_d), \end{aligned} \quad (15)$$

where we used notations $(K_p^l, K_R^l) = (K_p, K_R)(g_l g, g_l g_d)$ and $(K_p, K_R) = (K_p, K_R)(g, g_d)$ to avoid clutter. This shows the left-invariance of $e_g(g, g_d)$ and $f_g(g, g_d)$. \square

Domain Randomization: Domain randomization is a crucial technique in both BC and RL to enhance robustness by allowing the neural network to explore a broader range of state space [31]. In conventional impedance gains learning problem [3], domain randomization is typically applied to both the initial and goal end-effector poses. However, under the proposed GIC framework, domain randomization is only necessary for the initial pose of the end-effector, as demonstrated in the following proposition.

Proposition 1. *For the learning variable impedance control problem based on GIC law (5), the following equation holds true:*

$$f_g(g, g_l g_d) = f_g(g_l^{-1} g, g_d) \quad (16)$$

Proof. By lemma 1, $f_g(g, g_d)$ is left-invariant to arbitrary $SE(3)$ transformation g_l acting on the left, i.e., $f_g(g_l g, g_l g_d) = f_g(g, g_d)$. Then, (16) becomes

$$f_g(g, g_l g_d) = f_g(g_l g_l^{-1} g, g_l g_d) = f_g(g_l^{-1} g, g_d). \quad (17)$$

\square

The effects of Proposition 1 on learning strategies are as follows. First, since the main driving force of (8) with gain scheduling policy is $f_g(g, g_d)$ in (6), we focus on the properties of $f_g(g, g_d)$. The domain randomization on the target pose can be represented by $f_g(g, g_l g_d)$ where $g_l \in SE(3)$ is arbitrary. Then, the result of Proposition 1 reads that

$$f_g(g, g_l g_d) = f_g(g_l^{-1} g, g_d)$$

Note that following the axioms of groups (Chap 2.1, [6]), g_l^{-1} can be denoted by another group element g'_l since g_l is arbitrary. Finally, $f_g(g'_l g, g_d)$ means that the domain ran-

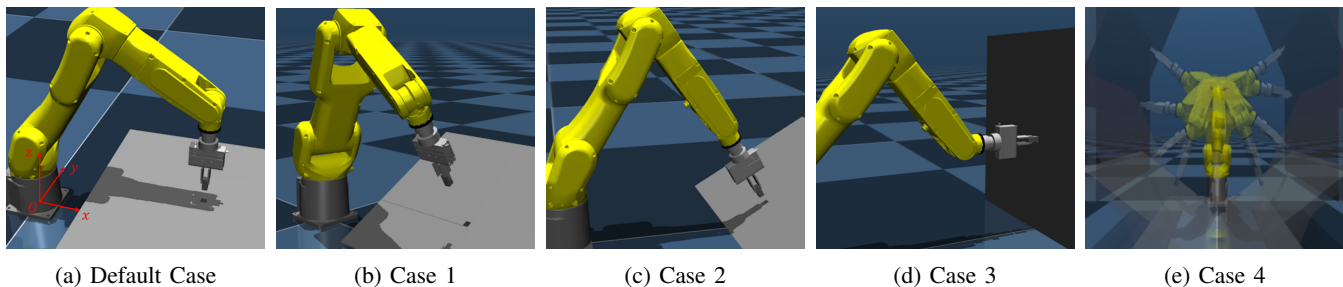


Fig. 1: Robot performing a peg into a hole insertion task in different scenarios for testing learning transferability. The data is collected, and the policy is trained only in performing the task shown in (a). The trained policy is then tested on the tasks shown in (b)-(e), where the insertion hole is translated in different orientations. (b): tilted in $+x$ direction for 30° (c): tilted in $-y$ direction for 30° (d): tilted in $-y$ direction for 90° (e): tilted in $+x$ direction in arbitrary angle $\phi \in [-135^\circ, 135^\circ]$. The coordinate frame is attached to the first figure.

domization on the target pose of the end-effector is identical to randomization on the initial pose. Therefore, independent domain randomizations on both the target pose and the initial pose are not necessary during the training process. Only the initial pose relative to the target pose needs to be randomized, or vice versa, but not both. This result is crucial in robotics applications where the cost of collecting data in the real world is substantial.

2) *Behavior Cloning (BC)*: We employed naive behavior cloning to directly estimate the impedance gain for the given input states in a supervised learning fashion from the expert’s demonstration. The learning problem for behavior cloning can be defined as $\theta^* = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^N \|a_i - \mu_{\theta}(s_i)\|_2^2$, where N is the length of the dataset. We collect 300 expert demonstration trajectories with $N \sim 450k$. The BC policy is trained following the standard deep learning fashion - stochastic gradient descent on a sampled batch dataset with an Adams optimizer, learning rate schedule, and early stopping.

To collect the required dataset in the simulation environment, we develop a heuristic rule-based scripted expert policy. The main intuition is to use small z gains and high gains in the x and y directions when the robot approaches the hole while using high z gains during the insertion to overcome the friction between the peg and the hole. As a result, the robot first aligns the peg with the hole’s axis and then gradually pushes the peg into the hole. The expert’s gains are selected by trial and error, depending on the error signal e_G . During data collection for BC, small amounts of noise are added to the scripted expert policy to enhance robustness.

V. EXPERIMENTS AND DISCUSSIONS

A. Behavior Cloning (BC)

The results of the BC experiments are presented in Table. I. Each task was tested 100 times, and the success cases were counted. The BC policy trained with the GCEV and executed with the GIC (GIC+GCEV) successfully transferred the trained policy to the other tasks, without significant drop in the success rate. However, the BC policy trained with a CEV and executed with CIC (CIC+CEV) failed to transfer the trained policy, resulting in a dramatic decrease in the success rate. The reason for this difference in transferability can be attributed to the error vector representation. The relationship between left invariance and transferability is further explained in Remark 2.

TABLE I: Success rates of the BC policies for the proposed and the benchmark approaches (Tested 100 times each, Values in Percentage %)

Method	Default	Case 1	Case 2	Case 3
BC Proposed (GIC+GCEV)	100	99	95	100
BC Benchmark (CIC+CEV)	100	0	0	1
BC Mixed 1 (GIC+CEV)	99	54	49	27
BC Mixed 2 (CIC+GCEV)	100	0	0	0

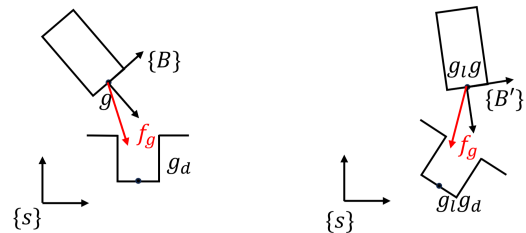


Fig. 2: (Left) Peg and hole configurations are represented by g and g_d , respectively. (Right) Peg and hole undergo left transformation via an action $g_l \in SE(3)$. The GCEV e_G (9) and elastic force f_G (6) are invariant to the left transformation of $SE(3)$. $\{s\}$ represents spatial coordinate frame, while $\{B\}$ and $\{B'\}$ represent body coordinate frames.

Remark 2. Why does left-invariance matter?

The left-invariance of the error vector implies that the chosen error vector is invariant to the selection of the coordinate system. In this paper, we interpret left-invariance in a slightly different manner. Consider the situation where the desired and current configurations are transformed through a left action of the $SE(3)$ group, which corresponds to a change in the spatial coordinate frame – See Fig. 2. Due to the left-invariance of the GCEV (9), the error vector remains unchanged in cases (a) and (b) in Fig. 2. Therefore, from the perspective of the proposed approach, the task remains invariant to translational/rotational perturbations. In this perspective, the use of the left-invariant error vector e_G can help address distributional shift or out-of-distribution issues, as the trained policy will consistently encounter the same input e_G .

B. Left-Invariance is not enough

The question arises whether training a policy based on GCEV or equivalent left-invariant features enables trans-

ferability to translational/rotational perturbations. To answer this question, we test different combinations of trained gain scheduling policies and control methods: BC policy with Cartesian error vector executed with GIC (GIC+CEV) and BC policy with a GCEV executed with CIC (CIC+GCEV). If left-invariance of the feature were the only factor for transferability, we would expect the BC policy with the Cartesian error vector executed with GIC (GIC+CEV) to not be transferable due to the lack of left-invariance in the Cartesian error vector. Conversely, the BC policy using a GCEV executed with CIC would be transferable.

However, the experimental results presented in 3rd and 4th rows of Table I contradict the hypothesis. The GIC with a gain scheduling policy trained with the Cartesian error vector (GIC + CEV) showed some transferability in Case 1 and Case 2, with success rates near 50%. However, in Case 3, with a tilt of 90 degrees, the success rate drops to 27% due to encountering an unexperienced state distribution. This suggests that a left-invariant gain scheduling policy alone is not enough for transferability.

On the other hand, the CIC using a GCEV produced results consistent with our hypothesis. Even though the gain scheduling policy for CIC was trained with a GCEV, the direction of the gains and resulting force direction are still represented in the Cartesian frame. Thus, the policy outputs the same gains it was trained on, but the resulting force direction is not suitable for tilted cases – see in Fig. 3.

Consequently, it is concluded that the left-invariant gain scheduling policy alone is not enough, and a more fundamental factor is needed to address transferability - the direction of forces. Unlike CIC, the forces in GIC are defined in the body frame, resulting in the automatic change in the direction of forces (See Fig. 2, f_G on each case) – which implies an *equivariance* property. We state that the key to transferability lies in a control law represented in the body-frame coordinate and the left-invariant gain scheduling policy. The invariant gain scheduling policy is obtained from the neural network with a GCEV as input, while the left-invariant feedback control law is inherited from the structure of GIC. To establish a connection between our statement and the equivariance property, we first present the definition of equivariance.

Definition 1. Consider a function $f : \mathcal{X} \rightarrow \mathcal{Y}$, i.e., $y = f(x)$ with $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. The function f is equivariant to the group g if the following condition is satisfied [4]:

$$f(\rho^{\mathcal{X}}x) = \rho^{\mathcal{Y}}f(x) \quad (18)$$

where $\rho^{\mathcal{X}}$ represents the action of group g in the domain \mathcal{X} and $\rho^{\mathcal{Y}}$ represents the action of group g in the codomain \mathcal{Y} .

Based on Definition 1, we propose the following proposition.

Proposition 2. The feedback terms in GIC law (5) described in the body frame are equivariant if it is described in the spatial frame.

Proof. Consider the feedback terms $f_G(g, g_d)$ (6) and $e_V(g, g_d)$ (7), which are denoted on the body frame. See Fig. 2 for the coordinate systems. To show the equivariance property,

we first show that the invariance on the body frame implies equivariance on the spatial frame.

Let $f_G^s(g, g_d)$ be $f_G(g, g_d)$ denoted in the spatial frame $\{s\}$. We note that g and g_d are described on the spatial frame $\{s\}$. Then, the left-transformed elastic force can be denoted by $f_G(g_lg, g_lg_d)$ on the transformed body frame g_lg and is left-invariant by lemma 1, i.e., $f_G(g_lg, g_lg_d) = f_G(g, g_d)$. We now consider the left-transformed elastic force $f_G^s(g_lg, g_lg_d)$ with respect to the spatial frame $\{s\}$. The coordinates of the wrenches between the body and the spatial frame can be transformed by the following equations (see Ch. 2.5 of [6]):

$$f_G^s(g, g_d) = \text{Ad}_{g^{-1}}^T f_G(g, g_d) \quad (19)$$

$$f_G^s(g_lg, g_lg_d) = \text{Ad}_{(g_lg)^{-1}}^T f_G(g_lg, g_lg_d) = \text{Ad}_{g^{-1}g_l^{-1}}^T f_G(g, g_d),$$

where the Adjoint map is defined in (7). Therefore, the following equations hold:

$$\begin{aligned} f_G^s(g_lg, g_lg_d) &= \text{Ad}_{g^{-1}g_l^{-1}}^T (\text{Ad}_{g_l^{-1}}^T)^{-1} f_G^s(g, g_d) \\ &= (\text{Ad}_g \text{Ad}_{g_l^{-1}})^T f_G^s(g, g_d) = \text{Ad}_{g_l^{-1}}^T f_G^s(g, g_d), \end{aligned} \quad (20)$$

where we use a composition rule for the Adjoint map, $\text{Ad}_{g_1} \text{Ad}_{g_2} = \text{Ad}_{g_1 g_2}$, and an inverse property of the Adjoint map, $(\text{Ad}_g^T)^{-1} = \text{Ad}_{g^{-1}}^T$. Here, the domain \mathcal{X} in Def. 1 is $SE(3)$ (g or g_d), and the representation $\rho^{\mathcal{X}}$ is g_l , while the codomain \mathcal{Y} is $se^*(3)$ ($f_G^s(g, g_d)$ or $f_G^s(g_lg, g_lg_d)$) and the representation $\rho^{\mathcal{Y}}$ is $\text{Ad}_{g_l^{-1}}^T$. Therefore, $f_G^s(g, g_d)$ is equivariant in $SE(3)$.

Similarly, to show the equivariance of $e_V(g, g_d)$ (7) on the spatial frame, the invariance on the body frame is only needed. The body frame velocities V^b and V_d^b are invariant to the left transformation since it is defined on the body frame and the Adjoint map $\text{Ad}_{g_{ed}}$ is invariant as the relative transformation matrix $g_{ed} = g^{-1}g_d$ is invariant to the left transformation. As a result, the feedback terms in GIC law (5) are equivariant in $SE(3)$. \square

Note that we only consider feedback terms in (5) since the feedforward terms are just employed to cancel the manipulator dynamics, not affecting the closed-loop dynamics. A remark on the Proposition 2 is provided.

Remark 3. Extensions to general force-based policy

It is worth mentioning that our concept proposed in Proposition 2 can also be extended to general force-based policies. If a force-based policy is left-invariant in the body-frame, e.g., implemented using a neural network with left-invariant features as input, and described in the end-effector body frame, it will be guaranteed to be left-equivariant in the spatial frame. The GCEV (9) introduced in this paper is an example of such a left-invariant feature.

C. Demonstration of Sample Efficiency

In addition to its enhanced learning transferability, equivariant learning approaches offer a significant advantage in terms of increased sample efficiency. To evaluate whether our proposed method demonstrates improved sample efficiency, we trained a gain scheduling policy with CEV, making the GIC+CEV non-equivariant, while employing data augmentation, as illustrated in Case 4 of Figure 1(d). To simplify the

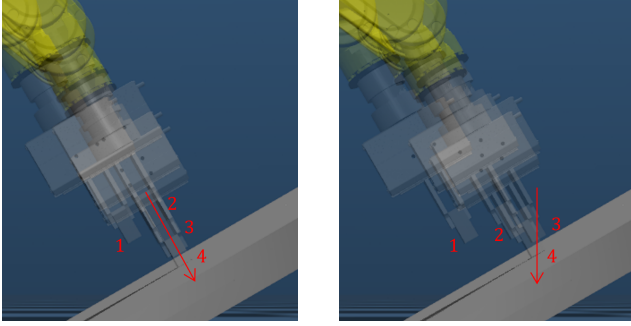


Fig. 3: Time-snap plots of for PiH task executed by (Left) GIC+GCEV, and (Right) CIC+GCEV. The numbers in the figure denote the time snap index, and the red arrow denotes the insertion direction. The desired point is located at the tip of the hole as expert policy, CIC first aligns with the Cartesian z axis as it is trained in the default case (Fig. 1(a)).

TABLE II: Comparison between GIC+GCEV and GIC+CEV with data augmentation. Tested 100 times for uniform randomly sampled tilting angles within range.

Default	GIC+GCEV	GIC+CEV w/ aug.
Success Rate*	99 %	93 %
Success Rate to OOD**	96 %	43 %
Size of Dataset	488,854	4,194,514
Size of NN	$[128] \times 3$	$[128] \times 7$

* $\phi \in [-90^\circ, 90^\circ]$,

**OOD: Out-of-distribution data, $\phi \in [-135^\circ, -90^\circ) \cup (90^\circ, 135^\circ]$

testing, we assumed that the goal pose is only rotated in the $+x$ axis. The dataset was collected across tilting angles ranging from -90° to 90° in the $+x$ axis. As the state space region that the neural network needs to memorize expands, it becomes necessary to increase its size.

The comparison results are summarized in Table II. When tested within the training region, both GIC+GCEV and GIC+CEV with data augmentation showed near-perfect performance, with a slight advantage for GIC+GCEV. However, in OOD scenarios, the success rate of GIC+CEV with data augmentation dropped as it encountered OOD data inputs. Even with the goal pose tilting in only one direction, it required 8.6 times more data points and neural networks that were 2.3 times larger. It is important to note that if tilting occurs in different dimensions, the required samples and neural network size will increase in power scale. These results demonstrate the superior sample efficiency of the equivariant policy compared to the non-equivariant one.

VI. HARDWARE EXPERIMENT

To validate the proposed concept, we implemented our methods on the hardware robot; Fanuc 200iD/7L. The proposed approach is tested in a tight PiH task, where the clearance between the peg and hole is 0.05mm. For ease of hardware implementation, we implement the GIC (geometric impedance control) as a Geometric Admittance Control (GAC) law. To define the admittance control version of the GIC (GAC), the desired closed-loop control system $M\dot{e}_V + K_d e_V + f_G(g, g_d) = \tilde{T}_e$ is first considered, where M is a fixed desired inertia matrix, K_d is a fixed damping matrix, \tilde{T}_e is an external

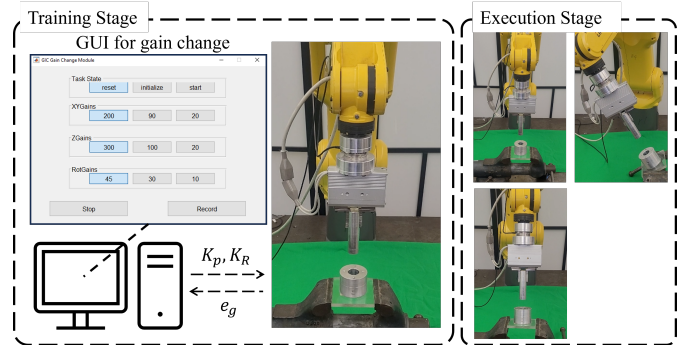


Fig. 4: Real robot experiment description. **Training Stage:** Schematic for the data collection method is shown. The data is collected on the default case. **Execution Stage:** The trained policy is then directly tested on the holes with different positions/orientations. Upper-left: Default Case. Upper-right: the hole is tilted 30° in $+x$ axis (Case 1), Lower-left: -22.5° in $+y$ axis. (Case 2)

force acting on the end-effector, and f_G is our beloved elastic wrench (6) in $SE(3)$. Since our task is a PiH, we again fixed the desired configuration g_d as a constant, which leads to the following desired closed-loop system:

$$M\dot{V}^b + K_d V^b + f_G(g, g_d) = \tilde{T}_e. \quad (21)$$

In what follows, we formulate (21) into a discrete-time setting for the derivation of the admittance control law.

$$V^b(k+1) = V^b(k) + \Delta t \cdot M^{-1} \left(\tilde{T}_e(k) - f_G(k) - K_d V^b(k) \right),$$

where we use $f_G(k)$ to denote $f_G(g, g_d)$ at k instance for simplicity, and Δt is a sampling time. The $V^b(k+1)$ term can then be considered to be the desired velocity at the $k+1$ instance. Thus, the geometric admittance control law now boils down to a joint-space velocity PD control, with desired joint-space velocity $\dot{q}_d(k)$ given as $\dot{q}_d(k) = (J_b(k))^{-1} V^b(k)$. For the impedance learning problem, the dataset $\{(e_G, (K_p, K_R))\}_{i=1}^N$ is required. The data collection process is summarized in the training stage of Fig. 4. The robot is provided with $g_d = (R_d, p_d)$ and is controlled by the GAC to execute a PiH task. The human expert supervising this task process changes the gains (K_p, K_R) of GAC in real-time using the GUI, and the gain signals are sent to the manipulator with an ethernet UDP communication protocol. The output signal e_G is received from the communication module and is recorded alongside the gain command signal as the dataset. We collected 75 trajectories, which sums up to $\sim 250k$ of dataset size. Similar to the simulation experiment, we also collected data and trained policy only on the default case, and tested on the tilted cases, as summarized at the execution stage in Fig. 4. We set the failure case when the peg is stuck to the whole, e.g., the task cannot be completed within 60s.

The result of the hardware validation is presented in Table III. As can be seen in the result, the proposed approach of utilizing GIC (or GAC) together with the gain scheduling policy implemented with GCEV showed perfect success rates, and the trained policy was transferrable to previously unseen cases.

TABLE III: Success rates of the BC policy for the proposed approach in the hardware implementation (Tested 10 times each, Values in Percentage %)

Method	Default	Case 1	Case 2
Proposed (GAC+GCEV)	100	100	100

VII. CONCLUSIONS AND FUTURE WORKS

In this paper, a geometric approach leveraging $SE(3)$ group invariance and equivariance for contact-rich robotic manipulation task learning is presented. To solve the Peg-in-Hole (PiH) task, the proposed approach builds on top of the geometric impedance control (GIC), where its impedance gains are changed via a left-invariant gain scheduling policy. Expert behavior cloning is chosen for training the gain scheduling policy. Through theoretical analysis, we prove that the proposed GIC and the geometrically consistent error vector (GCEV) used for learning are left-invariant relative to $SE(3)$ group transformations when represented in the end-effector's body frame system, enabling learning transferability. Furthermore, we show that left invariance in the body-frame representation leads to $SE(3)$ equivariance of the proposed approach when described in a spatial frame. A PiH simulation experiment confirms the learning transferability of our proposed method, which is not exhibited by the well-known Cartesian space-based benchmark approach. These results are further validated on an actual PiH robotic hardware implementation, and the pipeline for the hardware implementation is also presented.

For future work, we will address more realistic scenarios where the exact goal poses are unknown and need to be estimated via sensors, e.g., images or point cloud-based inputs. Moreover, the proposed approach will also be demonstrated in the other types of contact-rich and force-related tasks, such as surface wiping, cable assembly, and pivoting [32].

REFERENCES

- [1] H. Ravichandar *et al.*, "Recent advances in robot learning from demonstration," *Annual review of control, robotics, and autonomous systems*, vol. 3, pp. 297–330, 2020.
- [2] X. Zhang *et al.*, "Learning variable impedance control via inverse reinforcement learning for force-related tasks," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2225–2232, 2021.
- [3] C. C. Beltran-Hernandez *et al.*, "Variable compliance control for robotic peg-in-hole assembly: A deep-reinforcement-learning approach," *Applied Sciences*, vol. 10, no. 19, p. 6923, 2020.
- [4] T. Cohen and M. Welling, "Group equivariant convolutional networks," in *International conference on machine learning*. PMLR, 2016, pp. 2990–2999.
- [5] E. J. Bekkers *et al.*, "Roto-translation covariant convolutional networks for medical image analysis," in *Medical Image Computing and Computer Assisted Intervention—MICCAI 2018: 21st International Conference, Granada, Spain, September 16–20, 2018, Proceedings, Part I*. Springer, 2018, pp. 440–448.
- [6] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 1994.
- [7] J. Seo *et al.*, "Geometric impedance control on $SE(3)$ for robotic manipulators," *IFAC World Congress 2023, Yokohama, Japan*, 2023.
- [8] A. Zeng *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.
- [9] A. Simeonov *et al.*, "Neural descriptor fields: $SE(3)$ -equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6394–6400.

- [10] H. Ryu *et al.*, "Equivariant descriptor fields: $SE(3)$ -equivariant energy-based models for end-to-end visual robotic manipulation learning," in *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.
- [11] —, "Diffusion-edfs: Bi-equivariant denoising generative modeling on $se(3)$ for visual robotic manipulation," *arXiv preprint arXiv:2309.02685*, 2023.
- [12] J. Kim *et al.*, "Robotic manipulation learning with equivariant descriptor fields: Generative modeling, bi-equivariance, steerability, and locality," in *RSS 2023 Workshop on Symmetries in Robot Learning*, 2023.
- [13] C. Pan *et al.*, "Tax-pose: Task-specific cross-pose estimation for robot manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 1783–1792.
- [14] H. Ha and S. Song, "Flingbot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding," in *Conference on Robot Learning*. PMLR, 2022, pp. 24–33.
- [15] S. Kim *et al.*, " $SE(2)$ -equivariant pushing dynamics models for tabletop object manipulations," in *Conference on Robot Learning*. PMLR, 2023, pp. 427–436.
- [16] V. der Pol *et al.*, "MDP homomorphic networks: Group symmetries in reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4199–4210, 2020.
- [17] D. Wang *et al.*, "Equivariant q learning in spatial action spaces," in *Conference on Robot Learning*. PMLR, 2022, pp. 1713–1723.
- [18] D. Wang, R. Walters, and R. Platt, " $SO(2)$ -equivariant reinforcement learning," *arXiv preprint arXiv:2203.04439*, 2022.
- [19] T. Inoue *et al.*, "Deep reinforcement learning for high precision assembly tasks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 819–825.
- [20] S. Kozlovsky, E. Newman, and M. Zacksenhouse, "Reinforcement learning of impedance policies for peg-in-hole tasks: Role of asymmetric matrices," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 898–10 905, 2022.
- [21] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.
- [22] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal on Robotics and Automation*, vol. 3, no. 1, pp. 43–53, 1987.
- [23] F. Bullo and R. M. Murray, "Tracking for fully actuated mechanical systems: a geometric framework," *Automatica*, vol. 35, no. 1, pp. 17–34, 1999.
- [24] T. Lee *et al.*, "Geometric tracking control of a quadrotor uav on $SE(3)$," in *49th IEEE conference on decision and control (CDC)*. IEEE, 2010, pp. 5420–5425.
- [25] F. Caccavale *et al.*, "Six-dof impedance control based on angle/axis representations," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 289–300, 1999.
- [26] Y. Zhu *et al.*, "robosuite: A modular simulation framework and benchmark for robot learning," in *arXiv preprint arXiv:2009.12293*, 2020.
- [27] H. Ochoa and R. Cortesão, "Impedance control architecture for robotic-assisted mold polishing based on human demonstration," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 4, pp. 3822–3830, 2021.
- [28] S. Shaw, B. Abbatematteo, and G. Konidaris, "RMPs for safe impedance control in contact-rich manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2707–2713.
- [29] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.
- [30] "Berkeley RL Kit," <https://github.com/rail-berkeley/rlkit>, accessed: 2023-07-01.
- [31] J. Tobin *et al.*, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.
- [32] X. Zhang *et al.*, "Efficient sim-to-real transfer of contact-rich manipulation skills with online admittance residual learning," *arXiv preprint arXiv:2310.10509*, 2023.