2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Relative Geometrical Constraint on Finger Motion for Dexterous Teleoperation of Multifingered Hand

Yohei Kitahara[1] and Manoj Bhadu[1]

***Abstract*— This paper presents a method that enables intuitive and stable in-hand manipulation in teleoperation of a multifingered robotic hand. Our method comprehensively handles the fundamental components of in-hand manipulation, such as object pose control, finger sliding on an object, and finger gaiting. This is achieved by constraining the operator's finger commands relative to the intended object motion, preventing their penetration into the object. Moreover, this method does not rely on any visual-based sensors and instead utilizes the six-axis force-torque sensors at the fingertips. The proposed method was evaluated with hardware through in-hand manipulation of a screwdriver, and the intuitive manipulation with high tracking performance of the operator's finger motion was demonstrated. Additionally, we present imitation learning results using the data collected by the proposed method. Policy learned with constrained finger motion as expert action maintained the performance even without the constraint during policy execution. This shows that the dataset is scalable as it is usable even without implementing the proposed constraint.**

I. Introduction

Teleoperation of robots enables working in remote locations and interacting with objects without being on the spot in person. In dealing with unpredictable situations, human judgment is still essential, and many technologies that target the intuitive teleoperation of robots have been investigated. Especially, for humans to perform intuitive interaction with objects, the teleoperation of a human-like multifingered robotic hand is important. Therefore, intuitive object manipulation with a teleoperated multifingered robotic hand has been one of the focused research areas since the early development of robotic hands [1], [2], and, through the years, many advancements have been made [3], [4], [5].

At the same time, certain tasks that do not require advanced human judgment are expected to be automated to reduce the burden on human operators. Object manipulation data obtained through teleoperation is of great importance in achieving the automation and is expected to be used in the field of learning from demonstrations. By enabling intuitive interaction with objects, expert-level manipulation data can be collected in the short term. Therefore, a method for intuitive object manipulation can also contribute to data collection for learning purposes.

Based on the classification in [3], object manipulation has two categories depending on whether they accompany the interaction with the external environment or not: object interaction, such as peg-in-holes, and in-hand manipulation, respectively. In this paper, for the first step, we focus on in-hand manipulation that does not interact with the external environment. In-hand manipulation requires three fundamental components: finger-gaiting, object pose manipulation, and finger sliding on the object surface [7]. Therefore, a framework capable of handling these three components is needed.

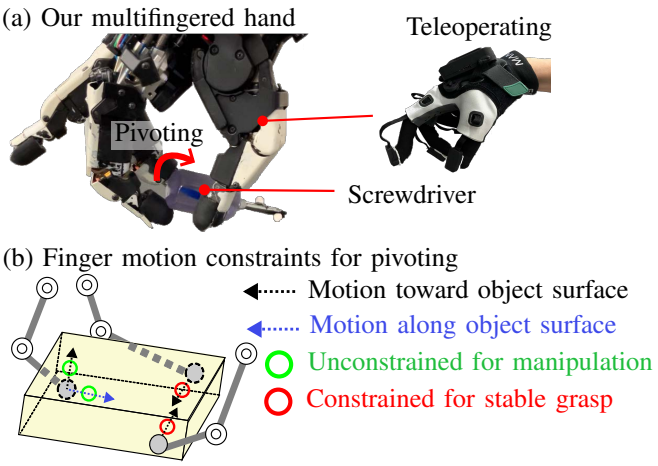


Fig. 1: (a) shows our multifingered hand [6] teleoperated using MANUS^{TM} Quantum Mocap Metagloves. (b) illustrates the necessary constraints on finger motions for pivoting. Two groups of fingers need to coexist; the motions of the thumb and middle toward the object surface must be constrained, and the motion of the index should not be constrained to perform object pose manipulation and sliding.

In teleoperation, the operator's finger joint angles are commonly measured using a data glove [8] or a vision-based approach [9], and mapped to those of a robotic hand. Although these method appears effective for intuitive hand pose control in the absence of contacts with object, they face limitations when manipulating objects stably. This is because it is difficult for the operator to estimate the contact forces generated due to the discrepancy between the commanded and actual joint angles. Currently, several methods have been proposed to achieve intuitive and stable object pose control by following the operator's finger motion [10], [11], [12]. However, a control scheme that comprehensively handles the fundamental components in accordance with operator's finger motion has not yet been proposed.

In this paper, we propose a method that stably and comprehensively handles the fundamental components. It constrains the operator's finger motion relative to the object motion intended by the operator, which is the novel aspect of our method. In addition, since objects are easily occluded by fingers and visual object recognition tends to be unstable

[1]Honda R&D Co., Ltd., 8-1 Hon-cho, Wako, Saitama, Japan yohei_kitahara@jp.honda

during in-hand manipulation, the constraint does not rely on visual-based sensors and instead utilizes six-axis force-torque sensors at the fingertips. Furthermore, to demonstrate the use of data collected by our proposed method, we present imitation learning results based on it.

A. Related works

Some approaches succeeded in stably manipulating object pose in accordance with the operator's finger motion without requiring the object's parameters and any object detection methods [10], [11], [12]. However, these methods did not consider sliding motion. In other works, a sliding motion was achieved to move a rigidly held object [13] or an end-effector [14] on the surface of another object. These works constrained the position commands to avoid penetrating the surface, thereby mitigating excessive contact forces and enabling smooth sliding. However, in object manipulation by the teleoperated multifingered hand, there are some situations where position commands toward the surface should be allowed. For example, the object pose control called pivoting requires the coexistence of fingers whose position commands toward the surface are constrained and unconstrained, as shown in Fig. 1. Therefore, a constraint on penetration that can handle both object pose manipulation and sliding motion in a unified manner is necessary, but such a method has not yet been proposed. To address this, we propose a novel method that geometrically constrains the operator's fingertip position relative to the object motion intended by the operator.

In the context of manipulation learning, some works utilize control algorithms to ensure grasp stability within reinforcement learning frameworks [15], [16], or enhance the robustness of learned model execution [17]. However, these controllers are required during policy execution, which limits data scalability to other tasks or hardware. While Khandate *et al.* [18] demonstrated controller-free policy deployment, it requires sub-skill controllers specific to each in-hand manipulation component during training. In contrast, we demonstrate a strategy for imitation learning that enables policy execution without the proposed constraint, covering the three fundamental components.

B. Statement of contributions

(i) This work proposes a method that comprehensively handles the fundamental components of in-hand manipulation by constraining the operator's finger commands relative to the intended object motion. Moreover, this method does not use vision-based sensors and instead utilizes six-axis force-torque sensors at the fingertips. (ii) We provide experimental results on hardware to show the effectiveness of the proposed method. (iii) Finally, we demonstrate imitation learning results on hardware using the constrained fingertip position command as expert action. The learned policy maintained the performance during execution even without the constraint. This shows that the collected expert dataset remains usable without implementing the constraint, thereby enhancing the scalability of the dataset.

TABLE I: Mathematical Notation

Notation	Unit	Definition
$k \in \mathbb{Z},\ k \geq 0$	-	Discrete time step.
$i = \{\text{th, in, mi}\}$	-	The thumb, index, and middle fingers.
$\boldsymbol{x}^i_{o,k} \in \mathbb{R}^3$	mm	The i-th fingertip position commands from the operator at time step k.
$\boldsymbol{x}^i_{r,k} \in \mathbb{R}^3$	mm	The current robot's fingertip positions.
$\boldsymbol{x}^i_{p,k} \in \mathbb{R}^3$	mm	The generated robot's fingertip position for object pose control based on IOP (Intended Object Pose).
$\boldsymbol{x}^i_{g,k} \in \mathbb{R}^3$	mm	The final commands of the robot's fingertip positions from the proposed constraint.
$\boldsymbol{f}^i_{cmd,k} \in \mathbb{R}^3$	N	The robot's fingertip force command for making grasp force equilibrium.
$\boldsymbol{f}^i_{act,k} \in \mathbb{R}^3$	N	The robot's fingertip force on the object, obtained from sensor measurements.

Subscripts o, r, p, and g denote (o)perator, (r)obot, IOP(p)-based, and command from the (g)eometrical constraint, respectively.

II. Proposed Method

We propose a method that enables stable execution of three fundamental components of in-hand manipulation, using the measurements from six-axis force-torque sensors at the fingertips. By considering the object motion intended by the operator, we propose a relative geometrical constraint that enables the in-hand object manipulation the operator aims for. The mathematical notation used in the following explanation is summarized in the Table I. We assume rigid contact between the fingers and the object and that each finger has three degrees of freedom to control the fingertip position and force in Cartesian space.

Our control pipeline has two paths as shown in Fig. 2. The obtained operator's fingertip positions $\boldsymbol{x}^i_{o,k}$ are processed through both the Grasp Force Calculation path and Fingertip Position Constraint path. In the Grasp Force Calculation path, based on the method in [6], fingertip forces $\boldsymbol{f}^i_{cmd,k}$ are calculated for grasping fingers so that the resultant force makes equilibrium based on a target grasping strength[1]. In the Fingertip Position Constraint path, the fingertip commands are constrained using our novel approach, depicted as the green block. This prevents the fingertip position command from penetrating the object's surface and disturbing the latter contact force control, ensuring stable in-hand manipulation. The latter part consists of admittance control based on fingertip force measurements $\boldsymbol{f}^i_{act,k}$ for contact force control, inverse kinematics converting the fingertip position to the joint angle, and joint impedance control for the joint angle tracking. To enable both sliding and finger motions toward the object's surface for pose manipulation while avoiding penetration, our method constrains the operator's finger command relative to the object motion intended by the operator.

A. Judge Grasp/Release Fingers

In the Grasp Force Calculation path, at first, we determine the fingers to which the grasping force is applied. To handle

[1]We assume that the object is manipulated quasi-statically.

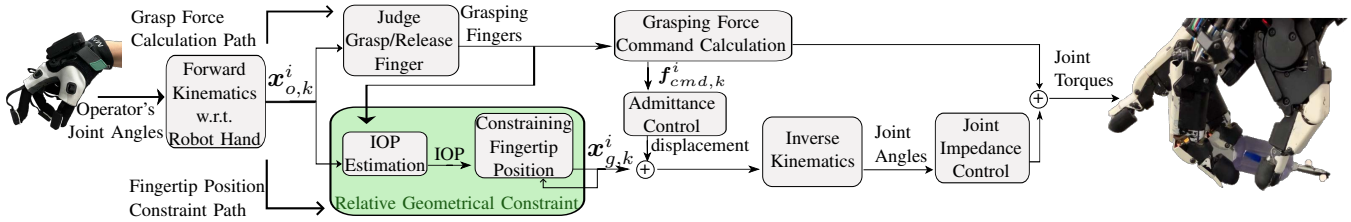


Fig. 2: Block diagram of our control pipeline. The green block depicts our novel constraint.

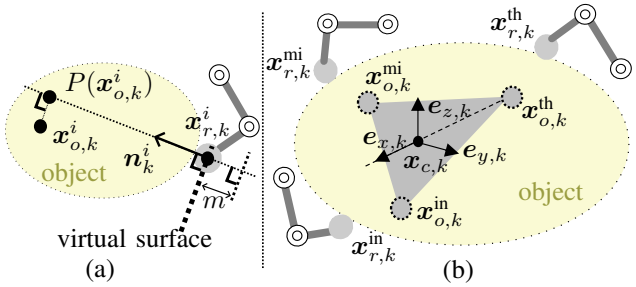


Fig. 3: (a) describes the penetration amount of $\boldsymbol{x}^i_{o,k}$, $P(\boldsymbol{x}^i_{o,k})$, for a certain finger. (b) illustrates a geometrical relationship used to obtain the operator's intended object pose (IOP).

objects of unknown shape, we assume a virtual surface of an object (described in Fig. 3(a)) based on the robot's actual fingertip position $\boldsymbol{x}^i_{r,k}$ and the unit normal vector $\boldsymbol{n}^i_k$ at the contact point (hereafter contact normal, positive direction towards the object from a finger). The contact normal is calculated from the contact point estimated based on the six-axis force-torque sensor and the spherical shape of the fingertip [19]. The surface is tangential to the contact normal $\boldsymbol{n}^i_k$ and includes the robot's actual fingertip position. Here, letting the penetration amounts of $\boldsymbol{x}^i_{o,k}$, $P(\boldsymbol{x}^i_{o,k})$, be the distance from $\boldsymbol{x}^i_{r,k}$ to the projection of $\boldsymbol{x}^i_{o,k}$ onto the contact normal direction, we define,

$$P(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{x}^i_{r,k}) \cdot \boldsymbol{n}^i_k. \tag{1}$$

For determining the grasping fingers, a release threshold $m \in \mathbb{R}^+$ is set to prevent unintended finger releases during teleoperation. A finger is considered as grasping finger if one of the following conditions is satisfied.

- The fingertip position $\boldsymbol{x}^i_{o,k}$ is in the object side from the virtual object surface.
- The distance between the fingertip position $\boldsymbol{x}^i_{o,k}$ and the virtual object surface is within the release threshold.

That is, this is when the following one condition is satisfied:

$$P(\boldsymbol{x}^i_{o,k}) \geq -m. \tag{2}$$

The fingertip forces $\boldsymbol{f}^i_{cmd,k}$ are calculated to make force equilibrium among the grasping fingers. Conversely, (2) being not satisfied indicates that the operator intends to release the corresponding fingers. Then, the controller excludes these fingers from the group of fingers that make force equilibrium. Finally, the operator can move the fingers to their desired positions and release the fingers.

B. Proposed Constraint

This section describes the constraints on the operator's fingertip positions $\boldsymbol{x}^i_{o,k}$ for quasi-static manipulation processed in the green block in Fig. 2. This constraint keeps the penetration amount of its output $\boldsymbol{x}^i_{g,k}$, $P(\boldsymbol{x}^i_{g,k})$, being zero. Since we use a multifingered hand, the proposed constraint needs to consider multiple end-effectors. The constraints to the motions of individual end-effectors can not be imposed before the direction of the object's 6D motion is derived. Moreover, the 6D motion needs to be derived in accordance with the operator's intended object motion. Therefore, the controller first estimates the operator's intended object pose (IOP). Then, it constrains operator's fingertip positions relative to the object which is manipulated according to IOP. In the following sections, we first describe how to estimate IOP.

1) IOP Estimation: Let N be the set of the grasping fingers, and n be the number of them. IOP is estimated using operator's fingertip positions of these grasping fingers. Depending on the task, we can change from which grasping fingers the IOP is derived, but here, we consider a simple example. The translation of IOP, $\boldsymbol{d}_k \in \mathbb{R}^3$, is defined as the center point, $\boldsymbol{x}_{c,k}$, of operator's fingertip positions of grasping fingers:

$$\boldsymbol{x}_{c,k} = \frac{1}{n}\sum_{i \in N} \boldsymbol{x}^i_{o,k}, \text{ therefore } \boldsymbol{d}_k = \boldsymbol{x}_{c,k}. \tag{3}$$

Regarding the orientation of IOP, we first explain the three-fingers case, as described in Fig. 3(b). Let a rotation matrix $\boldsymbol{R}_k \in SO(3)$ represent the orientation of the IOP in the world frame. Then, each column vector of $\boldsymbol{R}_k$ corresponds to each basis vector of the IOP's frame. We define the unit vector $\boldsymbol{e}_{x,k}$ from the operator's thumb position $\boldsymbol{x}^{\text{th}}_{o,k}$ to the center point $\boldsymbol{x}_{c,k}$ as the x-axis:

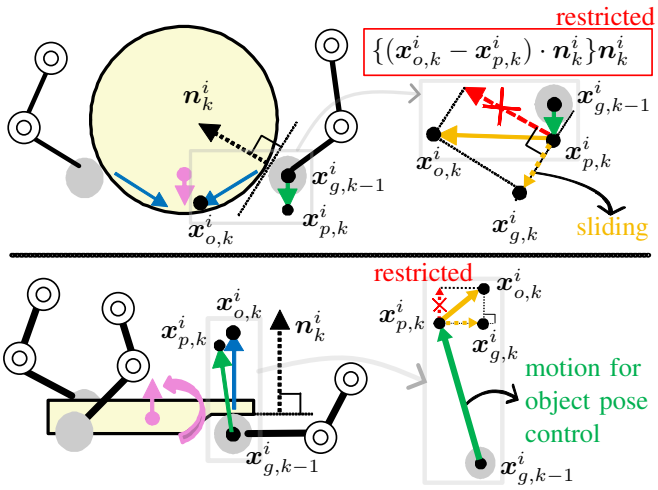
$$\boldsymbol{e}_{x,k} = \frac{\boldsymbol{x}_{c,k} - \boldsymbol{x}^{\text{th}}_{o,k}}{\|\boldsymbol{x}_{c,k} - \boldsymbol{x}^{\text{th}}_{o,k}\|_2}. \tag{4}$$

The unit vector $\boldsymbol{e}_{z,k}$, perpendicular to the plane formed by the operator's fingertip positions $\boldsymbol{x}^i_{o,k}$, is used as the z-axis:

$$\boldsymbol{e}_{z,k} = \frac{(\boldsymbol{x}^{\text{mi}}_{o,k} - \boldsymbol{x}^{\text{th}}_{o,k}) \times (\boldsymbol{x}^{\text{in}}_{o,k} - \boldsymbol{x}^{\text{th}}_{o,k})}{\|(\boldsymbol{x}^{\text{mi}}_{o,k} - \boldsymbol{x}^{\text{th}}_{o,k}) \times (\boldsymbol{x}^{\text{in}}_{o,k} - \boldsymbol{x}^{\text{th}}_{o,k})\|_2}. \tag{5}$$

Then, the unit vector that corresponds to the y-axis, $\boldsymbol{e}_{y,k}$, is defined as $\boldsymbol{e}_{y,k} = \boldsymbol{e}_{z,k} \times \boldsymbol{e}_{x,k}$. Finally, we have $\boldsymbol{R}_k = \begin{pmatrix}\boldsymbol{e}_{x,k} & \boldsymbol{e}_{y,k} & \boldsymbol{e}_{z,k}\end{pmatrix}$.

For the two-fingers case, we consider using the thumb and middle fingers. Using (3) gives us the translation $\boldsymbol{d}_k$ as the



From time $k-1$ to k,
change of operator's fingertip position
change of IOP
transformed by IOP change (eq. (6))
$\boldsymbol{x}^i_{g,k} = \boldsymbol{x}^i_{p,k} +$ (first line in eq. (7))

Fig. 4: Visualization of the proposed constraint. Black dots represent positions of $\boldsymbol{x}^i_{o,k}$, $\boldsymbol{x}^i_{p,k}$, $\boldsymbol{x}^i_{g,k-1}$, $\boldsymbol{x}^i_{g,k}$. In the top case, the operator is attempting a transition from pinching to power grasp by sliding the fingers along the object, and in the bottom case, attempting to pivot the object.

middle point of the two fingers. For the orientation matrix $\boldsymbol{R}_k$, the unit vector from the thumb to the middle finger corresponds to $\boldsymbol{e}_{x,k}$. Then, we consider a quaternion $\boldsymbol{r}$ that represents a rotation to convert the x-axis of the world frame to $\boldsymbol{e}_{x,k}$. Finally, $\boldsymbol{R}_k$ is the rotation matrix corresponding to the $\boldsymbol{r}$.

Note that the IOP definition is not limited to the above method. For example, in the above three-fingers case, the x-axis of IOP is the vector from the thumb to the center point, but any finger can be used to define it. Moreover, in the case of more than three fingers, for example, the translation $\boldsymbol{d}_k$ can be calculated using (3). Furthermore, $\boldsymbol{R}_k$ can be computed based on positional relationships of $\boldsymbol{x}^i_{o,k}$, for instance, for the four-fingers case, similarly to [20].

2) Constraining fingertip position: Using the proposed constraint, the fingertip commands from the operator $\boldsymbol{x}^i_{o,k}$ are corrected to $\boldsymbol{x}^i_{g,k}$ as the final result of the constraint. This process is visualized in Fig. 4. First, based on IOP, the commands for manipulating the object pose $\boldsymbol{x}^i_{p,k}$ are computed by transforming the final result at the previous time step, $\boldsymbol{x}^i_{g,k-1}$, using the transformation matrix[2]:

$$\begin{pmatrix} \boldsymbol{x}^i_{p,k} \\ 1 \end{pmatrix} = \begin{pmatrix} \boldsymbol{R}_k & \boldsymbol{d}_k \\ \boldsymbol{0}^T & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{R}_{k-1} & \boldsymbol{d}_{k-1} \\ \boldsymbol{0}^T & 1 \end{pmatrix}^{-1} \begin{pmatrix} \boldsymbol{x}^i_{g,k-1} \\ 1 \end{pmatrix}. \tag{6}$$

Then, we can consider the relative position vector $\boldsymbol{x}^i_{o,k} - \boldsymbol{x}^i_{p,k}$ that expresses the operator's fingertip positions relative to the object during pose manipulation. By adding the sliding component (perpendicular to $\boldsymbol{n}^i_k$, the yellow dashed arrow in Fig. 4) of this vector to $\boldsymbol{x}^i_{p,k}$, the final result of this constraint $\boldsymbol{x}^i_{g,k}$ is obtained, enabling object pose manipulation and sliding in a unified manner:

$$\begin{aligned} \boldsymbol{x}^i_{g,k} &= \boldsymbol{x}^i_{p,k} + (\boldsymbol{x}^i_{o,k} - \boldsymbol{x}^i_{p,k}) - \{(\boldsymbol{x}^i_{o,k} - \boldsymbol{x}^i_{p,k}) \cdot \boldsymbol{n}^i_k\}\boldsymbol{n}^i_k \\ &= \boldsymbol{x}^i_{o,k} - \{(\boldsymbol{x}^i_{o,k} - \boldsymbol{x}^i_{p,k}) \cdot \boldsymbol{n}^i_k\}\boldsymbol{n}^i_k. \end{aligned} \tag{7}$$

The second line in (7) shows that this method restricts the component along $\boldsymbol{n}^i_k$ of the relative position vector $\boldsymbol{x}^i_{o,k} - \boldsymbol{x}^i_{p,k}$, which is $\{(\boldsymbol{x}^i_{o,k} - \boldsymbol{x}^i_{p,k}) \cdot \boldsymbol{n}^i_k\}\boldsymbol{n}^i_k$, depicted as red dashed arrow in Fig. 4. This ensures that only unnecessary penetration of operator's fingertip positions into the object surface, which is not related to both of object pose manipulation and sliding motion, is restricted. This makes the proposed constraint novel.

Fig. 4 illustrates the geometrical relationship among $\boldsymbol{x}^i_{o,k}$, $\boldsymbol{x}^i_{p,k}$, and $\boldsymbol{x}^i_{g,k-1}$, $\boldsymbol{x}^i_{g,k}$ in two representative cases. In the top case, where the operator is attempting a transition from pinching to power grasp by sliding the fingers along the object, the contact normal component of the relative vector (shown as red dashed arrow) is restricted, avoiding penetration. Consequently, the constraint allows the sliding motion along the contact tangential direction to be outputted. For the case where the operator is attempting to pivot the object, shown at the bottom of Fig. 4, $\boldsymbol{x}^i_{p,k}$ for object pose manipulation calculated from the change of IOP is close to the position of $\boldsymbol{x}^i_{o,k}$. Since the relative position vector of them is small, from (7), the constraint on $\boldsymbol{x}^i_{o,k}$ is small as well. As a result, the fingertip motion toward the object surface required for this object pose manipulation can be outputted.

In conclusion, proposed constraint can handle sliding motion and object pose manipulation in a unified manner, including finger motions toward the object surface like pivoting. Additionally, finger gaiting can be handled by switching the grasp/release fingers and applying grasping force according to (2). Therefore, our method can comprehensively handle the three fundamental components of in-hand manipulation.

Furthermore, while the proposed method uses six-axis force-torque sensors, it can also be applied to robotic hands capable of detecting the normal vector at the contact point and the three-axis force on each fingertip.

III. Experiment

Two experiments were conducted on hardware. The first experiment validated the feasibility of the proposed method, performing a task that consists of the three fundamental components of in-hand manipulation. In the second experiment, the utility of the data collected with the proposed method was assessed through imitation learning.

A. Hardware and Software implementation

In the experiments, the Honda multifingered hand [6] shown in Fig. 1 was used. The hand adopts a wire-driven system. A unique dynamic pulley mechanism was designed

[2]At the beginning of grasping ($k = 0$), instead of using (6), $\boldsymbol{x}^i_{p,0}$ is set to the robot's actual fingertip position at initial contact, ensuring it is on the object's surface and without penetration.

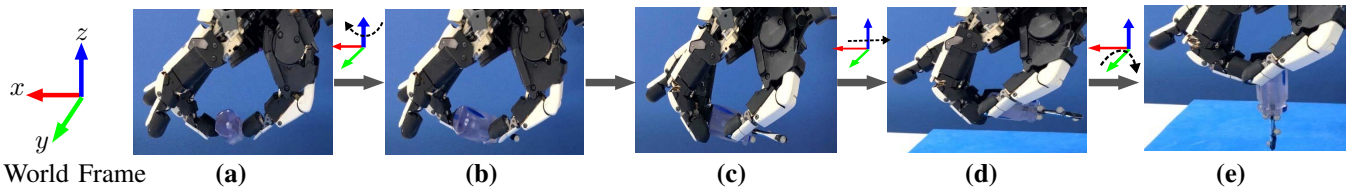


Fig. 5: Target regrasping task. (a) → (b) shows in-grasp manipulation, (b) → (c) demonstrates index contact initialization, (c) → (d) illustrates sliding, and (d) → (e) depicts pivoting. The dashed arrow indicates the object's motion direction.

to achieve a human-like output torque with a human-like size. The hand is a four-fingered structure with four respective joints. The distal and proximal interphalangeal joints are coupled in each finger, and the adduction and abduction axes of the middle and ring are linked. Thus, the hand has a total of 11 degrees of freedom. In the experiments, the thumb, index, and middle fingers with three degrees of freedom were used. Each fingertip is fitted with a 3 mm thick rubber-like skin attached to a rigid frame. The fingertips are equipped with six-axis force-torque sensors. The operator's joint angles were obtained by MANUS™ Quantum Mocap Metagloves for teleoperation, that publishes the operator's joint angles at the frequency of 100 Hz. The controller of the robotic hand shown in Fig. 2 calculates desired joint torques at the frequency of 250 Hz.

For the implementation of the proposed method, the finger release threshold m in (2) was set to 10 mm based on a previously performed pilot test to prevent unintentional finger releases. To calculate IOP, we used the same approach shown in the proposed method except for the $\boldsymbol{d}_k$ of the three-fingers case. Since the index finger motion mainly contributes to the translational motion of the object in the following task setting, we assumed that the translation $\boldsymbol{d}_k$ depended solely on the index finger's motion: $\boldsymbol{d}_k = \boldsymbol{x}^{\text{in}}_{o,k}$, instead of (3). The target grasping strength for making force equilibrium [6] was set to 3 N, considering the weight of the object. The admittance control in Fig. 2 is implemented simply, considering only the proportional gain, $K_p = 1\,\text{mm/N}$. Its output, $displacement = K_p(\boldsymbol{f}^i_{cmd,k} - \boldsymbol{f}^i_{act,k})$.

B. Validation of the Proposed Method

In this experiment, to validate the feasibility of the proposed method, we quantitatively compared the results of in-hand manipulation with and without the proposed method. For a fair comparison of the performance of tracking the operator's motion, which is important for intuitive teleoperation, we use the same recorded operator joint angle trajectory in both cases. The manipulation without the proposed method was performed based on pure joint impedance control following the operator's joint angles.

We designed an experimental task that covered fundamental components of in-hand manipulation [7] in a continuous manner, starting from a pinching posture and ending with grasping a screwdriver handle using a tripod grip. Fig. 5 shows the flow of the designed task, and each fundamental component of in-hand manipulation appeared as follows.

- Finger-gaiting appeared in (b) → (c) as a motion of initializing the index finger's contact,
- object pose control appeared in (a) → (b) and (d) → (e) as in-grasp manipulation and pivoting, respectively, and
- sliding appeared in (c) → (d).

Focusing on the object pose control and sliding task, the penetration amounts of $\boldsymbol{x}^i_{o,k}$, $P(\boldsymbol{x}^i_{o,k})$, and constrained $\boldsymbol{x}^i_{g,k}$ by the proposed constraint, $P(\boldsymbol{x}^i_{g,k})$, were evaluated. The definition of the penetration amount is illustrated in Fig. 3(a). Moreover, we evaluated the effect of the proposed method on the achievement of the desired IOP and sliding amount according to the task. We prepared five recorded operator joint angle trajectories for each task. Fig. 6 and Fig. 7 show plots of one trial out of five for each task.

Finally, to show the versatility of the proposed method for different object shapes, we performed the manipulation tasks sequentially using two additional screwdrivers and, as another object type, a plug adapter. In the proposed method, as the target grasp strength is given and the control relies on the six-axis force-torque sensors, the approach is not specific to object shapes. In this demonstration, the fingers were teleoperated online, where the operator's finger motion is online reproduced with the robotic hand.

1) In-Grasp Manipulation: Regarding the task from (a) to (b) in Fig. 5, the results were compared using the same operator's joint angles trajectory with and without the proposed method. Fig. 6(a) shows that while the maximum value of the penetration amounts was approximately 14 mm for the thumb and 12 mm for the middle finger without the proposed constraint, the constrained $\boldsymbol{x}^i_{g,k}$ effectively reduced the penetration amount to approximately ±2 mm for both fingers. In terms of the achievement of the desired IOP, the blue plot in Fig. 6(b) depicts the change in the angle around the world z-axis of the vector $\boldsymbol{x}^{\text{mi}}_{o,k} - \boldsymbol{x}^{\text{th}}_{o,k}$ representing the IOP orientation, while the pink plot shows the same for the vector $\boldsymbol{x}^{\text{mi}}_{r,k} - \boldsymbol{x}^{\text{th}}_{r,k}$. With regard to approximately 35° of the IOP rotation command, the proposed method achieved a steady-state error of about 2°, but the pure joint impedance control's error was about 20°. Over five trials, the reduced penetration was within ±2 mm for the both fingers. IOP rotation command was on average 37° (max 40°). The steady-state error was on average 2° (max 3°) with the proposed method, and on average 15° (max 20°) without the proposed method. This improvement stemmed from the proposed method's consideration of fingertip forces in task space for object manipulation, rather than relying solely on joint angle space tracking.

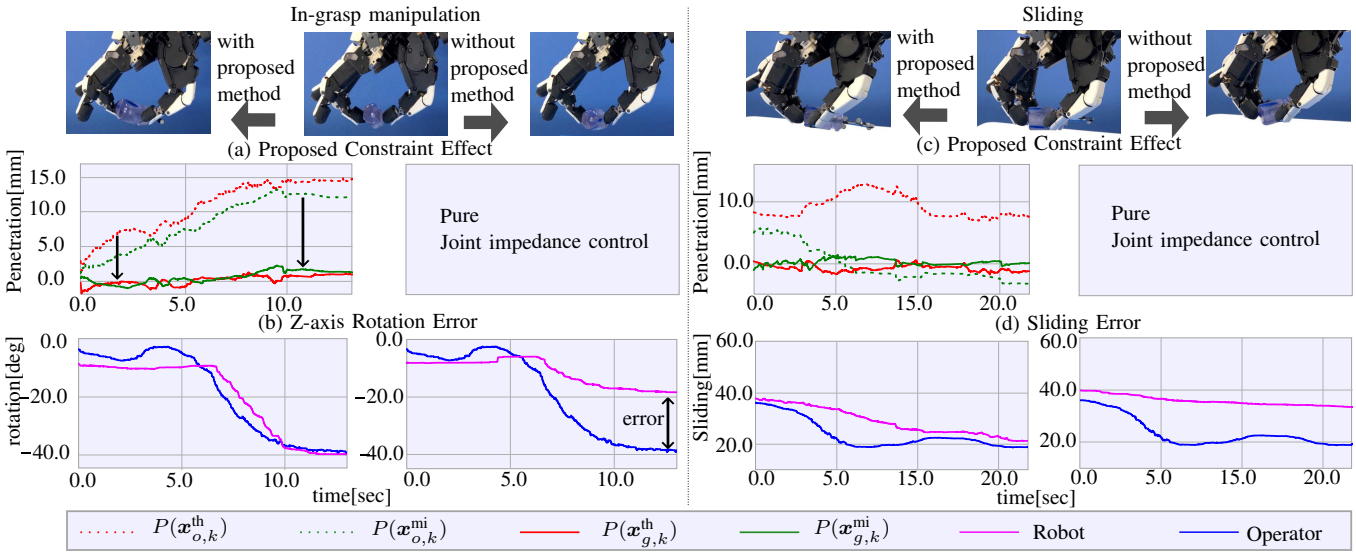


Fig. 6: w/o and w/ proposed method comparison on in-grasp manipulation task and sliding task.

2) Sliding: On the sliding task from (c) to (d) in Fig. 5, the results were compared using the same operator's joint angles trajectory with and without the proposed method. Fig. 6(c) shows that, with the proposed method case, the penetration amounts of the operator's fingertip positions were approximately a maximum of 12 mm for the thumb and 5 mm for the middle finger. On the other hand, by the proposed constraint, these penetrations were reduced to around -2 mm to 1 mm for thumb, and ± 1 mm, for middle, respectively. The sliding performances are shown in Fig. 6(d) . The distance from the midpoint between the thumb and middle finger to the index finger was evaluated according to the operator's fingertip position (blue plot) and the robot's actual fingertip position (pink plot). The blue plot indicates that the operator aimed for a slide of about 16 mm. Using the proposed method, the steady-state error was approximately 2 mm, whereas pure joint impedance control resulted in about 14 mm. Over five trials, the reduced penetration was within -3 mm to 2 mm for thumb and ± 2 mm for middle. The operator aimed to slide 17 mm on average (max 18 mm). The steady-state error was on average 2 mm (max 3 mm) with proposed method, while on average 15 mm (max 16 mm) without proposed method. This was attributed to excessive fingertip contact forces caused by penetration, which generated significant friction that inhibited smooth sliding.

3) Pivoting: For the pivoting task, Fig. 7(a) shows the penetration amounts of the operator's fingertip positions were approximately a maximum of 6 mm for the thumb and 4 mm for the middle finger. On the other hand, the proposed constraint reduced these penetration amounts to around ± 1 mm for both fingers. Fig. 7(b) illustrates the rotation error in the pivoting task. This task with the index finger was characterized by the rotation of $\boldsymbol{e}_{z,k}$ in IOP around the world y-axis (blue plot), with the rotational command of about 35°. The tracking accuracy was validated by calculating the rotation amount using $\boldsymbol{e}_{z,k}$ derived from the robot's actual fingertip positions $\boldsymbol{x}^i_{r,k}$ instead of the operator's fingertip positions

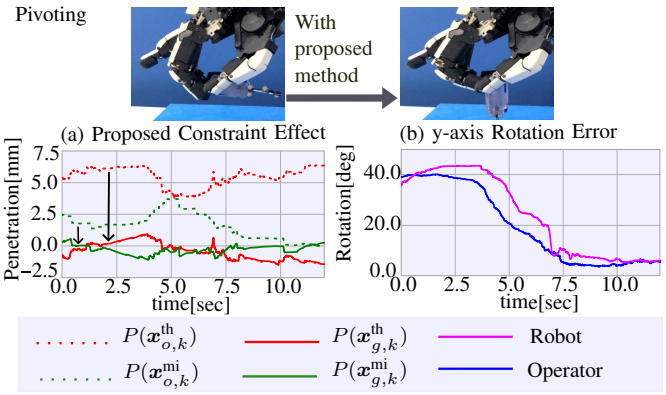


Fig. 7: Proposed method validation on pivoting task.

(pink plot). The proposed method achieves a steady-state error of within 1°. Over five trials, the reduced penetration was within ± 2 mm for both fingers. The rotational command was on average 35° (max 40°). The steady-state error was on average 1° (max 2°).

4) Differently shaped objects: Using the same implementation, we demonstrated the versatility of our method for two other screwdriver grip shapes (textured cylindrical and wavy-surfaced) and a plug adapter. Fig. 8(a) shows the appearances of these objects. Additionally, Fig. 8(b)-(d) depict that the same task as in Fig. 5 can be successfully performed sequentially with these objects. The video recordings of these experiments are included in the supplementary movie.

C. Data Utilization

The second experiment assessed the utility of the data collected by teleoperation with the proposed method. By employing the constrained fingertip position $\boldsymbol{x}^i_{g,k}$ as the expert action in imitation learning, this study examined whether the policy could maintain the performance when executed even without the proposed constraint. The key feature of the proposed constraint is that it allows the coexistence of fingers whose position commands toward the object surface

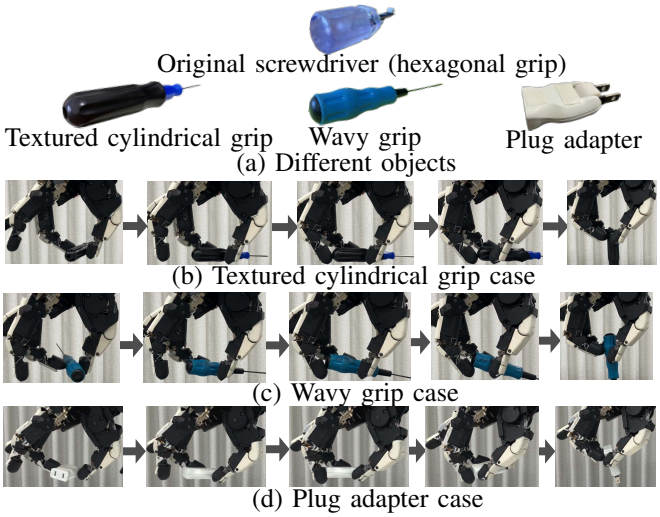


Fig. 8: Versatility of proposed method for differently shaped objects.

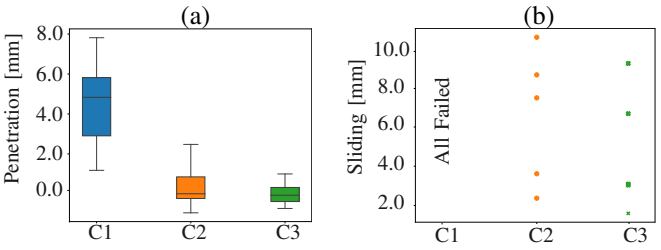


Fig. 9: C1, C2, and C3 were each executed five times. (a) Boxplots of the average penetration of the thumb and middle finger at each time step. (b) The sliding amount on the object. C1 failed all five times, so there is no plot.

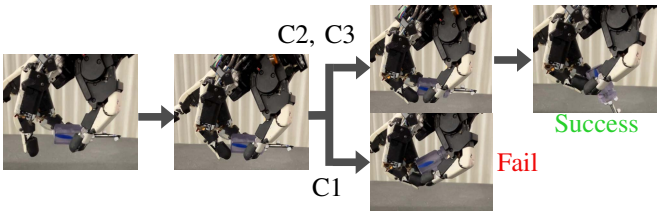


Fig. 10: The top is an example of a policy learned with constrained fingertip position as the expert action. The bottom is an example of executing C1.

are constrained and unconstrained. To demonstrate that the policy can learn this feature, we focused on the procedures that include both constrained and unconstrained fingers: the sequence depicted in Fig. 5 (b) $\rightarrow$ (e). Task success was evaluated across three distinct experimental combinations of our control pipeline in Fig. 2 and learned policies, where a policy's output was used as the operator command in the pipeline on each combination. The configurations are as follows:

1) C1: Executing a policy learned with the operator's fingertip positions $\boldsymbol{x}^i_{o,k}$ as the expert action with the control pipeline excluding relative geometrical constraint.
2) C2: Executing a policy learned with the constrained commands $\boldsymbol{x}^i_{g,k}$ as the expert action with the control pipeline excluding relative geometrical constraint.
3) C3: Executing a policy learned with the constrained commands $\boldsymbol{x}^i_{g,k}$ as the expert action with the entire control pipeline.

To demonstrate the utilization of data collected using the proposed method, we gathered 30 trajectories through online teleoperation by a human operator using a MANUS glove. These trajectories consisted of state-action pairs (detailed in the following section). Using this training dataset, we trained imitation learning policies to replicate the demonstrated behaviors. The trained policy was executed five times for each of the three combinations to evaluate its performance statistically. The termination condition was set to be either the angle between the screwdriver's longitudinal axis and the world z-axis being 30° or less (this threshold was selected since most of the demonstrations in the training dataset achieved final angles under 30°), or the policy's action no longer changing.

1) Imitation Learning Algorithm: For the imitation learning algorithm, we employed simple behavior cloning [21]. In behavior cloning, the objective is to learn a policy $\hat{\pi}^* \in \Pi$ that closely imitates an expert's demonstrated behavior, where the policy space Π is defined as $\Pi := \pi : S \rightarrow A$, with each policy π mapping the state space S to a distribution over the action space A. Given expert demonstrations organized as trajectories $D = \{o^*_j\}^M_{j=1}$, where each trajectory o^*_j comprises a sequence of state-action pairs $o^*_j = (s^*_{j,1}, a^*_{j,1}), (s^*_{j,2}, a^*_{j,2}).....(s^*_{j,H_j}, a^*_{j,H_j})$ over a time horizon H_j, the goal is to approximate the expert's policy by minimizing the discrepancy between predicted and expert actions. Each state $s^*_{j,h}$ consists of the robot's current joint angles, six-axis force-torque sensor data, and the object's position and orientation, obtained via an OptiTrack Motion Capture System by attaching makers on the screwdriver. The corresponding expert action $a^*_{j,h}$ specifies the joint angle commands (converted from the expert fingertip positions using inverse kinematics). To obtain the policy $\hat{\pi}^* \in \Pi$ that best approximates the expert policy π^*, we formulated a supervised learning problem with the objective of minimizing the mean squared error (MSE) loss:

$$\hat{\pi}^* = \arg\min_{\pi \in \Pi} \sum_{j=1}^{M} \frac{1}{H_j} \sum_{h=1}^{H_j} \|\pi(s^*_{j,h}) - a^*_{j,h}\|^2_2 \qquad (8)$$

2) Results: Fig. 9(a) illustrates the fingertip penetration analysis using BoxPlots, showing the average penetration of the thumb and middle finger across all experimental combinations (C1, C2, and C3), computed over the entire task duration. For calculating confidence intervals for the difference in means ($\mu_{C2} - \mu_{C3}$), we used the bootstrap method [22] since it is a nonparametric method that does not assume a specific distribution for the subject of analysis. A pair of resampled datasets was generated by randomly drawing data points with replacement from each of the two original datasets separately, keeping the sample sizes equal

to the originals. The difference in means was then computed for this pair. This procedure was repeated 5000 times to obtain a set of differences in means, from which the 2.5th and 97.5th percentiles were extracted as the bounds of the 95% confidence interval ($CI_{95\%}$). The statistical analysis for penetration was as follows:

- Comparison of C2 and C3: $CI_{95\%}(\mu_{C2} - \mu_{C3}) = [0.40, 0.43]$ mm. This difference is within the error range of the constraint performance in Section III-B.
- Comparison of C1 and C2: $CI_{95\%}(\mu_{C2} - \mu_{C1}) = [-4.1, -4.0]$ mm. This difference is approximately ten times larger than the C2 and C3 comparison.

Regarding task success, C1 failed all five attempts. Fig. 10 shows how C1 failed. The screwdriver was lifted up and not manipulated properly. Due to the excessive penetration, the balance of forces necessary for the manipulation was disrupted. C2 and C3 successfully completed the tasks, with sliding amounts shown in Fig. 9(b). The sliding amount was calculated as the displacement of the midpoint between the thumb and middle finger contact points along the screwdriver's longitudinal axis. The sliding analysis yielded:

- $CI_{95\%}(\mu_{C2} - \mu_{C3}) = [-2.0, 5.3]$ mm. The inclusion of zero in this interval indicates no statistically significant difference was found in sliding behaviors.

From the perspectives of penetration amount and task success, imitation learning using constrained fingertip position $\boldsymbol{x}^i_{g,k}$ as expert action enhanced the task performance. Even without the proposed constraint during policy execution, the performance was not degraded significantly.

IV. CONCLUSION

We propose a method that enables intuitive and stable in-hand manipulation in teleoperation of a multifingered robot hand. This method can comprehensively handle the fundamental in-hand manipulation components by constraining operator finger motion relative to the object motion intended by the operator. Experimental results show the proposed method's high tracking performance of the operator's finger motion and versatility for differently shaped objects. Furthermore, the imitation learning validation was conducted using data collected by the proposed method. The performance of the policy, trained via imitation learning with constrained fingertip positions as expert action, was not degraded when executed even without the proposed constraints. This indicates that the collected dataset remains usable even without implementing the constraint, thereby enhancing the scalability of the dataset. Future work will include the following:

1) Development of a constraint that considers the cases where not only the fingertips but also other parts of the effectors are used.
2) Extension to handle the cases where the finger and object are in soft contact rather than rigid contact.
3) For learning validation, instead of using the constrained fingertip positions as expert action, using the final joint torque commands including the grasping force commands as expert action.

REFERENCES

[1] H. Asada and H. Hanafusa, "Prehension and handling of objects by robot hands with redundant fingers," *Transactions of the Society of Instrument and Control Engineers*, vol. 15, no. 5, pp. 666–671, 1979.

[2] T. Okada, "Computer control of multijointed finger system for precise object-handling," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 12, no. 3, pp. 289–299, 1982.

[3] R. Ozawa and K. Tahara, "Grasp and dexterous manipulation of multi-fingered robotic hands: a review from a control view point," *Advanced Robotics*, vol. 31, no. 19-20, pp. 1030–1050, 2017.

[4] D. Prattichizzo *et al.*, "Dexterous manipulation," in *Encyclopedia of Robotics*. Springer Berlin, Heidelberg, 2020, pp. 1–8.

[5] Y. Li *et al.*, "A survey of multifingered robotic manipulation: Biological results, structural evolvements, and learning methods," *Frontiers in Neurorobotics*, vol. 16, p. 843267, 2022.

[6] T. Hasegawa *et al.*, "Powerful and dexterous multi-finger hand using dynamical pulley mechanism," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 707–713.

[7] R. R. Ma and A. M. Dollar, "On dexterity and dexterous manipulation," in *2011 15th International Conference on Advanced Robotics (ICAR)*. IEEE, 2011, pp. 1–7.

[8] M. Caeiro-Rodríguez *et al.*, "A systematic review of commercial smart gloves: Current status and applications," *Sensors*, vol. 21, no. 8, p. 2667, 2021.

[9] K. Gao *et al.*, "Challenges and solutions for vision-based hand gesture interpretation: A review," *Computer Vision and Image Understanding*, p. 104095, 2024.

[10] R. Ozawa *et al.*, "Supervisory control strategies in a multi-fingered robotic hand system," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 965–970.

[11] R. Ozawa and N. Ueda, "Supervisory control of a multi-fingered robotic hand system with data glove," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 1606–1611.

[12] Y. Yoshimura and R. Ozawa, "A supervisory control system for a multi-fingered robotic hand using datagloves and a haptic device," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5414–5419.

[13] D. Berenson *et al.*, "Manipulation planning on constraint manifolds," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 625–632.

[14] M. T. Mason, "Compliance and force control for computer controlled manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.

[15] T. Li *et al.*, "Learning hierarchical control for robust in-hand manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8855–8862.

[16] F. Veiga *et al.*, "Hierarchical tactile-based control decomposition of dexterous in-hand manipulation tasks," *Frontiers in Robotics and AI*, vol. 7, p. 521448, 2020.

[17] F. Khadivar and A. Billard, "Adaptive fingers coordination for robust grasp and in-hand manipulation under disturbances and unknown dynamics," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3350–3367, 2023.

[18] G. Khandate *et al.*, "Dexterous in-hand manipulation by guiding exploration with simple sub-skill controllers," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6551–6557.

[19] A. Bicchi, "Intrinsic contact sensing for soft fingers," in *Proceedings., IEEE International Conference on Robotics and Automation*. IEEE, 1990, pp. 968–973.

[20] T. Wimboeck *et al.*, "Passivity-based object-level impedance control for a multifingered hand," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 4621–4627.

[21] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[22] A. C. Davison and D. V. Hinkley, *Bootstrap methods and their application*. Cambridge university press, 1997, no. 1.