IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Lie Group-Based User Motion Refinement Control for Teleoperation of a Constrained Robot Arm

Jonghyeok Kim, Donghyeon Lee, Youngjin Choi, *Senior Member, IEEE*, and Wan Kyun Chung, *Fellow, IEEE*

***Abstract*—In unilateral teleoperation systems, robots often face challenges when performing tasks with specific geometric constraints. These constraints restrict the robot's movements to certain directions, requiring accurate control of its position and orientation. If the operator's commands do not consider these constraints, excessive contact force may occur, potentially damaging the robot and its environment. Such scenarios can also trigger frequent emergency stops, even with conventional admittance control. To mitigate these issues, we propose a new teleoperation framework tailored for handling geometric constraints. This framework comprises two main components. 1) Geometric Constraint Identification: We use a straightforward line regression method based on Lie group theory to identify geometric constraints. 2) Motion Command Reshaping: The operator's motion commands are safely recalculated using a projection filter coupled with a Lie group setpoint controller. This approach ensures that the robot's movements strictly conform to the identified geometric constraints. As a result, this approach significantly reduces the interaction forces and prevents the risk of severe failures or accidents.**

***Index Terms*—Telerobotics and teleoperation, recognition, compliance and impedance control.**

I. Introduction

ROBOTIC teleoperation, which is traditionally used for remote manipulation in hazardous environments and to teach robots through physical demonstrations [1], [2], has evolved to perform various complex tasks [3], [4]. Despite these advances, these tasks remain challenging due to geometric constraints restricting the degree of freedom of motion. For example, a robot is restricted to linear, circular, or helical motion when opening a drawer, opening a door, or screwing, respectively. In particular, it becomes more complex when position and orientation constraints are coupled [5], [6].

In teleoperation, operators face significant challenges due to constraints in motion generation, particularly with positional and orientational constraints. Typically, robots are controlled remotely using task space commands. However, it is difficult to perceive these constraints accurately. This lack of clarity can lead to the unintentional generation of dangerous motion commands. Motion commands that violate the constraint cause excessive contact force between the robot and the environment, leading to task failures and hardware damage, e.g., robotic tool breakage, environmental damage, and emergency stops.

As a remedy, an admittance control-based teleoperation system can be used since it allows physical interaction by modulating the robot's reactive motion with a measured external wrench. It can prevent the generation of an excessive interaction force while simultaneously controlling the position of the robot [7], [8], [9]. However, its main objective remains to follow the operator's motion commands, which may not always be consistent with the geometric constraint and can increase the interaction force if the commands severely violate the constraint.

To prevent the abovementioned failures, motion commands that consistently satisfy the constraint are needed. In other words, the robot system should limit or ignore dangerous motion commands to remain within the allowable range. Identifying and utilizing the constraint information can be a promising solution since this knowledge provides a way to inform the robot to restrict its movements to the permissible path. There is a constraint inference method using a force-torque (F/T) sensor based on the principle of virtual work [10]. However, they focused on classifying the constraints rather than accurately estimating the constraint parameter. On the other hand, naïve fitting methods to fit geometry to data points are also available [11], [12], but they only cover simple primitives such as lines, planes, and circle arcs. The combined position/orientation cases, such as the helical constraint, cannot be identified in this case.

When geometric constraints are known, virtual fixtures (VFs) apply force feedback to restrict robot movement within the allowable region [13], using simple functions such as P control or PD control between the constraint and the closest point of the robot [14], [15]. However, this method cannot completely confine the robot's movement to the constraint under arbitrary motion commands.

Manuscript received 17 December 2023; accepted 23 April 2024. Date of publication 15 May 2024; date of current version 22 May 2024. This letter was recommended for publication by Associate Editor M. Frego and Editor A. Peer upon evaluation of the reviewers' comments. This work was supported in part by Industrial Strategic Technology Development Program (General purpose multi mode robot teaching device for difficult task where 0.1 mm resolution of position and vel acc force teachings are essential) funded by the Ministry of Trade, Industry and Energy of Korea (MOTIE) under Grant 20009396, and in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2020R1I1A2074953. *(Corresponding author: Wan Kyun Chung.)*

Jonghyeok Kim and Wan Kyun Chung are with the Department of Mechanical Engineering, Pohang University of Science and Technology, Gyeongbuk 37673, South Korea (e-mail: sentojh@postech.ac.kr; wkchung@postech.ac.kr).

Donghyeon Lee is with the Celloid Company Ltd., Pohang, Gyeongbuk 37673, South Korea (e-mail: sansoveria@gmail.com).

Youngjin Choi is with the Department of Robotics, Hanyang University, Ansan 15588, South Korea (e-mail: cyj@hanyang.ac.kr).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3401135, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3401135

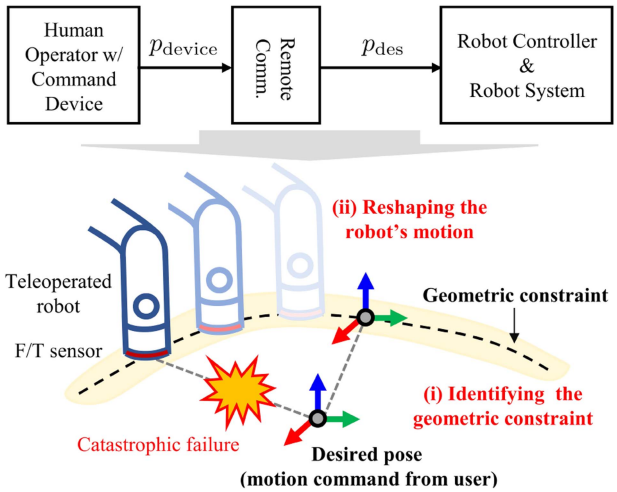


Fig. 1. Conceptual illustration for the situation of the target problem. Under geometric constraints, the robot should follow (black dotted line), but dangerous motion commands will cause catastrophic failures. Therefore, this work proposes methods to avoid such failures: 1) identifying the geometric constraints and 2) reshaping the robot's motion onto the constraint.

This letter introduces a teleoperation framework to perform tasks with geometric constraints. The framework comprises geometric constraint identification and motion reshaping strategy based on the identified constraint. Our key idea is that the representation of screw motion simplifies nonlinear rigid body motion using the Lie group theory [6], [16]. This approach simplifies the identification of complex 3D motions under coupled positional and orientational constraints, giving linearity. Moreover, the proposed motion reshaping allows the robot to move in a permissible path that adheres to the identified constraint; thus, excessive interaction force can be prevented during teleoperation.

The remainder of this letter is organized as follows. In Section II, the problem statement with the preliminary is introduced. In Section III, the details of the proposed method are presented. In Section IV, the experimental results are reported. In Section V, the discussion is presented.

II. Problem Statement and Preliminary

A. Problem Statement

Recent robot manipulation research has remarkably increased the necessity of satisfying constraints. The Riemannian Motion Policy (RMP) successfully superposes multiple reactive motion policies, which account for collision avoidance, joint limits, and long-range arm navigation [17]. Safe exploration in robotic Reinforcement Learning (RL) often uses physical and safety constraint knowledge to resolve the gap between simulation and the real world [18]. Furthermore, the Control Barrier Function (CBF) has been used for safety-critical applications through forward invariance. It is often combined with optimization-based control algorithm synthesis [19], [20]. These approaches require predefined constraint knowledge and usually focus on collision avoidance and joint position and velocity limits.

This letter is slightly different from the ones mentioned. We aim to control the teleoperated robot, where the human operator remotely performs tasks[1] with geometric constraints. The situation of the target problem is shown in Fig. 1. The human operator gives motion commands[2] using a command device. In free space, the robot has no constraints in terms of interacting with an environment, so it is out of our scope. However, during task execution, the problem arises that the robot cannot freely move to follow the desired pose due to geometric constraints. The robot's motion should be satisfied if a specific path is strictly followed. Otherwise, as mentioned earlier, it increases the contact force and eventually causes failures. This frequently occurs because dangerous motion commands are unavoidable. Therefore, this letter focuses on reshaping (or refining) the robot's motion to satisfy geometric constraints rather than directly tracking the motion command. Consequently, *geometric constraints* are defined as the path the robot should follow during the task. The explanation of the geometric constraints handled in this letter is suggested in the following subsection.

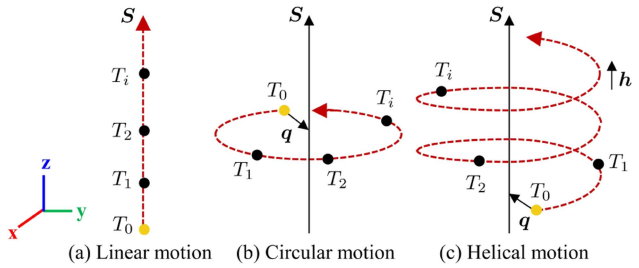


Fig. 2. Types of geometric constraints. The red dotted line shows the constrained path. Well-known prismatic, revolute, and helical constraints impose linear, circular, and helical motion, characterized as a linear relation in Lie algebra. $\{q, S, h\}$ is a screw notation represented by Lynch and Park [5] to explain the constraint extraction to be considered as line regression. In Section III-B, the screw parameters are computed by our geometric constraint representation.

B. Preliminary

The geometric constraints handled in this letter are verified in Fig. 2. We define the (constrained) motion as a set of transformation matrices that comprise the constrained path: $\{T_0, T_1, \ldots, T_i\}$ for each $T_i \in SE(3)$. $SE(3)$ is called *Special Euclidean group*, which is a group of homogeneous transformation matrices.[3] Since $SE(3)$ is a matrix Lie group, the tangent space at the identity forms a special structure called the Lie algebra. In particular, the Lie algebra of $SE(3)$ is a set of 4×4 matrix representations of twist, denoted by $\mathfrak{se}(3)$, is isomorphic to $\mathbb{R}^6$. In Lie group theory, the maps between the Lie group and the corresponding Lie algebra are always defined ($\exp$ and $\log$). Moreover, we explore the differential of the exponential map, which links the body twist and the time derivative of the exponential coordinate. Due to a space limit, please refer to [6] and [16] for a more detailed explanation, especially the

[1]These tasks directly involve interacting with an environment, so we basically consider the admittance control-based teleoperation system. The detailed control structure is explained in Section III-A.

[2]The motion command from the operator is represented as a homogeneous transformation matrix of the robot's end effector. It is denoted as p_{des} in the control structure or T_{cmd} in the proposed motion reshaping part.

[3]For simplicity, the homogeneous transformation matrix is sometimes denoted by $T = (R, r) \in SE(3)$.

differential of the exponential map and its time derivative; we will utilize them in this letter.

Returning to geometric constraints, if we compute the left translation to start position T_0, relative transformation is obtained as $\widetilde{T}_i = T_0^{-1} T_i$. Then, we can convert the relative transformation set $\{\widetilde{T}_i\}$ to the Lie algebra $\mathfrak{se}(3)$ by using the matrix logarithm and the differential of the exponential [6], [16]. For $\widetilde{T}_i = (\tilde{R}_i, \tilde{r}_i) \in SE(3)$, its corresponding exponential coordinate $\lambda_i = \log(\widetilde{T}_i) = (\eta_i^T, \xi_i^T)^T \in \mathfrak{se}(3) \simeq \mathbb{R}^6$ is calculated as

$$\xi_i = \log\left(\tilde{R}_i\right), \tag{1}$$

$$\eta_i = \mathrm{dexp}_{\xi_i}^{-1}\left(\tilde{r}_i\right). \tag{2}$$

In particular, the corresponding exponential coordinate set $\{\lambda_0, \lambda_1, \ldots, \lambda_i\}$ forms a linear relation in $\mathfrak{se}(3) \simeq \mathbb{R}^6$. For example, the constrained motion in Fig. 2 can be expressed as the corresponding exponential coordinates such that

$$\lambda_i = \begin{cases} \alpha\,(S, 0_{1\times 3})^T & \text{linear motion} \\ \alpha\,(0_{1\times 3}, S)^T & \text{circular motion} \\ \alpha\,(-S \times q + hS, S)^T & \text{helical motion} \end{cases}. \tag{3}$$

Here, $\alpha \in \mathbb{R}$ is a scaling factor; therefore, the geometric constraints shown in Fig. 2 can be specified as a one-parameter, i.e., a six-dimensional vector. From now on, we consider the one-parameter a six-dimensional line vector since the exponential coordinates can be regarded as points on $\mathbb{R}^6$. Thus, the geometric constraint identification problem is converted to a line regression problem.

III. Proposed Methods

A. Overall Control Structure

The basic control structure shown in Fig. 3(a) starts with an admittance controller with feedforward input [7], [21]. To follow the motion commands, the task space PD controller is utilized, and the control torque is obtained as

$$\tau_c = J^T(Ke - D\dot{p}_n) + g, \tag{4}$$

where K and D are the task space stiffness and damping matrices, respectively. $e = p_{\mathrm{des}} - p_n$ is the task space pose error. Since the desired velocity is zero, it can be considered a task space setpoint control during teleoperation.

Moreover, the nonlinear robust inner-loop compensator (NRIC) [22] is utilized. It provides robustness under large model uncertainties and disturbances [20], [23] and further introduces a nominal robot as a proxy in the admittance controller [24]. Under the NRIC structure, the auxiliary torque to make the real robot following the nominal robot is obtained through the PID-type position controller as

$$\tau_a = K_\gamma\left(\dot{e}_{nr} + K_p e_{nr} + K_i \int e_{nr}\right), \tag{5}$$

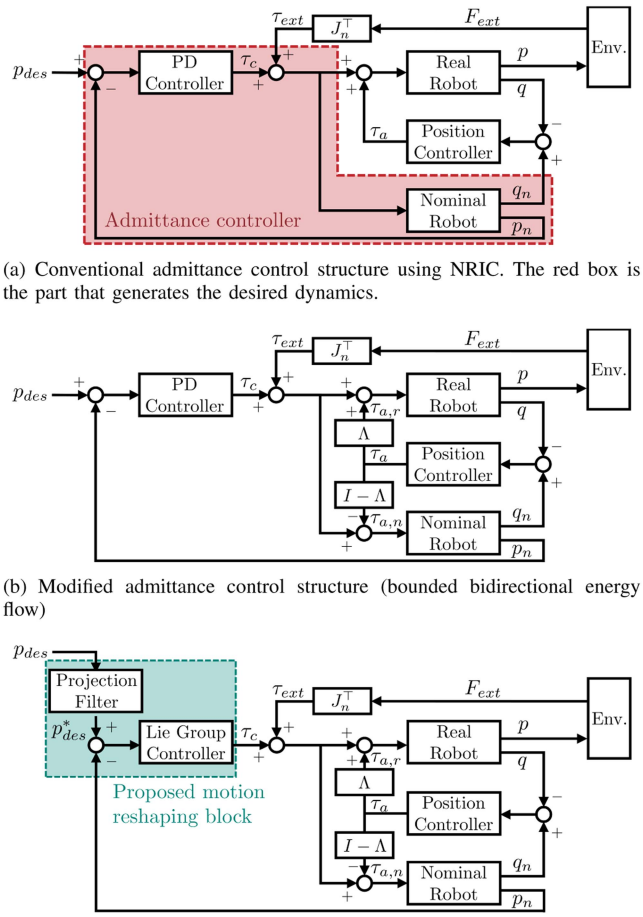
where K_γ, K_p, and K_i are the PID control gain matrices. $e_{nr} = q_n - q$ is the joint position error between the nominal and the real robot.

In the target situation, the conventional admittance controller often leads to instability, such as excessive contact force and oscillation due to a non-collocation problem between the F/T sensor and the proxy [25], [26]. To mitigate this, we use a bidirectional energy flow [24], [27], which adds additional force feedback to the proxy. This feedback loop prevents internal energy generation. However, determining the optimal amount of energy flow is challenging because of the trade-off between the system's passivity and control performance. We apply the concept of a torque bound [23], [27] by setting a constant bound to determine the division of the energy flow. We call this a modified admittance control structure, as shown in Fig. 3(b), and the auxiliary torque that flows into the real robot, denoted by $\Lambda\tau_a$, is calculated as follows:

$$(\Lambda\tau_a)_i = \begin{cases} c_i & \text{if } \tau_{a,i} > c_i \\ \tau_{a,i} & \text{if } |\tau_{a,i}| \le c_i \\ -c_i & \text{if } \tau_{a,i} < -c_i \end{cases}. \tag{6}$$

$\Lambda \in \mathbb{R}^{n\times n}$ is a diagonal matrix for the bound and c_i is the user-defined constant bound.

The proposed control method using the modified admittance control structure is shown in Fig. 3(c). The shadowed box is the proposed motion reshaping block consisting of two parts: a projection filter and a Lie group controller. The projection filter alters the motion commands to be safe under the geometric



(a) Conventional admittance control structure using NRIC. The red box is the part that generates the desired dynamics.

(b) Modified admittance control structure (bounded bidirectional energy flow)

(c) Proposed control structure (motion reshaping method based on projection filter and Lie group setpoint controller)

Fig. 3. Overall control structure.

constraints, and the Lie group controller enforces the robot's motion to follow them. The details are explained in the following subsections.

B. Geometric Constraint Identification

The first step is to identify the geometric constraints. We propose a geometric constraint identification method that exploits the linearity assumption in Lie algebra, as mentioned in Section II-B. The robot motion trajectory is mapped to the Lie algebra as in (1) and (2), the constraints are specified through the line regression. It uses a weighted singular value decomposition (SVD) on this mapped trajectory. The weighting factor is based on the L2 norm of the measured force and moment, reflecting the likelihood of a large measured wrench when the robot deviates significantly from the constrained path. The proposed weight is a modified version of the methods in [11], [28] to penalize the mapped trajectory with a high exerted force or moment. If we let the measured wrench[4] set $\mathcal{F}_i = \{F_1, F_2, \dots, F_i\}$ for each $F_i = (f_i^T, n_i^T)^T$, then the weighted SVD procedure is as below (7), (8), and (9):

$$w_i = \frac{1}{2}\left(\frac{w_{f_i}^{-1}}{\sum_i^N w_{f_i}^{-1}} + \frac{w_{n_i}^{-1}}{\sum_i^N w_{n_i}^{-1}}\right), \tag{7}$$

where

$$w_{f_i} = \frac{\|f_i\|}{\sum_i^N \|f_i\|} \quad \text{and} \quad w_{n_i} = \frac{\|n_i\|}{\sum_i^N \|n_i\|},$$

$$Z = \begin{bmatrix} \sqrt{w_1}\,(\lambda_1^T - \bar{\lambda}^T) \\ \sqrt{w_2}\,(\lambda_2^T - \bar{\lambda}^T) \\ \vdots \\ \sqrt{w_N}\,(\lambda_N^T - \bar{\lambda}^T) \end{bmatrix} = U\Sigma V^T, \tag{8}$$

where

$$\bar{\lambda} = \frac{\sum_i^N w_i \lambda_i}{\sum_i^N w_i}. \tag{9}$$

Here, $Z \in \mathbb{R}^{N\times 6}$ is the data matrix that stacks each exponential coordinate $\lambda_i \in \mathbb{R}^6$ with the inverse weight $w_i \in \mathbb{R}$. On (8) and (9), $\bar{\lambda} \in \mathbb{R}^6$ is a point on the line and $l = (\eta^T, \xi^T)^T \in \mathbb{R}^6$ is a line vector that is the first column of the matrix V, resulting in a line representation[5] in $\mathbb{R}^6$. Furthermore, we can extract screw parameters such as the closest location on the screw axis from the coordinate frame q, the direction of the screw axis S, and the pitch along the screw axis h as shown in Fig. 4. They are computed by

$$q = \frac{\xi \times \eta}{\|\xi\|^2} \in \mathbb{R}^3, \tag{10}$$

$$S = \begin{cases} \frac{\eta}{\|\eta\|} := S_\eta \in \mathbb{R}^3 & \text{prismatic case} \\ \frac{\xi}{\|\xi\|} := S_\xi \in \mathbb{R}^3 & \text{otherwise} \end{cases}, \tag{11}$$

$$h = \frac{\xi^T \eta}{\|\xi\|^2}\frac{2\pi}{\|\xi\|} \in \mathbb{R}. \tag{12}$$

[4]We adopt the notation of (body) wrench as $F = (f^T, n^T)^T \in \mathbb{R}^6$ where $f, n \in \mathbb{R}^3$ are exerted force and moment, respectively [6].

[5]In usual, the line is represented by a point p on the line and nonzero direction vector v in $\mathbb{R}^n$, i.e., set of all points which comprised the line $y = p + tv, t \in \mathbb{R}$.

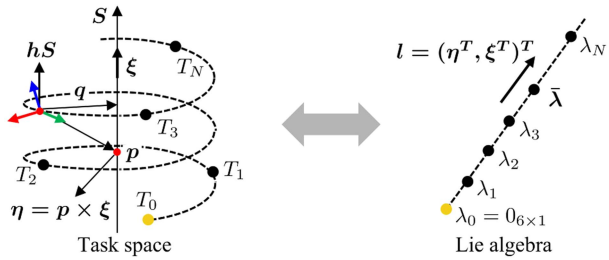


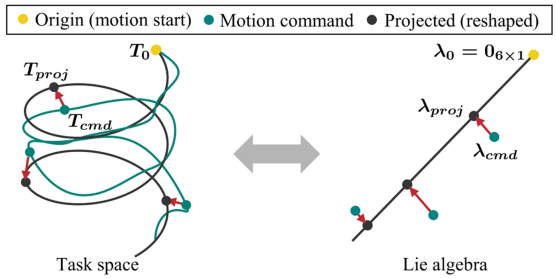
Fig. 4. Relation between the task space and Lie algebra. The motion trajectory under the geometric constraints in task space is represented as the line in Lie algebra. The line vector l is obtained via SVD. Still, physically, the vector l is similar to the normalized screw, or Plücker coordinates [29], thereby η is the moment vector and ξ is the direction vector of a screw axis. p is an arbitrary point on the screw axis that passes through.

Algorithm 1: Constraint Identification Algorithm.

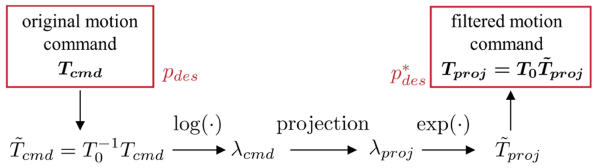
Initialize: motion starting frame T_0
while 1 **do**
  Update traveling distance during teleoperation
  if *traveling distance* $>$ *threshold* **then**
    Sample $\{T_1, \cdots, T_N\}$ and $\{\mathcal{F}_1, \cdots, \mathcal{F}_N\}$
    for $i = 1, 2, \dots N$ **do**
      $\tilde{T}_i = (\tilde{R}_i, \tilde{r}_i) \leftarrow T_0^{-1} T_i$
      $\xi_i \leftarrow \log(\tilde{R}_i)$ (using (1))
      $\eta_i \leftarrow \text{dexp}_{\xi_i}^{-1} \tilde{r}_i$ (using (2))
      Express λ_i as $\lambda_i = (\eta_i^T, \xi_i^T)^T$
      Calculate the inverse weight w_i (using (7))
    Obtain $\bar{\lambda} \leftarrow \frac{\sum_i^N w_i \lambda_i}{\sum_i^N w_i}$ // point on the line
    Construct the data matrix Z
    $U, \Sigma, V \leftarrow$ weighted SVD(Z) (using (8))
    Obtain $l = (\eta^T, \xi^T)^T \leftarrow V(:, 0)$ // line vector
    Compute MED (using (13))
    if $MED_{cur} < MED_{prev}$ **then**
      Update identification result ($\{l_{cur}, \bar{\lambda}_{cur}\}$)
    else
      Keep identification result ($\{l_{prev}, \bar{\lambda}_{prev}\}$)
    Calculate the screw parameters q, S, h (using (10)-(12))
  else
    pass

The aforementioned identification procedure is repeated whenever the traveling distance exceeds the threshold, and the data matrix Z is sampled. For batch-weighted SVD application, we heuristically set the threshold (prismatic case: 10 cm, revolute/helical case: 10 deg in this letter). The identification result is updated if the newly identified line has a smaller Mean Euclidean Distance (MED). The Euclidean distance between the exponential coordinate and the line in $\mathbb{R}^6$ computes the error. The MED is calculated as

$$\text{MED} = \frac{1}{N}\sum_i w_i \|\lambda_i - (P\lambda_i + (I_6 - P)\bar{\lambda})\|, \tag{13}$$



(a) Relation between the task space and Lie algebra in the projection filter



(b) Overall procedure of the filter. The parts in red are the same as the notation of the proposed motion reshaping block mentioned in Fig. 3(c).

Fig. 5. Proposed projection filter.

where I_6 represents a six-dimensional identity matrix. N is the number of the sampled data (empirically set $N = 100$), the inverse weight w_i is calculated by using (7). $P \in \mathbb{R}^{6\times6}$ is the projection matrix used to calculate the error. The precise description of the projection matrix is explained in the next Section III-C. The entire procedure of the identification algorithm is summarized in Algorithm 1.

C. Projection Filter

The next step is to modify the arbitrary motion commands to conform to the identified line l using a projection filter. The projection matrix $P \in \mathbb{R}^{6\times6}$ is computed by

$$P = \frac{ll^T}{l^T l}. \tag{14}$$

Therefore, the arbitrary motion command $\lambda_{cmd} = \log(\tilde{T}_{cmd}) \in \mathfrak{se}(3)$ which is the mapping of $\tilde{T}_{cmd} = T_0^{-1}T_{cmd} \in SE(3)$ will be modified to λ_{proj} which always satisfies the geometric constraint as

$$\begin{aligned}\lambda_{proj} &= P\lambda_{cmd} + (I_6 - P)\bar{\lambda} \\ &= \left(\eta_{proj}^T, \xi_{proj}^T\right)^T. \end{aligned} \tag{15}$$

It is converted back to the task space by

$$T_{proj} = T_0\,\tilde{T}_{proj}, \tag{16}$$

where $\tilde{T}_{proj} = (\tilde{R}_{proj}, \tilde{r}_{proj})$ with $\tilde{R}_{proj} = \exp(\xi_{proj})$, $\tilde{r}_{proj} = \mathrm{dexp}_{\xi_{proj}}(\eta_{proj})$. In summary, the task space motion commands are mapped to the Lie algebra and projected to the identified line. The projected motion command is reconverted to the task space as shown in Fig. 5.

D. Lie Group Setpoint Control

We need to enforce the robot's motion to follow the constrained trajectory. Even if we modify the original motion command T_{cmd} to the filtered motion command T_{proj}, the task space

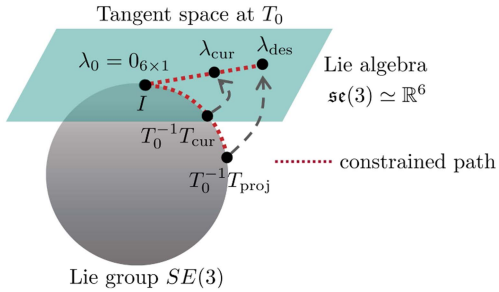


Fig. 6. Schematic of the proposed Lie group setpoint control. PD control between λ_{des} and λ_{cur} in Lie algebra remains the robot's movement to the identified line (the red dashed line in Lie algebra); it ensures that the robot moves within the constrained motion trajectory in the task space (the red dashed line in Lie group).

PD controller fails to follow the identified constraint. This is due to the linear convergence driven by the task space pose error description $e = p_{des} - p_n$. To restrict the robot's movement to the constraint, we suggest a new control scheme, namely *Lie group setpoint control*, which is conceptually illustrated in Fig. 6. The main idea is the PD control action in Lie algebra $\mathfrak{se}(3)$ instead of the task space $SE(3)$. The linear convergence tendency in Lie algebra automatically follows the constrained motion trajectory in the task space. The proposed controller is computed using the following procedures. First, we convert the current and desired transforms to the Lie algebra elements λ_{cur} and λ_{des} as

$$\lambda_{cur} = \log\left(T_0^{-1}T_{cur}\right), \tag{17}$$

$$\lambda_{des} = \log\left(T_0^{-1}T_{proj}\right), \tag{18}$$

where T_{cur} and T_{proj} are the current and filtered desired transforms through the proposed projection filter. T_0 is the starting transform utilized as the left translation. Additionally, we convert the current and desired velocity to Lie algebra as follows:

$$\dot{\lambda}_{cur} = \mathrm{dexp}^{-1}_{-\lambda_{cur}} V_{cur}, \tag{19}$$

$$\dot{\lambda}_{des} = \mathrm{dexp}^{-1}_{-\lambda_{des}} V_{proj} = 0_{6\times1}, \tag{20}$$

where $V_{cur} = (v_{cur}^T, \omega_{cur}^T)^T \in \mathbb{R}^6$ is the current body twist. Then, we can formulate PD control to calculate the reference acceleration in terms of the Lie algebra as

$$\begin{aligned}\ddot{\lambda}_{ref} &= K_{p,se3}(\lambda_{des} - \lambda_{cur}) + K_{v,se3}\left(\dot{\lambda}_{des} - \dot{\lambda}_{cur}\right) \\ &= K_{p,se3}(\lambda_{des} - \lambda_{cur}) - K_{v,se3}\dot{\lambda}_{cur}, \end{aligned} \tag{21}$$

where $K_{p,se3}$ and $K_{v,se3}$ are the P gain and D gain matrices in Lie algebra, respectively. After obtaining the reference acceleration, we convert them to the Lie group $SE(3)$

$$\dot{V}_{ref} = \mathrm{dexp}_{-\lambda_{cur}}\ddot{\lambda}_{ref} + \frac{d}{dt}\mathrm{dexp}_{-\lambda_{cur}}\dot{\lambda}_{cur}. \tag{22}$$

Using inverse dynamic control, we can calculate the control torque as

$$\tau_c = M(q)\ddot{q}_{ref} + C(q,\dot{q})\dot{q} + g(q), \tag{23}$$

where $\ddot{q}_{ref} = J^{-1}(\dot{V}_{ref} - \dot{J}\dot{q})$. $M(q)$, $C(q,\dot{q})$, $J(q)$, and $g(q)$ represent the inertia, Coriolis, Jacobian matrices, and the gravity vector, respectively.

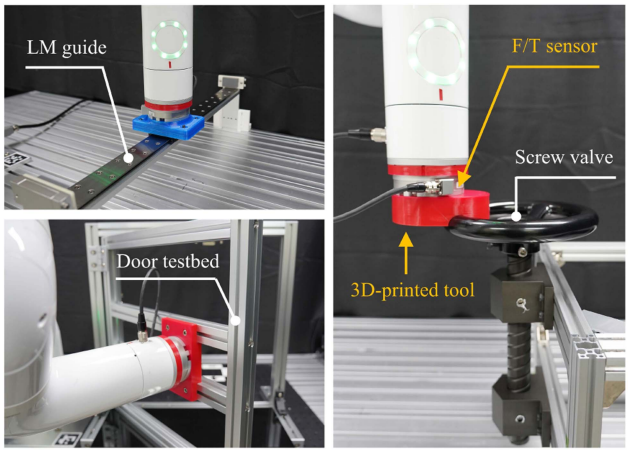


Fig. 7. Experimental setup for various geometric constraints.

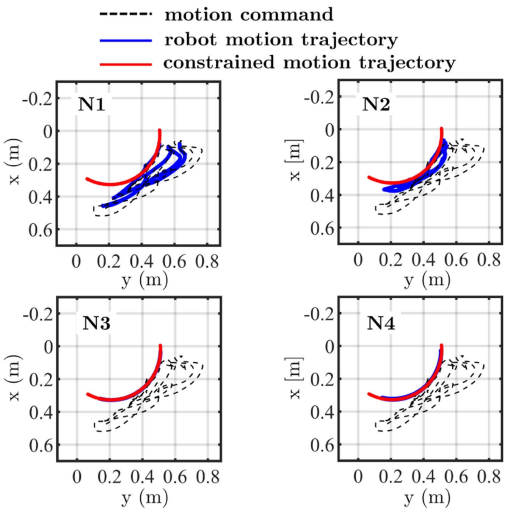


Fig. 8. 2D position plot of the free motion test. The graph on the upper left shows the motion trajectory under task space PD control. The graph on the upper right shows the motion trajectory under task space PD control with virtual fixture enforcement. The lower graphs show the motion trajectory under the proposed motion reshaping method, especially the right graph, which includes bounded bidirectional energy flow.

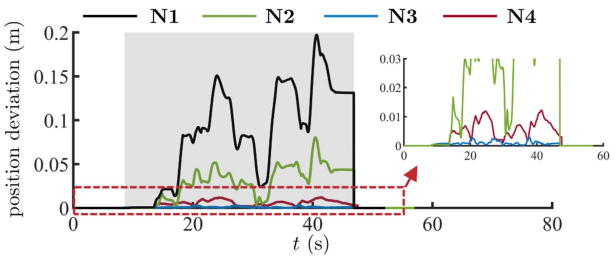


Fig. 9. Positional deviation graph from the constrained motion trajectory. The shaded area represents the period during teleoperation.

IV. Experiment

For the experimental setup, the linear guide, the door test bed, and the screw valve testbed were used to represent the prismatic, revolute, and helical constrained motion, as shown in Fig. 7 . A 6-DoF articulated robot arm (Indy7, Neuromeka) was used for the teleoperated robot. The end effector of the robot arm was rigidly attached to the environment with the 3D-printed tool, so the robot encountered situations that strictly followed each constraint. We utilized a 6-DoF remote controller (Vive Pro controller, HTC) that communicates with the robot via TCP/IP at a 50 Hz sampling rate to generate the desired motion commands. A 6-DoF F/T sensor (RFT80-6A01-A, Robotus) was used to measure the external wrench. In the modified admittance control structure, the user-defined constant bound was set as $c = (10.0, 10.0, 8.0, 6.0, 5.0, 3.0)\,\mathrm{Nm}$, where each component c_i corresponds to the bounded bidirectional energy flow in (6).

A. Geometric Constraint Identification

The proposed identification algorithm extracts the screw parameters from a single execution of the task by the user. The modified admittance control structure is used for identification, and the control parameters are set as follows:

$$K = \mathrm{diag}(100, 100, 100, 16, 16, 16),$$
$$D = \mathrm{diag}(250, 250, 250, 40, 40, 40), \tag{24}$$

where the first three components are for the translation, and the last three are for the orientation. Note that the desired damping was larger than the stiffness to prevent the robot from acting as an underdamped system. To safely extract the constraint in this identification process, vibration feedback is provided to the user through the remote controller when the interaction force/moment exceeds the threshold (30 N/15 Nm). With vibration feedback, the user stops entering the motion command, waits a few seconds to stabilize the robot, and resumes. This encourages the user to operate the robot carefully by considering the contact force indirectly.

We tested eight geometric constraints by adjusting the radius and pitch for the revolute and helical constraints to check the generalizability and versatility of the proposed identification algorithm. The identification results and the ground truth values measured manually in terms of the screw parameters are shown in Table I. The proposed identification algorithm estimates the constraint parameters and even captures the parameter changes.

B. Free Motion Test: Reducing Geometric Constraint Violation

In the second experiment, we used the identification results from the previous experiment and analyzed whether the proposed motion reshaping method improved the robot's adherence to the constrained motion trajectory. To do so, we detached the robot from the environment and remotely operated the robot in free space. We only tested the door case because it allows relatively large and dynamic motion to capture the difference effectively. The controllers verified in the experiment are the following:

1) N1: task space PD control w/o constraint enforcement,
2) N2: task space PD control with VF (virtual fixture),
3) N3: Lie group control with projection filter,
4) N4: Lie group control with projection filter (bounded bidirectional energy flow is added).

N1 represents the method used to generate motion, while N2 is an existing method for enforcing motion constraints. N3 denotes our proposed motion reshaping method, and N4 is a variation of N3, which is specifically modified for rigid contact scenarios. The trajectory recorded using N1 was replicated in N2, N3, and

TABLE I
IDENTIFICATION RESULTS FOR VARIOUS CONSTRAINT PARAMETERS

	constraint type	S_{id}	S_{gt}	$r_{id}(\|\|q_{id}\|\|)$[cm]	$r_{gt}(\|\|q_{gt}\|\|)$[cm]	h_{id}[cm]	h_{gt}[cm]
1	Prismatic	$(-0.9999, 0.0067, 0.0051)^T$	$(-1.0, 0.0, 0.0)^T$	-	-	-	-
2	Revolute	$(0.0085, 0.0018, 0.9999)^T$	$(0.0, 0.0, 1.0)^T$	25.09	25	0.95	0.00
3	Revolute	$(0.0075, 0.0013, 0.9999)^T$	$(0.0, 0.0, 1.0)^T$	30.18	30	1.10	0.00
4	Revolute	$(0.0081, 0.0015, 0.9999)^T$	$(0.0, 0.0, 1.0)^T$	35.09	35	1.21	0.00
5	Helical	$(-0.0013, -0.0021, -0.9999)^T$	$(0.0, 0.0, -1.0)^T$	7.72	7.7	2.58	3.0
6	Helical	$(-0.0073, -0.0013, -0.9999)^T$	$(0.0, 0.0, -1.0)^T$	7.73	7.7	14.64	15.0
7	Helical	$(-0.0021, -0.0027, -0.9999)^T$	$(0.0, 0.0, -1.0)^T$	10.47	10.5	2.39	3.0
8	Helical	$(-0.0069, 0.0034, -0.9999)^T$	$(0.0, 0.0, -1.0)^T$	10.48	10.5	14.69	15.0

Note that q, S, and h are screw parameters in (10), (11), and (12); only the direction of the screw axis S is converted to the robot base frame for a better comparison. The radius r for the revolute and helical constraints can be calculated by taking the L2 norm to the location of the screw axis q. $(\cdot)_{id}$ and $(\cdot)_{gt}$ mean the identification result and the ground truth value, respectively.

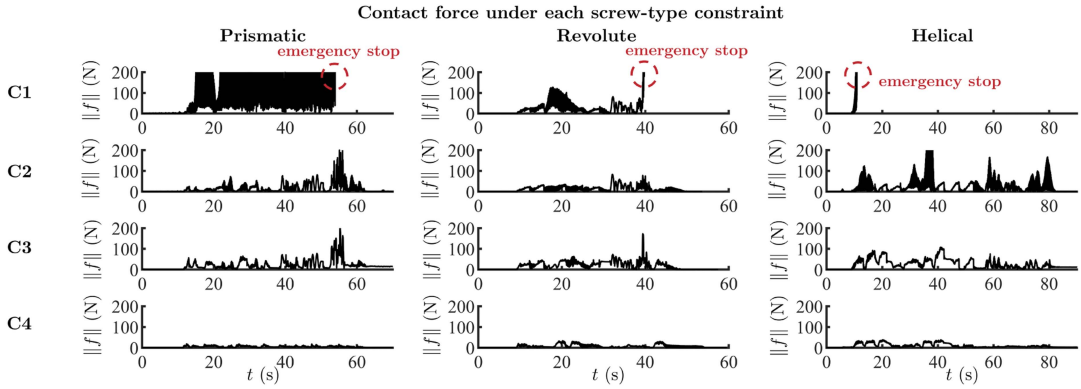


Fig. 10. The results of the interaction test include graphs showing contact forces under prismatic, revolute, and helical constraints in the first, second, and third columns, respectively.

N4 for a fair comparison. The control parameters for N1 and N2 were set as follows:

$$\begin{aligned} K &= \mathrm{diag}(250, 250, 250, 40, 40, 40), \\ D &= \mathrm{diag}(250, 250, 250, 40, 40, 40), \\ K_{vf} &= \mathrm{diag}(500, 500, 500, 120, 120, 120), \\ D_{vf} &= \mathrm{diag}(100, 100, 100, 24, 24, 24), \end{aligned} \tag{25}$$

where K_{vf} and D_{vf} are the PD gains of the VF. The control parameters for N3 and N4 were set as follows:

$$\begin{aligned} K_{p,se3} &= \mathrm{diag}(45, 45, 45, 45, 45, 45), \\ K_{v,se3} &= \mathrm{diag}(45, 45, 45, 45, 45, 45). \end{aligned} \tag{26}$$

Fig. 8 presents the free motion plots. In case N1, the robot significantly deviated from the constraint, following the motion commands without considering the constraint. For case N2, the virtual fixture enforcement led to better adherence to the constraint but did not fully restrict robot motion [13].

In contrast, in cases N3 and N4, the robot was confined to the constraint due to the proposed motion reshaping method. As a result, N3 and N4 showed significantly reduced deviations compared to N1 and N2, as shown in Fig. 9. In particular, N4 showed a slightly higher deviation than N3, a consequence attributed to control performance degradation [23]. However, N4 is chosen for further experiments because the direct application of N3 for the interaction task is not feasible due to the non-collocation problem as discussed in Section III-A.

C. Interaction Test: Reducing Contact Force

Finally, we investigated the contact force when the robot is rigidly attached to each constraint. The controllers verified in the experiment are the following:

1) C1: conventional admittance control (Fig. 3(a)),
2) C2: conventional admittance control with VF,
3) C3: modified admittance control (Fig. 3(b)),
4) C4: the proposed control method (Fig. 3(c)).

For a fair comparison, the trajectory recorded using C4 is replicated in C1, C2, and C3. The control parameters were the same as those in the previous experiment.

Fig. 10 shows the results. In case C1, the robot exhibited unstable behavior, resulting in severe oscillations of the contact force. In addition, the robot encountered catastrophic failures, such as emergency stops, in all constraints due to excessive contact force. On the contrary, C2 and C3 exhibited reduced oscillatory behavior and improved stability due to VF enforcement and additional force feedback, respectively. However, in both C2 and C3, the direct input of dangerous motion commands to the robot still resulted in large contact forces, which had the potential to cause catastrophic failures. The proposed control method, C4, successfully prevented excessive contact forces by filtering dangerous motion commands and aligning them with the constraint. Notably, C4 showed a lower contact force than C3, although it had the same control structure as C3. Therefore, the reduced contact force is inherited from the proposed motion reshaping block, validating that our method can prevent excessive contact force and avoid catastrophic failures during teleoperation.

V. Conclusion

This letter presents a teleoperation framework designed for tasks with geometric constraints, which forms a linear relation in Lie algebra. Our framework effectively estimates these constraints, even in cases where the screw parameters are variable. Building upon the identification results, we introduce a motion reshaping method employing a projection filter and a Lie group setpoint control. This approach ensures that the robot's motion remains within the identified constraints. The experimental results validate that the proposed method limits the robot's motion, preventing excessive contact forces and avoiding catastrophic failures.

The remaining challenge is the requirement for precise motion commands from the human operator during the constraint identification phase. However, it is essential to note that once the constraints are identified, the user can enter arbitrary motions, which are then reshaped using our proposed method. Integrating constraint identification with motion reshaping is an area for future development. In addition, this letter treats geometric constraints that satisfy the linearity assumption in Lie algebra. Thus, the proposed identification method should be extended to other constraints.

Acknowledgment

The authors thank Seohyun Choi for assisting with the experiment procedure and validation.

References

[1] T. Klamt et al., "Supervised autonomous locomotion and manipulation for disaster response with a centaur-like robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 1–8.

[2] J.-G. Ge, "Programming by demonstration by optical tracking system for dual arm robot," in *Proc. IEEE ISR*, 2013, pp. 1–7.

[3] A. Pettinger, C. Elliott, P. Fan, and M. Pryor, "Reducing the teleoperator's cognitive burden for complex contact tasks using affordance primitives," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 11513–11518.

[4] Y. Wu, P. Balatti, M. Lorenzini, F. Zhao, W. Kim, and A. Ajoudani, "A teleoperation interface for loco-manipulation control of mobile collaborative robotic assistant," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3593–3600, Oct. 2019.

[5] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[6] J. Park and W.-K. Chung, "Geometric integration on Euclidean group with application to articulated multibody systems," *IEEE Trans. Robot.*, vol. 21, no. 5, pp. 850–863, Oct. 2005.

[7] A. Q. Keemink, H. v. d. Kooij, and A. H. Stienen, "Admittance control for physical human–robot interaction," *Int. J. Robot. Res.*, vol. 37, no. 11, pp. 1421–1444, 2018.

[8] F. Ficuciello, L. Villani, and B. Siciliano, "Variable impedance control of redundant manipulators for intuitive human–robot physical interaction," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 850–863, Aug. 2015.

[9] D. Bazzi, F. Roveda, A. M. Zanchettin, and P. Rocco, "A unified approach for virtual fixtures and goal-driven variable admittance control in manual guidance applications," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 6378–6385, Oct. 2021.

[10] G. Subramani, M. Zinn, and M. Gleicher, "Inferring geometric constraints in human demonstrations," in *Proc. Conf. Robot Learn.*, 2018, pp. 223–236.

[11] G. Subramani, M. Gleicher, and M. Zinn, "Recognizing geometric constraints in human demonstrations using force and position signals," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 1252–1259, Apr. 2018.

[12] R. Martín-Martín, C. Eppner, and O. Brock, "The RBO dataset of articulated objects and interactions," *Int. J. Robot. Res.*, vol. 38, no. 9, pp. 1013–1019, 2019.

[13] S. A. Bowyer, B. L. Davies, and F. R. Y. Baena, "Active constraints/virtual fixtures: A survey," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 138–157, Feb. 2014.

[14] M. Selvaggio, G. Notomista, F. Chen, B. Gao, F. Trapani, and D. Caldwell, "Enhancing bilateral teleoperation using camera-based online virtual fixtures generation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 1483–1488.

[15] V. Pruks and J.-H. Ryu, "A framework for interactive virtual fixture generation for shared teleoperation in unstructured environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 10234–10241.

[16] J. Park and K. Kim, "Tracking on lie group for robot manipulators," in *Proc. 11th Int. Conf. Ubiquitous Robots Ambient Intell.*, 2014, pp. 579–584.

[17] N. D. Ratliff, J. Issac, D. Kappler, S. Birchfield, and D. Fox, "Riemannian motion policies," 2018, *arXiv:1801.02854*.

[18] P. Liu, D. Tateo, H. B. Ammar, and J. Peters, "Robot reinforcement learning on the constraint manifold," in *Proc. Conf. Robot. Learn.*, 2022, pp. 1357–1366.

[19] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th Eur. Control Conf.*, 2019, pp. 3420–3431.

[20] D. Lee, D. Ko, W. K. Chung, and K. Kim, "Quadratic programming-based task scaling for safe and passive robot arm teleoperation," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 4, pp. 1937–1945, Aug. 2022.

[21] M. J. Kim and W. K. Chung, "Design of nonlinear $\mathcal{H}_\infty$ optimal impedance controllers," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1972–1978.

[22] M. J. Kim, Y. Choi, and W. K. Chung, "Bringing nonlinear $\mathcal{H}_\infty$ optimality to robot controllers," *IEEE Trans. Robot.*, vol. 31, no. 3, pp. 682–698, Jun. 2015.

[23] D. Ko, D. Lee, W. K. Chung, and K. Kim, "Bounded compensation with friction estimation for accurate motion tracking and compliant behavior of industrial manipulators," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 5235–5241.

[24] M. J. Kim, W. Lee, C. Ott, and W. K. Chung, "A passivity-based admittance control design using feedback interconnections," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 801–807.

[25] M. Dohring and W. Newman, "The passivity of natural admittance control implementations," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2003, vol. 3, pp. 3710–3715.

[26] F. Ferraguti, C. T. Landi, L. Sabattini, M. Bonfe, C. Fantuzzi, and C. Secchi, "A variable admittance control strategy for stable physical human–robot interaction," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 747–765, 2019.

[27] R. Kikuuwe, "Torque-bounded admittance control realized by a set-valued algebraic feedback," *IEEE Trans. Robot.*, vol. 35, no. 5, pp. 1136–1149, Oct. 2019.

[28] C. M. Shakarji and V. Srinivasan, "Theory and algorithms for weighted total least-squares fitting of lines, planes, and parallel planes to support tolerancing standards," *J. Comput. Inf. Sci. Eng.*, vol. 13, no. 3, 2013, Art. no. 031008.

[29] Y.-B. Jia, "Plücker coordinates for lines in the space," Problem Solver Techniques for Applied Computer Science, Com-S-477/577 Course Handout, vol. 3, 2020.