

On the Role of Jacobians in Robust Manipulation

Joshua T. Grace¹, Podshara Chanrungrameekul², Kaiyu Hang², and Aaron M. Dollar¹

Abstract—Traditional robot control relies on analytical methods that require precise system models, which are hard to apply in real-world settings and limit generalization to arbitrary tasks. However, systems like serial manipulators and passively adaptive hands feature inherently stable regions without control discontinuities like loss of contact or singularities. In these regions, approximate controllers focusing on the correct direction of motion enable successful coarse manipulation. When coupled with a rough estimation of the motion magnitude, precision manipulation is achieved. Leveraging this insight, we introduce a novel inverse Jacobian estimation method that independently estimates the primary motion direction and magnitude of the manipulator’s actuators. Our method efficiently estimates the direct mapping from task to actuator space with no need for *a priori* system knowledge enabling the same framework to control both hands and arms without compromising task performance. We present a novel control method with no *a priori* knowledge for precision manipulation. Experiments on the Yale Model O hand, Yale Stewart Hand, and a UR5e arm demonstrate that the inverse Jacobians estimated via our approach enable real-time control with submillimeter precision in manipulation tasks. These results highlight that online self-ID data alone is sufficient for precise real-world manipulation.

I. INTRODUCTION

Manipulation is vital for a range of real-world tasks, including assembling puzzles, buttoning shirts, and screwing on bottle caps. Achieving these tasks requires a combination of arm-based manipulation for large-scale movements and dexterous in-hand manipulation for precision alignment. Dexterous in-hand manipulation, often called Within Hand Manipulation or WiHM, is defined as the capability to reposition or reorient an object within a grasp using a set of finger contacts regulated by the finger joints [1]. In this work, we consider controlling WiHM and arm-based systems.

WiHM is generally used for precise control tasks where an object is manipulated in a constrained region. Traditional WiHM approaches use fully actuated hands, which may lose their grasp with small motor perturbations so require complex control. Passively adaptive hands leverage mechanical compliance to enable a region of stability in which the grasp is robust over a range of movements [2], [3]. These stable regions enable simple models to effectively

¹Department of Mechanical Engineering & Materials Science, Yale University, New Haven, CT 06520 USA josh.grace@yale.edu; aaron.dollar@yale.edu

²Department of Computer Science, Rice University, Houston, TX 77005, USA. pc45@rice.edu; kaiyu.hang@rice.edu

This work was supported by the US National Science Foundation grants FRR-2132823 and FRR-2133110 and funded in part by the Robotics and AI Institute. JTG is supported by a US NSF Graduate Research Fellowship

Paper code available at: <https://github.com/grablab/IROS2025-RoleOfJacobians>; Video available at <https://youtu.be/rbBaVNZH8Zk>

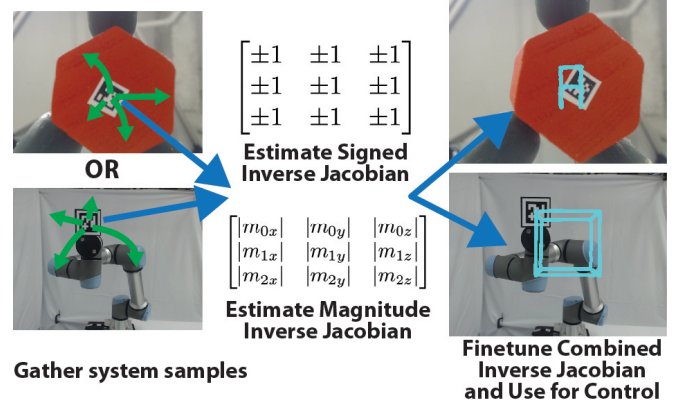


Fig. 1: Inverse Jacobians estimated with no *a priori* knowledge. Data from random motions used to estimate separate signed and magnitude inverse Jacobian. Matrices combined and finetuned to enable precision manipulation

perform precision manipulation tasks [4]. We also consider manipulation with robot arms which enable coarse control over a larger region of task space. Like passively adaptive mechanisms, robot arms with closed-loop controllers moving in free space outside singularities exhibit similar robustness to error [5]. As WiHM presents a greater control problem than serial manipulation, we focus on the problem’s WiHM formulation, but the same principles extend to both cases.

In both WiHM and arm manipulation, *a priori* knowledge requirements can be challenging. *A priori* information can include kinematic models, object characteristics, contact points, and eye-hand calibration. These parameters require human intervention to determine and prevent generalization to arbitrary tasks or environments. So, we aim to eliminate the need for this information in the controller. A more practical approach is to directly estimate these parameters online and instantiate a controller from the estimation.

We perform this parameter estimation using Self-Identification (self-ID) [6], described in Section III-A. Instead of requiring explicit knowledge of a robot’s kinematics, these values are learned via interactions with the environment. It enables a robot to reason about its actions in the real world and update the internal kinematic model by observing the results. Unlike previous self-ID approaches which use models requiring *a priori* knowledge, we estimate the controller fully online. We focus on self-identifying the inverse Jacobian of a manipulator, a direct mapping from task to actuator space.

This work leverages self-ID to estimate 3D precision manipulation models for hands and arms without *a priori* knowledge other than the number of actuators and a safe actuation magnitude. As shown in Fig. 1, we introduce

a novel framework for inverse Jacobians in manipulation, splitting them into a signed inverse Jacobian, which focuses on the primary direction of motion, and the magnitude inverse Jacobian, which focuses on the amount of motion. This enables a simpler estimation framework and eliminates the reliance on prior information.

II. RELATED WORK

A. Approximate Jacobians for Control

Jacobian-based control is a popular method for general manipulation. Prior work has shown stable Jacobian-based control for robot arms with kinematic uncertainty [5]. In continuum robot control, Jacobians have been estimated with Kalman filters [7], or using optimization to account for external obstacles [8]. Similarly, Jacobian estimation has been applied to soft objects to move them to a desired shape [9]. However, none of these methods extend to WiHM.

Applying Jacobian-based control to robot hands introduces challenges due to the need for a stable grasp. An adaptive Jacobian controller has been derived which handles uncertain contact location, hand kinematics, and object morphology [10]. Similarly, a Jacobian estimation process using neural networks has been proposed under the same constraints [11]. However, both methods are demonstrated only in simulation.

Recent work estimates the Jacobian of a soft compliant hand to repeat operator demonstrated object movements [12]. But, this approach only reproduces observed motion and does not perform generalizable precision manipulation. Our previous work demonstrated precision WiHM using inverse Jacobians estimated via a particle filter for a passively adaptive underactuated hand [13]. But, that method requires an *a priori* distribution of feasible inverse Jacobians, which limits the control frame and generalization to other hands. In contrast, this work achieves full 3D precision manipulation only using self-ID data without *a priori* knowledge.

B. Learning-Based Manipulation

Learning-based approaches have become a popular method of controlling robot hands. Initial work demonstrated learned dexterous manipulation in simulation environments [14]. Domain randomization allowed simulation training to generalize to the real world for cube rotation [15], and enhanced domain randomization enabled solving Rubik’s cube in hand [16]. Recent works have shown learned models performing object rotation, including z-axis rotation with depth camera images for a wide range of objects [17], arbitrary axis rotation using touch and proprioceptive information [18], and extreme aspect ratio objects rotation [19]. Precision WiHM on passively adaptive underactuated hands has also been shown with learned state transition models [20].

While these methods do not require explicit knowledge of the underlying system kinematics or dynamics, they are implicitly embedded in the model. System priors are encoded in the large parameter spaces of learned model which require significant data and are prone to overfitting. So, these methods struggle with generalization and often have slow control frequencies. Unlike learning-based methods, we

estimate a control with no *a priori* knowledge using simple computational techniques enabling efficient, precise control.

C. Control Methods for Passively Adaptive Hands

The grasp stability of passively adaptive hands allows simple control methods to be highly capable. 2D WiHM has been demonstrated using predefined hand motion sets [21], Gaussian Process Models trained on real-world data [20], and Gaussian Process Regression from self-ID data [22]. While sufficient for planar hands performing 2D manipulation, these methods are not demonstrated on 3D tasks which are more difficult given the additional degrees of freedom and control requirements for non-planar hands.

3D WiHM has been shown using a Random Forest regressor predicting required actuation velocity with MPC [23], using a particle filter to estimate the kinematic parameters of the hand [6], [24], and a particle filter predicting the inverse Jacobian [13]. These methods require *a priori* knowledge of the hand, reducing their generality. In contrast to these approaches, this method performs precision 3D control without *a priori* system knowledge. Leveraging online self-ID, the region of valid inverse Jacobians, and our novel estimation method, we show precise WiHM and generalization to a range of systems without modifying the estimator.

III. PRELIMINARIES AND PROBLEM FORMULATION

A. Self-Identification

Self-Identification (self-ID) is a process in which robots gather information about the world through their actions and update their internal model to improve task performance over time. The process assumes that the manipulator has accurate motor position tracking and a system to track end-effector pose in the desired task space. At each self-ID iteration, a maximum motor movement m_a which maintains the grasp or system safety is used to randomly sample an actuation $\delta\theta_{i_t}$ for the n actuators on the manipulator according to $\delta\theta_{i_t} = \text{uniform}(-m_a, m_a)$. The state of the system before actuation s_{t-1} and after actuation s_t is sampled and used to calculate δs_t . The model parameters are updated to reflect the new actuation-effect sample $(\delta\theta_{i_t}, \delta s_t)$, refining the system estimate. Initially, self-ID runs until the model converges to a useful controller. The system then executes the desired task while continuing estimation refinement.

B. Inverse Jacobians

Using self-ID, we aim to estimate the manipulation system’s inverse Jacobian. In particular, we find the inverse Jacobian mapping the twist of some point on the end effector to the manipulator’s actuators. Many manipulation tasks focus on controlling a specific point on an object or manipulator, like the tip of a screwdriver. We define this point as the Point of Manipulation (POM), χ , which combined with the motor positions defines the system state. In WiHM, χ is a point on the object, kinematically linked to the manipulator through the joints, links, and fingertip contacts. For arm-based manipulation, χ is a point on the end-effector.

The Jacobian $J(\theta)$ of a system maps joint velocities $\dot{\theta}$ to the spatial twist V of the end effector in task space as in Equation 1 [25]. The Jacobian controlling V in frame t with transformation T from the Jacobian frame origin is denoted $J_t(\theta)$. It is transformed from $J(\theta)$ using the adjoint $[A_{dT}]$ in Equation 2 allowing control in arbitrary frames. The twist of χ in the POM frame X can then be found using Equations 3, 4. We aim to identify the inverse Jacobian $J_\chi^{-1}(\theta)$ mapping from the desired POM twist to the required joint velocities as in Equation 5. In practice, the inverse Jacobian may be non-square and is approximated via the pseudoinverse $J_\chi^+(\theta)$. Once $J_\chi^+(\theta)$ is found, it can directly control the system.

$$V = J(\theta)\dot{\theta} \quad (1) \quad J_t(\theta) = [A_{dT}]J(\theta) \quad (2)$$

$$J_\chi(\theta) = [A_{dX}]J(\theta) \quad (3) \quad V_\chi = J_\chi(\theta)\dot{\theta} \quad (4)$$

$$\dot{\theta} = J_\chi^{-1}(\theta)V_\chi \quad (5)$$

For serial manipulators, $J^+(\theta)$ is found by computing $J(\theta)$ from forward kinematics and inverting the matrix. For parallel mechanisms like robot hands, $J^{-1}(\theta)$ is computed directly from the inverse kinematics. We define the $J_\chi^+(\theta)$ using x , y , z position, and the exponential form of rotations. To find $J_\chi^+(\theta)$, the transformation matrix of the reference frame is combined with the grasp and object information to compute the adjoint transform. However, this process assumes that both the grasped object and manipulator kinematics are fully known and are constant during manipulation, assumptions that rarely apply in the real world. Relying on *a priori* object or kinematic models limits manipulation to known objects reducing generality. Instead, we estimate the inverse Jacobian directly using online data gathered from self-ID. This enables generalization to a range of objects and manipulators.

C. Valid Inverse Jacobian Space

An important consideration when estimating the inverse Jacobian for manipulation is determining the required estimate accuracy for successful manipulation. Thus, we analyze how sensitive the inverse Jacobian is to error by evaluating how much the true inverse Jacobian elements can be perturbed while maintaining task performance.

Since the range of each element can vary significantly, we introduce perturbations by scaling the values by some factor ϵ . We evaluate robustness with the path completion proportion and the mean tracking error under different scalings. For a given maximum error scaling ϵ_{limit} , a scaling matrix m_ϵ is generated at the start of each test, where each element is sampled independently with $m_{\epsilon_{i,j}} \sim \text{uniform}(1, \epsilon_{limit})$. At each step, the inverse Jacobian of the system is scaled element-wise with $J_\epsilon^+(\theta) = m_\epsilon \circ J^+(\theta)$. Analyzing the path following performance with $J_\epsilon^+(\theta)$ shows the robustness of inverse Jacobian estimation errors, dictating how accurate the estimate must be for successful control.

We first analyze a two-link arm, with link lengths of 1, following a star path with the end effector. We sample 10 representative arm configurations away from singularities, and perform 10 tests per configuration, resampling m_ϵ for each test to gain representative statistics. We define the

Manipulator Valid Inverse Jacobian Region

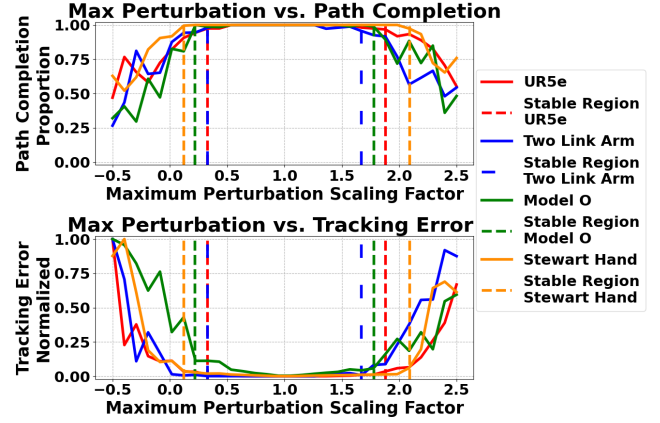


Fig. 2: Valid regions where incorrect inverse Jacobians maintain control. Valid regions (marked by matching vertical lines) have high path completion and low tracking error

valid region as the contiguous region where the mean path completion percent exceeds 95%. So, any inverse Jacobian in this region likely enables successful control. As seen in Fig. 2, the inverse Jacobian of the arm remains valid between 0.33x and 1.67x the true inverse Jacobian element values.

We repeat this evaluation in kinematic simulations of the Yale Model O hand [3], Yale Stewart Hand [26], and a UR5e arm. As shown in Fig. 2, the valid inverse Jacobian range is 0.33x to 1.88x for the UR5e, 0.22x to 1.78x for the Model O, and 0.12x and 2.09x for the Stewart Hand. Overall, we find a consistent region of valid inverse Jacobians across the studied systems, showing that control remains robust despite numerical variation. Therefore, the control problem is robust to model error and self-ID estimates only need to fall within this range for successful control.

D. Signed Inverse Jacobian

We introduce the signed inverse Jacobian, a matrix which describes each actuator's primary direction of motion in task space. In our prior work [13], we estimated the inverse Jacobian using a prior distribution computed in the frame of the in-palm camera. However, when controlling the system in other reference frames, the estimator consistently diverged leading to control failure. We found that significant rotations of the control frame relative to the inverse Jacobian distribution prevents successful estimation. We observed that while the absolute values of the parameters were often in distribution, the sign flips that occur during larger rotations caused the values to fall outside the prior distribution of the particle filter. Intuitively, when a parameter sign is wrong, the actuator moves the opposite direction of the intended motion, compounding over time and destabilizing the system. Additionally, when the sign of the estimated and true inverse Jacobian diverge for a given element, the scaling is negative and thus outside the stable distributions in Section III-C.

To address this challenge, we propose separating the inverse Jacobian estimation problem into two stages. First, we estimate the element signs of the inverse Jacobian, followed

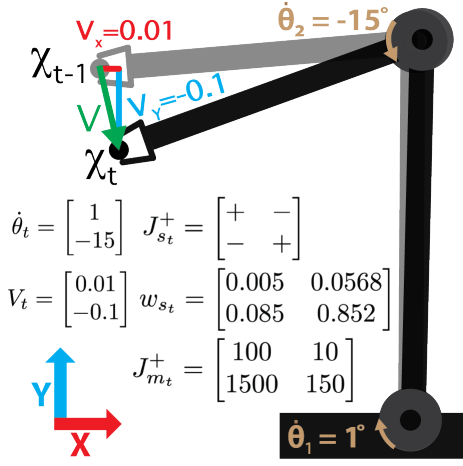


Fig. 3: One step of Sign and Magnitude Inverse Jacobian Estimation; 1. Sample actuation $\dot{\theta}$ and pose change V ; 2. Compute sign matrix J_{st}^+ from $\dot{\theta}_i$ and V_j signs and sign weights w_{s_i} from $\dot{\theta}_i$ and V_j magnitude; 3. Compute magnitude matrix from $\dot{\theta}_i$ and V_j ratio

by a separate estimation of the element magnitudes. We define the matrix of element signs as the signed inverse Jacobian, and the matrix of magnitudes the magnitude inverse Jacobian. The signed inverse Jacobian essentially captures the primary direction that each actuator moves to achieve a desired change in pose along a given axis, while the magnitude inverse Jacobian determines how far an actuator needs to move to achieve a desired magnitude of motion. This separation allows the estimator to treat sign estimation as a discrete classification problem, enabling more robust and precise estimation, while allowing coarser approximation of the magnitudes. We find that this estimation method significantly improves sample efficiency and enables precision manipulation. We introduce a weight-based classifier to find the signed inverse Jacobian, and compute the magnitude inverse Jacobian from the ratio between actuation and movement.

IV. ESTIMATION OF MANIPULATION MODELS

Inverse Jacobian-based control is a powerful method for WiHM, but its real-world use is limited by dependence on accurate system models. We propose a method for estimating the inverse Jacobian without *a priori* knowledge, enabling manipulation across unfamiliar objects and different manipulators. Moving actuators independently to find the partial derivatives to form the inverse Jacobian is a straightforward approach, but does not generalize to movements of multiple actuators, preventing updates online. Inverse Jacobian estimation with arbitrary movements is difficult because multiple actuators contribute to motion, so the influence of individual actuators is ambiguous. Direct optimization of the inverse Jacobian from self-ID data is unreliable for some systems as the non-convexity of the inverse Jacobian space causes the estimate to get stuck at invalid local minima. So, a better method is needed to estimate system inverse Jacobians from any combination of actuations.

To estimate the inverse Jacobian, we decompose the problem into two parts: the direction each actuator needs to move

for some V_χ , and the corresponding movement magnitude, as shown in Fig 3. The signed inverse Jacobian J_s^+ represents the primary motion direction of each actuator, with element values ± 1 estimated via a weight-based approach. The magnitude inverse Jacobian is found with the mean ratio of actuator displacement and resulting V_χ capturing the rough scale of the joint to task space relation. Both matrices are estimated from the same self-ID data, yielding a structured interpretation of the random self-ID data. These matrices are combined elementwise to generate an initial hypothesis of the inverse Jacobian. This estimate is then refined through optimization to estimate $J_\chi^+(\theta)$ as shown in Algorithm 1.

A. Signed Inverse Jacobian

The signed inverse Jacobian J_s^+ determines the primary direction of motion of each actuator on each component of V_χ . During manipulation, interactions between actuators and sensing noise complicate determining the exact relation between actuator inputs and task space movement. To address this complexity, we adopt a weight-based estimation system that accounts for these challenges.

For each actuation-observation pair, we compute the sign of the matrix element s_{ij} with Equation 6. We then compute the weight of each actuator's contribution to the overall motion with Equation 7, assigning higher weights to actuators with larger relative movement. Similarly, Equation 8 determines the relative weight of the resulting pose change to account for different magnitudes of motion across axes. At each self-ID iteration, the sign weights for $+1$, s_{ij+} and -1 , s_{ij-} are updated according to Equation 9. The final sign of each matrix element is given by the highest weighted sign as detailed in Equation 10.

$$s_{ij} = \text{sign}(\dot{\theta}_i) * \text{sign}(V_i) \quad (6) \quad w_{\dot{\theta}_i} = \frac{|\dot{\theta}_i|}{\sum |\dot{\theta}|} \quad (7)$$

$$w_{V_i} = \frac{|V_j|}{\sum |V|} \quad (8) \quad w_{s_{ij_t}} = w_{s_{ij_{t-1}}} + w_{\dot{\theta}_i} \cdot w_{V_i} \quad (9)$$

$$s_{ij} = \max(s_{ij+}, s_{ij-}) \quad (10)$$

A single step of the signed inverse Jacobian estimation process is illustrated in Fig. 3. While V_x and $\dot{\theta}_1$ have smaller values, V_z and $\dot{\theta}_2$ are larger. So, the weight of the $+1$ relation between actuator 2 and the Y axis, $w_{s_{2y}}$ should reflect the large input and corresponding output. Conversely, the weight of $+1$ for actuator 1 and the X axis should remain relatively small. As more self-ID samples are gathered, the weights will be refined until convergence is reached.

B. Magnitude Inverse Jacobian

The magnitude inverse Jacobian J_m^+ approximates how far an actuator needs to move for a given V_χ displacement. It is computed as the mean magnitude of all movements. The magnitude of a given element at time t is given by Equation 11. The final magnitude estimate is the mean of all samples for a given actuator, pose element pair, as shown in Equation 12. As shown in Fig. 3, although the magnitude estimation is quite coarse, the final values are consistently in the basin

of attraction of a valid inverse Jacobian, so the optimization is able to finetune the magnitudes to a valid estimate.

$$m_{ij_t} = \left| \frac{\dot{\theta}_t}{V_{j_t}} \right| \quad (11) \quad m_{ij} = \text{mean}(m_{ij_1}, \dots, m_{ij_t}) \quad (12)$$

C. Final Optimization

After computing the signed and magnitude inverse Jacobians, we combine them elementwise to form the initial inverse Jacobian as in Equation 13. While J_i^+ is near the valid inverse Jacobian space, inaccuracies in the magnitude estimation can introduce scaling errors causing control instability. Thus, we use J_i^+ to initialize an optimization process refining the estimate. The optimization objective is to find the inverse Jacobian that best maps from observed changes in V back to the actuator input $\dot{\theta}$, as shown in Equation 14. In experiments, $\dot{\theta}$ and V are drawn from self-id movements and the optimization uses SLSQP. As demonstrated in our experiments, this optimized inverse Jacobian consistently enables stable and precise control.

Algorithm 1 Inverse Jacobian Estimation Method

```

1:  $J_{f_0}^+ = 0$ 
2:  $V = []$ 
3:  $\dot{\theta} = []$ 
4: while !CONVERGED( $J_{f_{1-2,\dots,t}}^+$ ) do                                ▷ Sec. IV-D
5:    $\dot{\theta}_t \leftarrow \text{MAKERANDOMMOVEMENT}$ 
6:    $\chi_{t-1} \leftarrow \text{SAMPLEPOMPOSE}(\cdot)$ 
7:    $\text{MOVEACTUATORS}(\dot{\theta})$ 
8:    $\chi_t \leftarrow \text{SAMPLEPOMPOSE}(\cdot)$ 
9:    $V_t \leftarrow \text{DIFFERENCE}(\chi_t, \chi_{t-1})$ 
10:   $J_{s_t}^+ \leftarrow \text{SIGNEDINVERSEJACOBIAN}(\dot{\theta}, V)$                                 ▷ Sec. IV-A
11:   $J_{m_t}^+ \leftarrow \text{MAGNITUDEINVERSEJACOBIAN}(\dot{\theta}, V)$                     ▷ Sec. IV-B
12:   $J_{i_t}^+ \leftarrow J_{s_t}^+ \circ J_{m_t}^+$                                           ▷ Sec. IV-C
13:   $J_{f_t}^+ = \underset{J_i^+}{\text{argmin}} \|\dot{\theta} - J_i^+ V\|_2$ 
14: end while
15: return  $J_f^+$ 

```

$$J_i^+ = J_s^+ \circ J_m^+ \quad (13) \quad J_f^+ = \underset{J_i^+}{\text{argmin}} \|\dot{\theta} - J_i^+ V\|_2 \quad (14)$$

D. Convergence Criteria

An important consideration in self-ID is determining when sufficient data has been gathered to begin manipulation. Since self-ID sampling delays manipulation, it is crucial to select convergence criteria that minimize the identification time while ensuring robust estimation. We define convergence as the L2 norm of the difference between subsequent estimates dropping below some ϵ for two iterations: $\|J_{t-1}^+ - J_t^+\|_2 \leq \epsilon$. At this point, the estimate stabilizes, additional data has minimal impact, and it can be used for control.

V. RESULTS

We perform comprehensive real-world experiments to evaluate our inverse Jacobian estimation system, focusing on two key questions: 1. How does the removal of *a priori* knowledge impact task performance; 2. Can our estimator generalize across manipulators. We perform handwriting

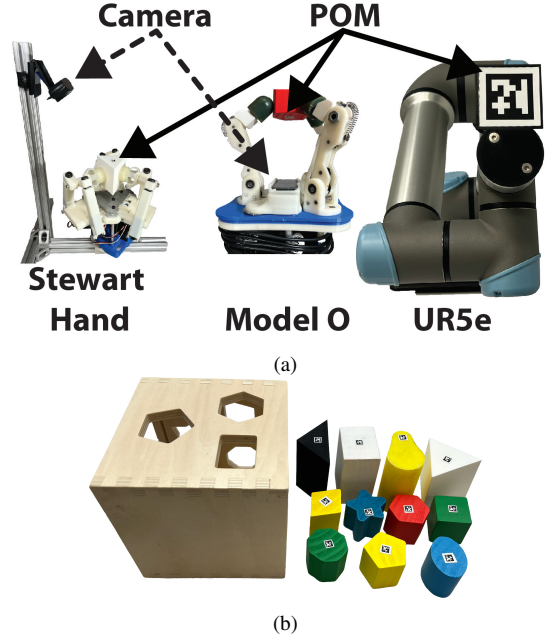


Fig. 4: Experimental Setup; 4a: Manipulators for experiments. POMs marked with AprilTag [27] controlled in camera frame; 4b: Objects for WiHM experiments: From left to right, front to back: Peg-in-Hole Box, Green Octagon, Yellow Pentagon, Blue Oval, Yellow Parallelogram, Blue Star, Red Hexagon, Green Cube, Black Triangle, White Cuboid, Yellow Pear, and White Triangle

experiments where the estimated inverse Jacobian traces a path with the POM of an object to assess task performance. These tests provide qualitative and quantitative evaluations of control performance and are standard benchmark for manipulation approaches with varying amounts of *a priori* system information [28]. To evaluate manipulator generalization, we evaluate our approach on WiHM with two passively adaptive hands, the Yale Model O hand [3], a Yale Stewart hand [26], and on serial manipulators using a UR5e arm.

Our experiment setup is shown in Fig. 4a. The POM of each system is marked with an AprilTag and tracked using an external camera. Throughout all experiments, the estimator has no *a priori* knowledge of the system other than the stable movement magnitude ϵ_{limit} and number of actuators n . The Yale Stewart Hand uses a single motor to pull three base joints into the object to stabilize the grasp, and six linear actuators controlled by the inverse Jacobian to manipulate the grasped object. The Yale Model O hand is underactuated, with each actuator controlling two finger joints without finger joint sensing. So, the estimated inverse Jacobian must directly map object pose changes to the motor movements. The UR5e arm is fully actuated with the inverse Jacobian controlling all six actuators. For WiHM experiments, the POM is a point on the grasped object manipulated using the hand's fingers while the POM on the UR5e is attached to the end-effector. In all experiments, the system starts by making random self-ID movements until the estimator converges, at which point the inverse Jacobian is ready for control. The range of systems with varied controllability, degrees of freedom, and motor types shows that we estimate useful

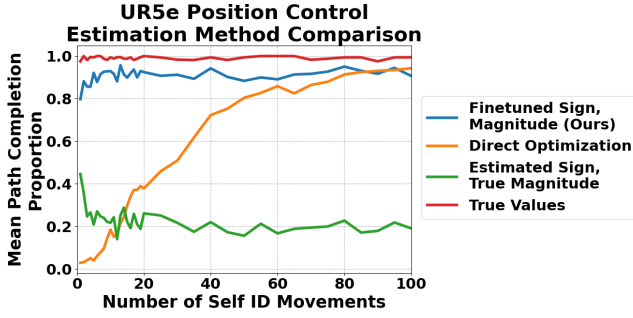


Fig. 5: Comparison of inverse Jacobian estimation methods

inverse Jacobians across manipulators.

The objects used for WiHM experiments are shown in Fig. 4b. We use a variety of objects shapes and sizes for testing, the Black Triangle, White Cuboid, Yellow Pear, and White Triangle measure approximately 50mm x 50mm x 75mm. The remaining objects are approximately 40mm x 40mm x 40mm. The range of object shapes and sizes demonstrates that our system generalizes to a variety of objects.

A. Estimation Method Ablation Study

We evaluate our inverse Jacobian estimation method against three baselines on a 3D positional path tracking task with a simulated UR5e. First, we directly optimize the inverse Jacobian from self-ID data. Second, we use the estimated signed inverse Jacobian and true element magnitudes. Finally, we compare control with the true inverse Jacobian. We run self-ID for the selected number of iterations across 9 UR5e configurations, repeating each trial 10 times while tracking a star-shaped path to find the mean path completion.

Our results in Fig. 5 show that finetuning the combined sign-magnitude inverse Jacobian outperforms the other baselines, particularly in the low-data regime. Our method quickly finds a useful inverse Jacobian as the sign-magnitude matrix is close to valid region. In contrast, directly optimizing an inverse Jacobian requires much more data to find a useful estimation, at ~ 80 iterations our method performs similar to direct optimization. We also find that the estimated sign matrix inconsistently achieves control depending on the arm configuration. A more capable sign estimation framework would likely improve this performance. Overall, these ablations validate the performance of the finetuned sign-magnitude matrix.

B. Handwriting Control Tests

We analyze the precision manipulation capabilities of our method on handwriting tasks using a Yale Model O hand manipulating a variety of small and large objects. Each path consists of position waypoints at the letter corners, which the controller tracks with incremental steps. The visual results in Fig. 6 show that the estimated inverse Jacobian performs precise control of the object. The absolute error at each control step is shown in Table I, confirming that our approach maintains submillimeter control across all tasks.

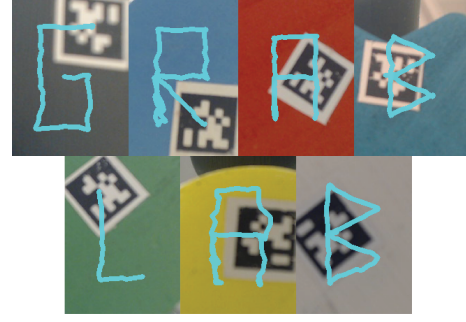


Fig. 6: Model O writing “GRABLAB” using estimated inverse Jacobian to manipulate Black Triangle, Blue Oval, Red Hexagon, Blue Star, Green Square, Yellow Cylinder, and White Cuboid. Paths shown in blue

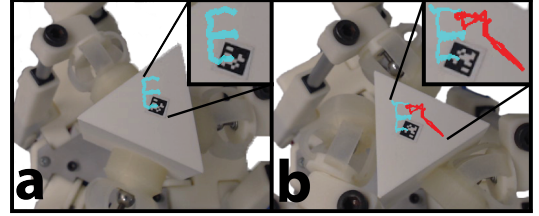


Fig. 7: Stewart Hand estimated inverse Jacobian writing “E” with White Triangle; 7.a: Nominal path completion; 7.b: Path completion despite finger slip (in red), note that finger pads move to side grasp

To compare against previous WiHM controllers, we analyze the waypoint tracking error. The evaluations are comparable as the letter heights are all 12mm, and use the Model O hand. Our method achieves a waypoint error of $0.41 \pm 0.16\text{mm}$ writing “GRABLAB”, converging in 5.7 ± 1.0 self-ID iterations. The VLR particle filter [6] writing “SCIENCE” had waypoint error $0.42 \pm 0.34\text{mm}$ converging in 15.1 ± 3.6 iterations. The inverse Jacobian particle filter [13] writing “ICRA” had $0.45 \pm 0.24\text{mm}$ of error and converged in 5.4 ± 2.8 iterations. Despite removing *a priori* system information, our approach achieves similar or better performance in tracking error and estimation time. This demonstrates that despite removing prior knowledge, we retain effective control.

C. Stewart Hand Control Tests

We also analyze handwriting on the Yale Stewart Hand. As shown in Fig. 7.a and Table I, the estimator successfully estimates a valid inverse Jacobian for the hand-object system,

TABLE I: Experimental Absolute Position Error (mm)

Experiment	Mean X Error	Mean Y Error	Mean Z Error
Model O G	0.37 ± 0.21	0.19 ± 0.13	0.38 ± 0.12
Model O R	0.17 ± 0.17	0.17 ± 0.10	0.28 ± 0.14
Model O A	0.24 ± 0.14	0.11 ± 0.08	0.29 ± 0.13
Model O B	0.11 ± 0.09	0.16 ± 0.11	0.21 ± 0.11
Model O L	0.27 ± 0.16	0.18 ± 0.11	0.29 ± 0.10
Model O A	0.31 ± 0.26	0.24 ± 0.19	0.29 ± 0.16
Model O B	0.20 ± 0.18	0.19 ± 0.12	0.30 ± 0.14
Stewart Hand E	0.27 ± 0.27	0.25 ± 0.23	0.18 ± 0.15
UR5e Cube Estimated	1.41 ± 1.31	1.18 ± 1.03	1.73 ± 1.39
UR5e Cube Analytical	0.35 ± 0.29	0.28 ± 0.22	0.93 ± 0.85

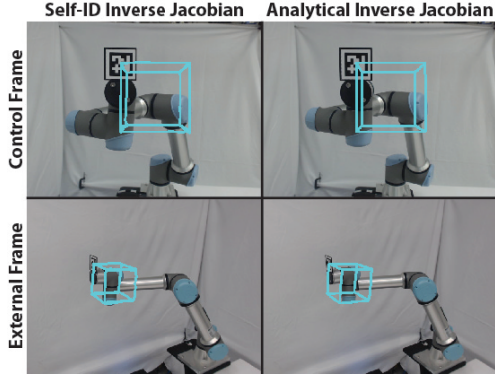


Fig. 8: Estimated inverse Jacobian vs. Analytical inverse Jacobian 6DOF pose regulation for Cube Trajectory. POM position controlled via waypoints, POM rotation kept constant

TABLE II: Self-ID Rotation Error (deg)

Experiment	Final Error	Control Step Error
Green Octagon X Axis	0.37	0.56 ± 0.39
Green Octagon Y Axis	0.74	0.57 ± 0.30
Yellow Parallelogram Y Axis	0.42	0.58 ± 0.26
UR5e Cube Estimated	3.58	4.07 ± 1.95
UR5e Cube Analytical	2.28	2.82 ± 1.74

achieving a waypoint error of $0.28 \pm 0.22\text{mm}$ converging in 10 iterations. This shows that our estimator enables precise WiHM across different manipulators without modification.

During experiments with the Stewart Hand, we observed an unexpected robustness to finger slip. Initially, the hand held the object similar to Fig. 7.a, but after self-ID, the fingers lost their grasp and began sliding along the object’s surface. As shown in Fig. 7.b, the grasp reconfigured from the fingerpad faces to their sides. Despite the reconfiguration of the mechanism and object, once the grasp re-stabilized, the estimator completed the path successfully. This highlights that if the inverse Jacobian estimate remains within the hand’s valid inverse Jacobian region, task performance can be maintained under significant perturbation.

D. Rotation Tests

We evaluate rotation control using the estimated inverse Jacobian. We test X and Y axis rotation with the Green Octagon, and Y axis rotation with the Yellow parallelogram using the Model O. Each experiment rotates the object 0.2 radians (11.5 deg) along the selected axis. We measure the angular difference between the goal and actual POM rotation at each control step and at the final rotation. As detailed in Table II, rotation error remains consistently below one degree, showing the precision of the inverse Jacobian estimate. The method is also data-efficient, with convergence in 5.0 ± 0.0 iterations. Like the position error results, we observe low error rates across all experiments, demonstrating that despite eliminating *a priori* knowledge from the system, there is no compromise in task performance, further showing the precision and robustness of our approach.

E. UR5e Pose Control

As the UR5e has a known analytical inverse Jacobian, we compare 6-DOF control with an estimated inverse Jacobian

TABLE III: Peg-in-hole Errors

Experiment	UR5e Pos (mm)	UR5e Rot (deg)	Model O Rot (deg)
Blue Oval	1.28 ± 0.66	0.41 ± 0.20	0.40
Blue Star	1.45 ± 0.71	0.20 ± 0.16	0.39
Green Octagon	1.54 ± 0.70	0.30 ± 0.22	1.28
Red Hexagon	1.33 ± 0.72	0.26 ± 0.16	0.80
Yellow Pentagon	0.97 ± 0.61	0.10 ± 0.07	0.41

to the analytical solution. We test tracing a $150 \times 150 \times 150\text{mm}$ Cube with the arm’s POM while maintaining a constant POM orientation. The trajectories are shown in Fig. 8 and the quantitative error metrics shown in Tables I and II. The estimator converged in 8 iterations. While the estimated inverse Jacobian has a minor increase in error versus the analytical solution, the absolute error is still low allowing usability in the real-world. These results further validate that our estimator enables precision manipulation.

F. Peg-in-hole Insertion Tests

To evaluate the real-world applicability of our approach, we perform position-based peg-in-hole insertion tasks, a tight tolerance test demonstrating high precision and accuracy control. The experiments use a Model O hand mounted on the UR5e. The arm follows a predefined series of 6DOF pose waypoints, maintaining constant X and Y rotation of the manipulator. First, the inverse Jacobian of the arm is estimated using self-ID and moves to the grasping waypoint, grabbing the peg. The arm then moves to slightly above the desired hole. Next, the hand runs self-ID on the grasped object and rotates it to the perpendicular orientation. This step is needed as the initial grasp induces an object rotation, preventing insertion in the initial configuration. The arm then moves the object to the hole and releases it.

We perform peg-in-hole insertions for five objects with objects and holes in different locations. An example experiment using the Green Octagon is shown in Fig. 9. The quantitative UR5e waypoint errors and final WiHM rotation errors are detailed in Table III. The UR5e estimator converged in 8.2 ± 0.4 iterations, while the Model O converged in 5.6 ± 1.0 iterations. Compared to the 6DOF UR5e control experiments, we observe a reduction in rotation error. This improvement occurs because the arm operates in a smaller range with higher manipulability, allowing the inverse Jacobian to maintain finer control capability. These results validate that our estimated inverse Jacobian is sufficiently accurate to enable real-world manipulation tasks with tight tolerances.

VI. CONCLUSION

We introduce the signed and magnitude inverse Jacobians as an effective framework for interpreting self-ID data. By separately estimating the principal motion directions and movement magnitudes, we efficiently find useful inverse Jacobians online. Using this approach, we present a novel method for 3D precision Within Hand Manipulation and arm control without *a priori* knowledge, generalizing across a range of manipulators without manual intervention.

Although our results confirm task success within the valid inverse Jacobian space, further work is needed to better

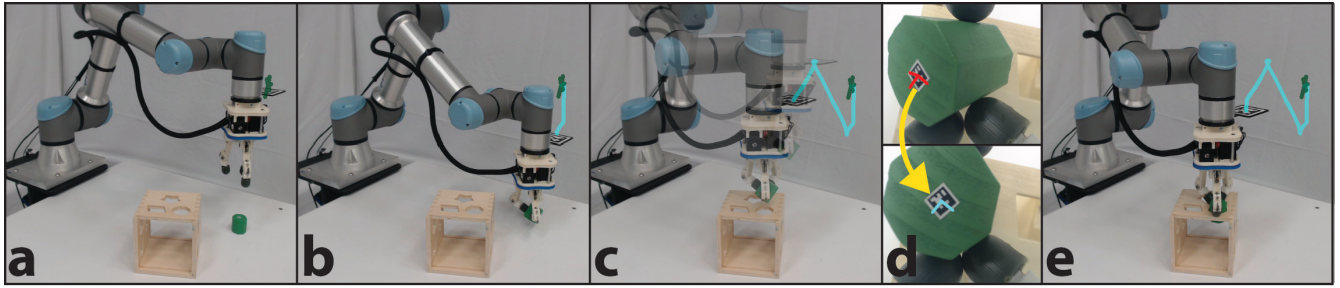


Fig. 9: Peg-in-hole insertion using estimated inverse Jacobians; a: Estimate arm inverse Jacobian (self-ID movements in green); b: Navigate to grasp waypoint, grab object; c: Go to insertion waypoint; d: Estimate hand inverse Jacobian, rotate object to vertical; e: Insert peg

define its boundaries, as some inverse Jacobians outside the valid region still enable control. Additionally, we plan to refine the magnitude estimation with a Kalman Filter or similar estimator to eliminate the need for fine-tuning the final estimate. Finally, while our method uses no *a priori* knowledge of the system or object, we rely on AprilTags for pose tracking. We plan to investigate noisier learning-based pose estimators to determine if they can enable precise manipulation without object modification. Additionally, object tracking could be achieved with touch sensing, eliminating the requirement for external cameras.

REFERENCES

- [1] A. Bicchi, “Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 652–662, 2000.
- [2] A. M. Dollar and R. D. Howe, “The highly adaptive sdm hand: Design and performance evaluation,” *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 585–597, 2010.
- [3] L. U. Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, and A. M. Dollar, “A compliant, underactuated hand for robust manipulation,” *The International Journal of Robotics Research*, vol. 33, no. 5, pp. 736–752, 2014.
- [4] L. U. Odhner, R. R. Ma, and A. M. Dollar, “Open-loop precision grasping with underactuated hands inspired by a human manipulation strategy,” *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 625–633, 2013.
- [5] C.-C. Cheah, M. Hirano, S. Kawamura, and S. Arimoto, “Approximate jacobian control for robots with uncertain kinematics and dynamics,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 692–702, 2003.
- [6] K. Hang, W. G. Bircher, A. S. Morgan, and A. M. Dollar, “Manipulation for self-identification, and self-identification for better manipulation,” *Science Robotics*, vol. 6, no. 54, p. eabe1321, 2021.
- [7] M. Li, R. Kang, D. T. Branson, and J. S. Dai, “Model-free control for continuum robots based on an adaptive kalman filter,” *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 286–297, 2017.
- [8] M. C. Yip and D. B. Camarillo, “Model-less feedback control of continuum manipulators in constrained environments,” *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 880–889, 2014.
- [9] D. Navarro-Alarcon, Y.-H. Liu, J. G. Romero, and P. Li, “Model-free visually servoed deformation control of elastic objects by robot manipulators,” *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1457–1468, 2013.
- [10] C.-C. Cheah, H.-Y. Han, S. Kawamura, and S. Arimoto, “Grasping and position control for multi-fingered robot hands with uncertain jacobian matrices,” in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3. IEEE, 1998, pp. 2403–2408.
- [11] Y. Zhao and C. C. Cheah, “Neural network control of multifingered robot hands using visual feedback,” *IEEE Transactions on Neural Networks*, vol. 20, no. 5, pp. 758–767, 2009.
- [12] A. Sieler and O. Brock, “Dexterous soft hands linearize feedback-control for in-hand manipulation,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8757–8764.
- [13] J. T. Grace, P. Chanrungrmaneeekul, K. Hang, and A. M. Dollar, “Direct self-identification of inverse jacobians for dexterous manipulation through particle filtering,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 862–13 868.
- [14] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, “Learning complex dexterous manipulation with deep reinforcement learning and demonstrations,” *arXiv preprint arXiv:1709.10087*, 2017.
- [15] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, *et al.*, “Learning dexterous in-hand manipulation,” *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.
- [16] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, *et al.*, “Solving rubik’s cube with a robot hand,” *arXiv preprint arXiv:1910.07113*, 2019.
- [17] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, “Visual dexterity: In-hand reorientation of novel and complex object shapes,” *Science Robotics*, vol. 8, no. 84, p. eadc9244, 2023.
- [18] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, “General in-hand object rotation with vision and touch,” in *Conference on Robot Learning*. PMLR, 2023, pp. 2549–2564.
- [19] J. Wang, Y. Yuan, H. Che, H. Qi, Y. Ma, J. Malik, and X. Wang, “Lessons from learning to spin” pens,” *arXiv preprint arXiv:2407.18902*, 2024.
- [20] A. Sintov, A. S. Morgan, A. Kimmel, A. M. Dollar, K. E. Bekris, and A. Boularias, “Learning a state transition model of an underactuated adaptive hand,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1287–1294, 2019.
- [21] B. Calli and A. M. Dollar, “Vision-based precision manipulation with underactuated hands: Simple and effective solutions for dexterity,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1012–1018.
- [22] P. Chanrungrmaneeekul, K. Ren, J. T. Grace, A. M. Dollar, and K. Hang, “Non-parametric self-identification and model predictive control of dexterous in-hand manipulation,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8743–8750.
- [23] A. S. Morgan, K. Hang, and A. M. Dollar, “Object-agnostic dexterous manipulation of partially constrained trajectories,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5494–5501, 2020.
- [24] K. Hang, W. G. Bircher, A. S. Morgan, and A. M. Dollar, “Hand-object configuration estimation using particle filters for dexterous in-hand manipulation,” *The International Journal of Robotics Research*, vol. 39, no. 14, pp. 1760–1774, 2020.
- [25] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.
- [26] C. McCann, V. Patel, and A. Dollar, “The stewart hand: A highly dexterous, six-degrees-of-freedom manipulator based on the stewart-gough platform,” *IEEE Robotics & Automation Magazine*, vol. 28, no. 2, pp. 23–36, 2021.
- [27] E. Olson, “Apriltag: A robust and flexible visual fiducial system,” in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3400–3407.
- [28] S. Cruciani, B. Sundaralingam, K. Hang, V. Kumar, T. Hermans, and D. Kragic, “Benchmarking in-hand manipulation,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 588–595, 2020.