

Vision-driven Compliant Manipulation for Reliable, High-Precision Assembly Tasks

Andrew S. Morgan^{1*}, Bowen Wen^{2*}, Junchi Liang²,
Abdeslam Boularias², Aaron M. Dollar¹, and Kostas Bekris²

¹Department of Mechanical Engineering and Materials Science, Yale University, USA
Email: {andrew.morgan, aaron.dollar}@yale.edu

²Department of Computer Science, Rutgers University, USA
Email: {bw344, jl2068, ab1544, kostas.bekris}@cs.rutgers.edu

Abstract—Highly constrained manipulation tasks continue to be challenging for autonomous robots as they require high levels of precision, typically less than $1mm$, which is often incompatible with what can be achieved by traditional perception systems. This paper demonstrates that the combination of state-of-the-art object tracking with passively adaptive mechanical hardware can be leveraged to complete precision manipulation tasks with tight, industrially-relevant tolerances ($0.25mm$). The proposed control method closes the loop through vision by tracking the relative 6D pose of objects in the relevant workspace. It adjusts the control reference of both the compliant manipulator and the hand to complete object insertion tasks via within-hand manipulation. Contrary to previous efforts for insertion, our method does not require expensive force sensors, precision manipulators, or time-consuming, online learning, which is data hungry. Instead, this effort leverages mechanical compliance and utilizes an object-agnostic manipulation model of the hand learned offline, off-the-shelf motion planning, and an RGBD-based object tracker trained solely with synthetic data. These features allow the proposed system to easily generalize and transfer to new tasks and environments. This paper describes in detail the system components and showcases its efficacy with extensive experiments involving tight tolerance peg-in-hole insertion tasks of various geometries as well as open-world constrained placement tasks.

I. INTRODUCTION

Developing robotic systems capable of operating in contact-rich environments with tight tolerances has remained an open research challenge. Despite progress, this is especially true for precision manipulation, where an object must be perceived, grasped, manipulated, and then appropriately placed given task requirements [1, 2, 3]. Such functionality is common for everyday insertion tasks e.g. stacking cups into one another, placing a key into a lock, or packing boxes, which are skills particularly advantageous to investigate for developing more capable robots in various application domains [4, 5].

A robotic system’s compliance, i.e., the adaptability of its kinematic configuration, has been key in dealing with uncertainty. Failures during manipulation are often attributed to uncertainty introduced in internal modeling, sensor readings, or robot state, which makes the system act undesirably in contact-rich scenarios, such as peg-in-hole or assembly tasks.

This work was supported by the U.S. National Science Foundation Grants IIS-1752134 & IIS-1900681.

* Authors contributed equally

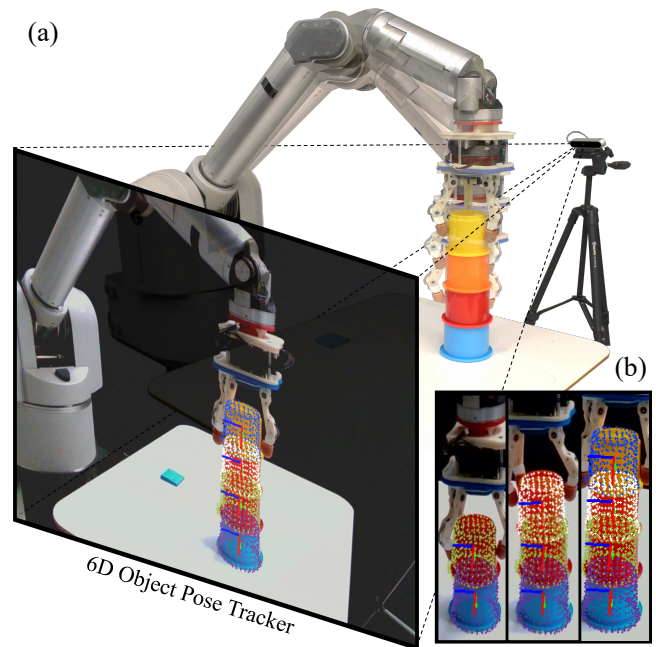


Fig. 1: (a) An RGBD-based 6D object pose tracker monitors the task state, serving as the primary sensing modality for the robot performing a variety of insertion tasks with tight tolerances, (b) the sequence of cup stacking.

Mechanical compliance, unlike its algorithmic counterpart [6, 7], is inherent to the robot’s design and enables the system to passively reconfigure when external contact is enacted [8, 9]. Although beneficial, the robot typically lacks otherwise required sensing modalities for precision control, i.e., joint encoders and force sensors can be absent. This poses a limitation, as there exists little knowledge about the robot’s true state [10, 11, 12], which complicates the achievement of sub- mm accuracy.

Estimating the task’s state dynamically, such as the object’s 6D pose [13, 14, 15], when onboard sensing is unavailable can be achieved through feedback from inexpensive, external alternatives, such as RGBD cameras [16, 17]. Although requiring additional computation, this sensing modality is advantageous as it does not require invasive, bulky sensor suites on the robot, while also providing a wider “field of sensing” for perceiving an extended workspace.

This work investigates whether a framework for compliant

systems using exclusively visual feedback can perform precision manipulation tasks. It showcases a complete system and conducts numerous peg-in-hole insertions of varied geometries, in addition to completing several open world constrained placement tasks. In this way, it tests the boundaries of what is possible for vision-driven, compliance-enabled robot assembly, focusing on three main goals: 1) Perform tight tolerance insertion tasks without force sensing, a precise manipulator, or task-specific, online learning; 2) Leverage the extended workspace afforded by compliant, within-hand manipulation to increase the reliability of insertion; and, 3) Demonstrate the role of vision-based feedback for tight tolerance tasks with system compliance. To accomplish these goals, this effort brings together the following components:

- **Vision-based Object Pose Tracker:** Utilizes state-of-the-art low-latency RGBD-based 6D object pose tracking [15], which is shown to be accurate enough when combined with mechanical compliance to solve the target tasks. The visual tracking monitors the task’s state in the form of the object’s 6D pose in a way that is robust to occlusions and varied lighting conditions, while trained solely offline on simulated data (Sec. III-A).
- **Object-Agnostic Within-Hand Manipulation:** Uses a learned model of the inverse system dynamics of an open source and underactuated dexterous hand that is object agnostic [18]. The model is used to perform within-hand manipulation for object orientation alignment that facilitates insertion (Sec. III-B).
- **Visual Feedback Controller:** Develops an insertion control strategy that relies exclusively on the task’s state, which closes the loop through the object tracker’s 6D pose estimate and generalizes to objects of differing geometries so that it is applicable to different scenarios (Sec. IV-B). The controller intentionally leverages contacts with the environment and compliance in order to increase reliability, similar to the notion of extrinsic dexterity [19].

This article evaluates the efficacy of the complete system by performing a variety of high-precision manipulation tasks. The results in Sec. V showcase that the proposed general object insertion algorithm, which relaxes rigid insertion constraints due to compliance, exhibits a high rate of success for tight tolerance applications. There are demonstrations indicating that the system further generalizes to a variety of everyday precision placement tasks – stacking cups, plugging a cord into an outlet, inserting a marker into a holder, and packing boxes – underscoring the system’s practical use for complex manipulation scenarios.

II. RELATED WORK

Strategies for peg-in-hole insertion, or more generally robot assembly tasks, have been studied from numerous viewpoints for several decades raising many challenges [1]. Insertion strategies for pegs of various geometries have been previously studied and attempted via a variety of possible techniques and system models: standard cylinders [20, 21, 22], multiple-peg objects [23, 24], soft pegs [25], industrial inserts [26], and

open world objects [27, 28, 29, 30, 31]. This work seeks to generalize insertion for various geometries.

Model-Based Insertion: Approaches using contact models reason about state conditions and optimal insertion trajectories while controlling the manipulator [24, 25]. These techniques typically require expensive force-torque sensing to detect peg-hole interactions [21, 32]. Visual-based methods have been less successful, as estimation uncertainty often causes the manipulator to apply unwanted forces either damaging the robot or its environment [33].

Compliance-enabled architectures, both hardware-based [27, 29] and software-based [20], are widely used to overcome uncertainties in modeling, sensing, and control of a purely rigid system. Such solutions can be applied to the manipulator, the end effector, or both. The majority of previous peg-in-hole insertion works investigate compliance in the manipulator’s control – fixing the object directly to the manipulator and evaluating trajectory search spaces or force signatures [20, 32]. Principally, these works do not address robot assembly tasks, as an end effector is vital for the completion of a pick-and-place style objective. Few works have previously investigated the role of within-hand manipulation for peg-in-hole [34], finding that compliance in a dexterous hand extends the workspace for task completion success. This prior work utilizes a rigid hand with tactile sensors and impedance control, which is computationally inefficient compared to the proposed framework.

Learning-Based Insertion: Regardless of whether a system is rigid or compliant, developing the control policies associated with tight tolerance tasks remain difficult [33]. To address this, learning has been widely performed on such systems – either collecting data through robot interactions in its environment or from human-in-the-loop demonstrations [22]. Reinforcement learning and self-supervised learning [35, 36] have been popular, as they enable optimal policy acquisition without manual data intervention. Conceptually, this feature can be favorable for manipulation learning, as it allows the robot to sufficiently explore its environment and collect representative data of its interactions [26, 28, 37, 38]. But it is also time consuming, computationally expensive, and the extensive number of interactions needed to learn in a physical environment increases the likelihood of robot damage. It is thus advantageous to develop planning and control techniques that do not require task-specific learning or numerous physical interactions, allowing the system to more easily generalize to novel tasks and environments, as the method proposed here.

Visual Feedback Closed-loop Manipulation: Tremendous progress has been made to adapt control policies reactively with sensing feedback. With recent advances in deep learning, a number of prior works learn a visuomotor controller either by directly mapping raw image observations to control policies [39, 40, 41, 42] or by using reinforcement learning based on learned visual dynamics models [43, 44, 45, 46]. However, this method usually requires a nontrivial amount of training data which is costly to collect in the real world, struggles to transfer to new scenarios, and suffers from the curse of under-modeling or modeling uncertainties [47]. Our work shares the

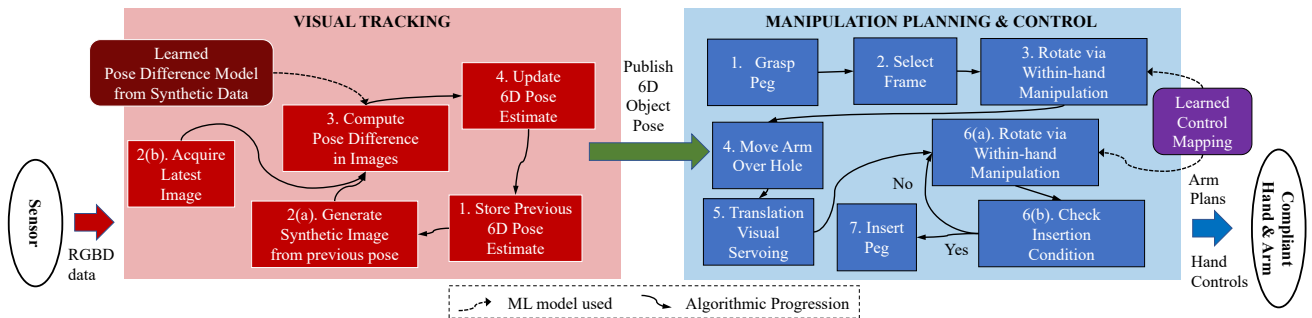


Fig. 2: **System pipeline:** (red) a visual tracking framework trained solely with synthetic data to estimate 6D pose differences, provides feedback for (blue) manipulation planning and control of a low-impedance manipulator and a compliant end-effector performing within-hand manipulation for insertion tasks.

spirit of another line of research that decouples the system into individual sensing and planning components. In particular, [48] developed a compliant manipulation system by integrating a 6D object pose tracker and a reactive motion planner. To generalize to objects of unknown shapes, [49] presents a system that maintains a dual representation of the unseen object’s shape through visual tracking, achieving constrained placement. Although promising results of manipulating one or a few objects have been shown, their scalability to other objects or precision placement tasks remains unclear. In [50], both vision and torque feedback were used for tight insertion. Our work aims instead to robustly tackle a wide range of high precision placement tasks by leveraging the synergy between mechanical compliance and visual feedback, without force or torque feedback.

III. PROPOSED FRAMEWORK

The proposed framework integrates the following components to complete tight tolerance and open world insertion tasks: a 6D object pose tracker given RGBD data, and a compliance-enabled insertion algorithm for a passively compliant arm and dexterous hand. As depicted in the system pipeline (Fig. 2), during manipulation, the object tracker (red) asynchronously estimates object poses in the task space, i.e., both the peg and the hole, based on a learned pose difference model, and provides feedback for the insertion planning and control algorithm (blue) to compute action decisions based on an arm motion planner and a learned control mapping of the hand. The following subsections describe these components in more detail.

A. Visual 6D Object Pose Tracking based on Synthetic Training

Uncertainty about the task’s state can arise from multiple sources in the target application: (a) the compliant robot arm, (b) the adaptive hand, (c) the potentially occluded object, and (d) the location of the hole. Therefore, dynamic reasoning about the spatial relationship between the peg and the hole is required to achieve reliable tight insertion. This work leverages recent advances in visual tracking that employ temporal cues to dynamically update the 6D pose of tracked objects. In particular, recent work in visual tracking [15] achieves robust and accurate enough estimates at a low latency to work with a wide range of objects. This allows easy integration of visual tracking with planning and control for closing the feedback loop. Additionally, it is also possible to disambiguate the 6D pose of

geometrically symmetric objects from semantic textures, thanks to jointly reasoning over RGB and depth data, enabling the overall system to perform a wider range of tasks. Alternative 6D pose tracking methods, which are based solely on depth data [51, 52], often struggle with this aspect. For instance, the green charger (Fig 6) exhibits a 180° shape symmetry from the top-down view, whereas only one of the orientations can result in successfully plugging it into a power strip.

The tracker requires access to a CAD model of the manipulated object for training purposes but does not use any manually annotated data. It generalizes from synthetic data to real-world manipulation scenarios. CAD models for this purpose can be imperfect and obtained through inexpensive depth scanning processes [53] or from online CAD libraries. The dimensional error of the CAD model can be larger than that of the tight insertion task considered – given the adaptability afforded to the system via compliance (see plug insertion in Sec. V-C). At time t , the visual tracker operates over a pair of RGBD images I_t and I_{t-1} and predicts the relative pose of the tracked objects parametrized by a Lie Algebra representation $\Delta\xi \in se(3)$. The 6D object pose in the camera’s frame is then recovered by $T_t = exp(\Delta\xi)T_{t-1}$. During both training and testing, I_{t-1} is a synthetically rendered image given the pose T_{t-1} and the CAD model. During training, I_t is also synthetic but it is the real image during online operation of the system. Thus, a sim-to-real domain shift exists only for the current I_t image.

The synthetic data generation process is physics-aware and aims for generalization and high-fidelity by leveraging domain randomization techniques [54], as shown in Fig. 3. To do so, external lighting positions, intensity, and color are randomized. The training process includes distractor objects from the YCB dataset [55] beyond the targeted objects for manipulation, which introduce occlusions and a noisy background. Object poses are randomly initialized and perturbed by physics simulation with random gravity directions until no collision or penetrations occurs. Background wall textures are randomly selected [56] for each rendering. This rendered image serves as the frame I_t . In order to generate the paired input image I_{t-1} for the network, the process randomly samples a Gaussian relative motion transformation $T_{t-1}^t \in SE(3)$ centered on the identity relative transform to render the prior frame I_{t-1} . During training, data augmentations involving random HSV shift,

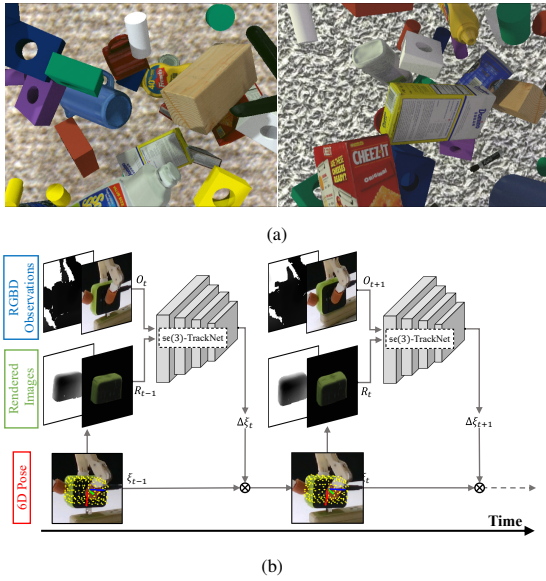


Fig. 3: (a) Physics-aware, high-fidelity synthetic training data are augmented via domain randomization. (b) 6D pose tracking on RGBD image observations streamed from the camera.

Gaussian noise, Gaussian blur, and depth-sensing corruption are applied to the RGB and depth data in frame I_t , following bi-directional domain alignment techniques [15]. Training on synthetic data takes 250 epochs and is readily applicable to real world scenarios without fine-tuning.

During the task execution, the proposed process tracks all manipulated objects at $30Hz$, and provides 6D poses for the planning and control module (Fig. 2). Before grasping, the initial 6D pose of each object on the support surface is estimated once via RANSAC-based plane fitting and removal [57], followed by a single-shot pose estimation approach [58] to initialize tracking. The tracking process is robust to occlusions and does not require pose re-initialization during manipulation, which is continuously executed until task completion.

B. Learning Object-Agnostic Within-hand Manipulation

The employed within-hand manipulation model of a 3-fingered, tendon-driven underactuated hand described is object-agnostic. It relies solely on the reference velocities of the object to suggest manipulation actions [18]. This learned model is realized by first generalizing grasp geometry – the object is represented according to the relative pose of the fingertip contacts after a grasp is acquired. From these contacts, it is possible to strictly define an object frame, \mathcal{X} , according to Gram-Schmidt orthogonalization. More formally, given k points of contact, $\mathcal{P} = p_1, \dots, p_k$ where $p_i \in \mathbb{R}^3, \forall i \in \{1, \dots, k\}$ between the fingertips and the object with respect to the hand frame, we use \mathcal{P} to calculate \mathcal{X} in a closed form [59]. Notably, the relative position of contacts in \mathcal{P} can sufficiently represent the local geometry of the object in its manipulation plane. From this observation, it is possible to calculate the contact triangle relationship, or distance between the fingertips,

$$\mathcal{T} = (\|p_1 - p_2\|_2, \|p_2 - p_3\|_2, \|p_3 - p_1\|_2) \in \mathbb{R}^3 \quad (1)$$

where $\mathcal{T} = (\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3)$ (Fig. 4). Note that this representation generalizes object geometry, but not necessarily object dynam-

ics, as the global geometry, Γ , and associated inertia terms of the object are disregarded for simplicity.

Given the object frame $\mathcal{X}_t \in SE(3)$ at time t , it is possible to model the configuration, and thus the next-state object frame \mathcal{X}_{t+1} of an underactuated system given an actuation velocity \dot{a} based on the system’s energy. With the hand’s joint configuration, $q \in \mathbb{R}^{\sum_{i=1}^k j_i}$, which has j_i joints per finger, the system’s equilibrated joint configuration q^* can be calculated such that the sum of potential energies between the fingers is minimized. As in any mechanically compliant mechanism, an underactuated system’s degrees of actuation are fewer than that of its configuration, i.e., $dim(a) < dim(q)$. For the adaptive hand, the fingers are actuated via a tendon transmission routed through the joints, providing a constraint:

$$r_{ai}\dot{a}_i = r_{pi}\dot{q}_{pi} + r_{di}\dot{q}_{di}, \quad (2)$$

where r_{ai}, r_{pi}, r_{di} represent pulley radii of the actuator and joints in finger i , respectively, and $\dot{a}_i, \dot{q}_{pi}, \dot{q}_{di}$ are the velocities of the actuator and joints, respectively (Fig. 4). Given this tendon constraint, and while ensuring $\mathcal{T}_t = \mathcal{T}_{t+1}$, i.e., by maintaining integrity on the contact triangle relationship, it is possible to solve for the equilibrated joint configuration:

$$q^* = \arg \min_q \sum_i E^i(q^i) \quad s.t. \quad (1), (2), \quad (3)$$

where E^i represents the potential energy in the i^{th} finger:

$$E^i(q^i) = \frac{1}{2}(k_p q_{pi}^2 + k_d q_{di}^2). \quad (4)$$

Through Eq. (3), which has been shown to successfully transfer physically to an underactuated hand [18], this work efficiently generates system dynamics data by varying relationships in \mathcal{T} and providing random actuation velocities \dot{a} to the hand. In doing so, we fill a buffer of 200k object transitions, $(\mathcal{X}_t, \dot{a}) \rightarrow \mathcal{X}_{t+1}$, from 50 contact triangle relationships. By taking the element-wise difference of \mathcal{X}_t and \mathcal{X}_{t+1} , we calculate $\dot{\mathcal{X}} \in se(3)$. Given these action-reaction pairs, we train a fully-connected network to compute the model, or partially constrained Jacobian:

$$g : (\dot{\mathcal{X}}_x, \dot{\mathcal{X}}_y) \rightarrow \dot{a} \quad (5)$$

that maps the desired rotational velocity of the object about the x - and y -axes to an actuation velocity of the hand, $\dot{a} \in \mathbb{R}^3$.

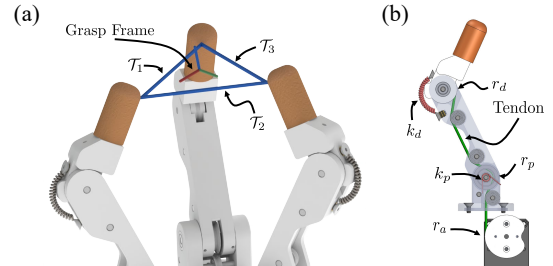


Fig. 4: (a) Object geometry can be generalized by its resultant contact triangle relationship, \mathcal{T} . (b) The response of a tendon-driven underactuated finger given actuation is dependent on spring constants and pulley radii.

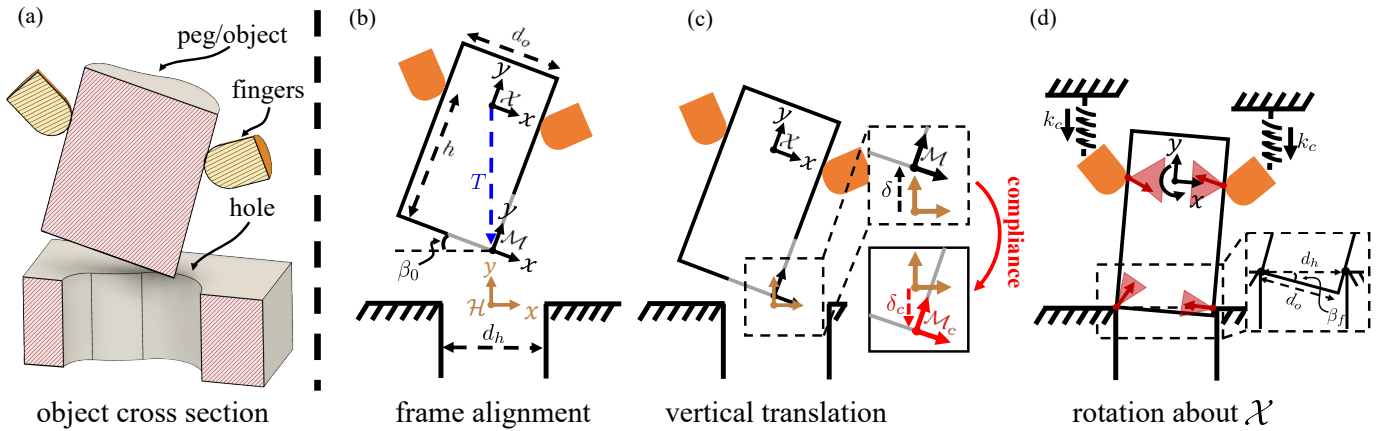


Fig. 5: (a) Peg insertion is viewed as a planar challenge. (b) For insertion via purely rotational motion about \mathcal{X} , the peg's edge is aligned directly above the hole with a starting angle, β_0 . (c) Translating \mathcal{M} downward to δ guarantees that a pure rotation will align the peg with the hole, where δ_c encourages premature contact and leverages compliance to aid in alignment. (d) Upon rotation about \mathcal{X} , the peg must overcome contact constraints (red friction cones) of the hole contacts to align for insertion, while aided by virtual spring forces k_c supplied by compliance.

IV. INSERTION STRATEGY

Assume the geometry of the peg, Γ , consists of two opposing, parallel faces with a continuous or discrete set of sidewalls connecting them, where, for the set of all antipodal point contacts on the object's sidewalls, each pair in the set has parallel contact normal force vectors. Conceptual examples of Γ could include standard cylinders or triangular prisms. This section analyzes spatial peg insertion as a planar problem as depicted in Fig. 5, i.e., along cross sections of the object. This approach leverages the notion that in practice we can rely on compliance and closed loop control to account for any out of plane misalignment as described in Sec. IV-B.

A. The Condition for Object Insertion

Consider that a rigid peg is grasped such that its antipodal, i.e. set of opposing, contacts are distanced d_o from one another across the width of the object. Moreover, the location of the contacts defines the height, h , of the grasp. Given d_o and h , it is possible to define the object's grasp frame, $\mathcal{X} \in SE(2)$, with a pose determined by the grasp-contact vector. The task goal is to insert a peg into its corresponding rigid hole, $\mathcal{H} \in SE(2)$, which is parameterized by a width of d_h . Given Γ , it is possible to define a manipulation frame, $\mathcal{M} \in SE(2)$, defined by a transformation, T , from \mathcal{X} , that acts as the initial controlled frame for object insertion. Determining \mathcal{M} for generalized geometries is achieved through PCA, as discussed in the accompanying Appendix.

In the context of insertion tasks, this work investigates the role of object rotation via within-hand manipulation while keeping the relative translational pose between the object frame and the hole fixed. To do so, the process vertically aligns \mathcal{H} , \mathcal{M} , and \mathcal{X} such that the horizontal component of T is equal to 0. This therefore sufficiently defines an initial object rotation and sets up spatial peg insertion as a planar problem with a starting angle:

$$\beta_0 = \tan^{-1}\left(\frac{d_o}{2h}\right) \quad (6)$$

to satisfy the alignment constraint. It is possible to start in a different angle than β_0 , but the position of \mathcal{M} will need to

change accordingly. Once β_0 is achieved, the system controls a vertical displacement, δ , between \mathcal{M} and \mathcal{H} , which can be calculated in closed form such that a pure rotation about \mathcal{X} places the object in a state aligned with the hole. This final angle of alignment, β_f , can be calculated as:

$$|\beta_f| \leq \cos^{-1}\left(\frac{d_o}{d_h}\right), \quad (7)$$

which is dependent on the peg's two-contact case (Fig. 5.d). Thus, the initial insertion height δ before rotation is given as

$$\delta = h(\cos(\beta_0) + \cos(\beta_f)) + \frac{d_o}{2}(\sin(\beta_0) + \sin(\beta_f)). \quad (8)$$

The sequential progression of steps for rotation-based object insertion are outlined in Fig. 5. In summary: 1) Rotate \mathcal{M} such that it aligns with \mathcal{H} ; 2) Translate \mathcal{M} downward to δ ; and, 3) Rotate about \mathcal{X} until the two-contact case is overcome, i.e., rotation of $\mathcal{X} < |\beta_f|$, and the peg is aligned with the hole. Once satisfied, the remaining actions for insertion seek object translation downward while avoiding jamming along the sidewalls. To account for this, this work leverages compliance and controlled object spiral motions [32] provided by within-hand manipulation.

On the Role of Compliance: While the above formulation holds for rigid insertion in the relaxed planar representation, the bounds of δ are extremely small for tight tolerance cases ($< 1.5mm$ if perfectly aligned). Moreover, when considering any out-of-plane misalignment in the true additional dimension that is not modeled, the peg can easily become off-centered with respect to the hole. The features inherent to compliance benefit this task, as it not only enables adaptability to the out-of-plane misalignment, but also allows to relax these strict bounds of δ by developing a compliance-enabled insertion distance, δ_c (Fig. 5.c). This deviation in depth is nontrivial to determine in its closed form – as it depends on models of the manipulator-hand system's compliance, the forces that can be applied to the object via the hand, and an approximation of the nonlinear contact dynamics associated with peg-hole interactions (Fig. 5.d). Although difficult to model, a properly tuned translational deviation in δ_c towards the hole from δ is

advantageous, as it encourages contact between the hole and the object prematurely so that the contact constraints encourage and assist in alignment before insertion.

B. Insertion Algorithm

The proposed system is controlled by closing the loop through visual feedback. In real time, the visual tracker monitors the state of the task space; specifically, the relative pose between the peg and the hole. This relationship is used to adjust the control reference setpoints of both the manipulator’s joints and the tendons in the hand, regardless of their believed configuration states. By continually servoing the object’s $SE(3)$ pose relative to the hole, it is possible to complete tight insertion tasks while adapting to external disturbances and noisy pose estimates (Sec. V).

Vision-Driven Object Insertion: The insertion sequence shown in Alg. 1 begins with the object tracker asynchronously monitoring the pose of both, the object and the hole. Once the object is grasped off of the support surface, the precomputed transformation, T , via PCA (9), from the manipulation frame, \mathcal{M} , can be rigidly attached to \mathcal{X} .

Algorithm 1 Vision-Driven Object Insertion

Input: $\Gamma, (\beta_0, \beta_f, \delta_c, \gamma, \sigma)$ ▷object geometry,
hyparams(initial object angle, final object angle, insertion
depth, alignment tolerance, step length)

- 1: $T, \pi_1 \leftarrow PCA(\Gamma)$ ▷POM transform, principal axis (9)
- 2: **tracker.start_async_perceive**(Γ) ▷start tracking thread
- 3: **tracker.attach_transform**(T) ▷attach T to \mathcal{X} for \mathcal{M}
- 4: $\mathcal{X}, \mathcal{H} \leftarrow \mathbf{tracker.get_poses}()$ ▷object & hole 6D poses
- 5: **system.grasp**(\mathcal{X}, π_1) ▷grasp and lift object
- 6: **hand.rotation_servo**(**tracker**, π_1, β_0) ▷Alg. 2
- 7: $\mathcal{M}, \mathcal{H} \leftarrow \mathbf{tracker.get_poses}()$
- 8: **arm.move_above**(\mathcal{M}, \mathcal{H}) ▷place edge above hole
- 9: **arm.translation_servo**(**tracker**, δ_c, γ, σ) ▷Alg. 3
- 10: **hand.rotation_servo**(**tracker**, π_1, β_f) ▷Alg. 2
- 11: **system.spiral_insertion**() ▷coordinated spiral insertion

Upon doing so, within-hand manipulation is performed such that the initial insertion angle, β_0 from Sec. IV-A is achieved along the principal axis, π_1 (Alg. 2). The velocity references, $\dot{\mathcal{X}}_x$ and $\dot{\mathcal{X}}_y$, during this within-hand manipulation process are chosen so as to minimize any rotation not along this principal axis. Upon reaching β_0 , the robot arm plans a trajectory such that the resultant position of the manipulation frame is directly above the hole. Due to the robot’s imprecision, however, the desired pose of \mathcal{M} above \mathcal{H} is often not accurately achieved.

At this point, the *translation_servo* method begins by sequentially adjusting the control reference of the manipulator based on feedback from the object tracker (Alg. 3). Specifically, given the hyper-parameter γ , if the hole xy -plane translational difference between \mathcal{M} and \mathcal{H} is within γ , the manipulator will move a step, σ , which can be fixed or adaptive, downward towards the hole by adjusting the joint target values of the robot. Otherwise, the robot will move in the Cartesian step,

σ , towards aligning \mathcal{M} with \mathcal{H} . This process is continually repeated until the vertical threshold, δ_c , is reached.

Algorithm 2 Within-Hand Rotation Visual Feedback Control

Input: **tracker**, $\pi_1, \theta \in \{\beta_0, \beta_f\}$

▷principal axis of rotation, object target angle

- 1: $\mathcal{X}, \mathcal{H} \leftarrow \mathbf{tracker.get_poses}()$ ▷object and hole 6D pose
- 2: $R_\Delta \leftarrow \mathcal{X}.R - \mathcal{H}.R$ ▷relative rotation $\in SO(3)$
- 3: **if** θ **is** β_0 **then** ▷increase angle for edge insertion
- 4: **while** $\mathcal{H}.R(\pi_1) - \mathcal{X}.R(\pi_1) \leq \beta_0$ **do**
- 5: $\dot{\mathcal{X}} \leftarrow [-R_\Delta(\pi_{1x}), -R_\Delta(\pi_{1y})]$ ▷rotate along π_1
- 6: $\dot{a} \leftarrow \mathbf{hand.model.predict}(\dot{\mathcal{X}})$ ▷Eq. (5)
- 7: **hand.actuate**(\dot{a}) ▷send action to motors
- 8: $\mathcal{X}, \mathcal{H} \leftarrow \mathbf{tracker.get_poses}()$
- 9: $R_\Delta \leftarrow \mathcal{X}.R - \mathcal{H}.R$
- 10: **if** θ **is** β_f **then** ▷decrease angle, align with hole
- 11: **while** $R_{\Delta x} \geq \beta_f$ **or** $R_{\Delta y} \geq \beta_f$ **do**
- 12: $\dot{\mathcal{X}} \leftarrow [R_{\Delta x}, R_{\Delta y}]$
- 13: $\dot{a} \leftarrow \mathbf{hand.model.predict}(\dot{\mathcal{X}})$
- 14: **hand.actuate**(\dot{a})
- 15: $\mathcal{X}, \mathcal{H} \leftarrow \mathbf{tracker.get_poses}()$
- 16: $R_\Delta \leftarrow \mathcal{H}.R - \mathcal{X}.R$

Algorithm 3 Arm Translation Visual Feedback Control

Input: **tracker**, δ_c, γ, σ

▷insertion depth, alignment tolerance, step length

- 1: $\mathcal{M}, \mathcal{H} \leftarrow \mathbf{tracker.get_poses}()$
- 2: **while** $\mathcal{M}.t_z - \mathcal{H}.t_z > \delta_c$ **do** ▷check translation along z
- 3: $\mathcal{E}_{\Delta x}, \mathcal{E}_{\Delta y}, \mathcal{E}_{\Delta z} \leftarrow 0, 0, 0$
- 4: **if** $-\gamma \leq \mathcal{M}.t_x - \mathcal{H}.t_x \leq \gamma$ **and**
 $-\gamma \leq \mathcal{M}.t_y - \mathcal{H}.t_y \leq \gamma$ **then**
- 5: $\mathcal{E}_{\Delta z} \leftarrow -\sigma$ ▷servo end effector pose downward
- 6: **else**
- 7: $\epsilon_x \leftarrow \mathcal{X}.t_x - \mathcal{H}.t_x$
- 8: $\epsilon_y \leftarrow \mathcal{X}.t_y - \mathcal{H}.t_y$
- 9: $\mathcal{E}_{\Delta x}, \mathcal{E}_{\Delta y} \leftarrow -\text{sgn}(\epsilon_x)\sigma, -\text{sgn}(\epsilon_y)\sigma$
 ▷servo translation to align with hole
- 10: $\mathcal{A} \leftarrow \mathbf{arm.motion_planner}(\mathcal{E}_\Delta)$ ▷plan motion delta
- 11: **arm.execute**(\mathcal{A}) ▷move end effector
- 12: $\mathcal{M}, \mathcal{H} \leftarrow \mathbf{tracker.get_poses}()$

The hand then attempts to reorient \mathcal{X} with \mathcal{H} while maintaining a small downward force between the object and the hole’s edges, provided by system compliance and δ_c . Orientation alignment of the peg is solely achieved through within-hand manipulation (Alg. 2); where the object no longer follows rotations along π_1 , but now by the true rotational difference R_Δ between the hole and the peg. This process continues until the rotational angle is less than β_f .

The algorithm concludes by performing a spiral insertion technique along the roll and pitch axes of the object. If the object were unconstrained, the hand’s actuation would provide a spiral pattern of \mathcal{M} , as in related work [32, 20], while the arm attempts to slowly translate downward. This coordinated hand/arm motion helps limit jamming and encourages proper insertion until the object is fully seated into the hole.

V. EXPERIMENTS

We test the proposed insertion framework using a robotic system comprised of a low-impedance manipulator for object translation, an underactuated hand performing within-hand manipulation for object rotation, and a low latency 6D object pose tracker based on RGBD data (Figs. 1, 7). The manipulator, a 7-DOF Barrett WAM arm, utilizes the *RRTCConnect* algorithm in OMPL [60] and is imprecise due to an inaccurate internal model of its true system dynamics, with translational errors as large as 2.6cm. The end effector, an adapted Yale OpenHand Model O [61], is a mechanically adaptive hand comprised of six joints and three controlled actuators, and is not equipped with joint encoders or tactile sensors. Changes to the open source design include joint bearings to reduce friction, and rounded fingertips to assist in within-hand manipulation. Finally, the 6D object pose tracker, which was calibrated to the robot’s world frame, takes the RGBD images streamed from an external, statically mounted Intel Realsense D415 camera, and in real-time, estimates the 6D pose of the manipulated object to provide feedback for the control loop.

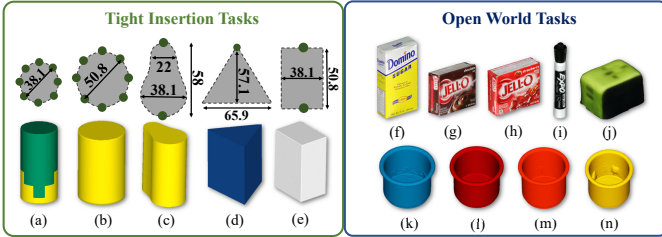


Fig. 6: Experimental objects considered in the **Tight Tolerance Tasks** and **Open World Tasks**. All lengths are in *mm* and points on object faces indicate PCA-determined edge manipulation frames. (a) small circle, (b) large circle, (c) pear, (d) triangle, (e) rectangle, (f) YCB *004_sugar_box*, (g) YCB *008_pudding_box*, (h) YCB *009_gelatin_box*, (i) YCB *040_large_marker*, (j) green charger. (k)-(n) YCB *065-cups*.

The system is tested via numerous insertion tasks. First, five 3D printed peg-in-hole objects were designed with $< 0.25mm$ hole tolerances and were painted different colors and/or patterns, for evaluation on both textured and textureless objects. These objects are described based on their face geometries: namely, the small circle, large circle, pear, triangle, and rectangle (Fig. 6). Insertion was tested for each of these objects individually, where we then sequentially isolate individual system components – compliance, control, and sensing – to evaluate their effects on task success. Finally, we assessed the efficacy of the approach with six open world insertion tasks, involving nine different objects with diverse and challenging properties (textureless, reflective, flat and thin shapes, etc.), to underscore the utility of the framework in complex manipulation scenarios (Fig. 7).

A. Tight Tolerance Object Insertion

This test involved 12 insertions for each of the five objects using Alg. 1. Upon task reset, objects were placed back onto the support surface in no predefined pose; it was up to the system to initialize and track this pose and reacquire a grasp. The results, presented in Table I, depict the planning time, total execution time, number of hand actions used for within-hand

TABLE I: PRECISION INSERTION FOR PROPOSED SYSTEM

Obj.	Planning (s)	Total (s)	Hand Actions	Success
Small Cir.	4.1 ± 0.92	94.8 ± 10.78	48.9 ± 8.76	10/12
Large Cir.	3.5 ± 0.52	88.8 ± 11.54	44.9 ± 9.10	11/12
Pear	4.0 ± 1.05	77.9 ± 10.34	35.2 ± 8.62	9/12
Triangle	3.7 ± 2.08	90.9 ± 14.00	27.4 ± 4.60	8/12
Rectangle	6.2 ± 3.53	106.4 ± 23.43	40.9 ± 5.90	9/12

manipulation, and success rate for each object scenario. The large circle had the highest rate of success compared to any of the other objects, while the triangle had the lowest. The two objects that did not have curved edges, i.e., the triangle and the rectangle, were the most difficult out of the five to insert. The interpretation is that the constraints of the task, i.e., sharp edges of the object’s face, did not enable system compliance to easily align the yaw rotation of the object with the hole. This posed a slight challenge for the relaxed planar insertion strategy from Sec. IV-A, and required precise yaw rotation via the manipulator, which was not required for the other objects.

TABLE II: SYSTEM ABLATION ANALYSIS

	Obj.	Planning (s)	Total (s)	Hand Actions	Success
Reducing Compliance	Rigid Hand / Compliant Arm / Rigid Hole				
	Large Cir.	7.8 ± 1.65	62.3 ± 8.74	-	6/12
	Pear	11.0 ± 6.76	76.4 ± 28.10	-	5/12
	Rectangle	12.6 ± 4.37	92.3 ± 24.95	-	3/12
	Rigid Hand / Rigid Arm / Compliant Hole				
	Large Cir.	1.4 ± 0.12	65.6 ± 3.53	-	9/12
	Pear	1.5 ± 0.13	68.9 ± 3.92	-	4/12
	Rectangle	1.5 ± 0.12	71.3 ± 4.35	-	12/12
	Rigid Hand / Rigid Arm / Rigid Hole				
	Large Cir.	2.2 ± 0.37	88.9 ± 10.48	-	6/12
	Pear	2.2 ± 0.25	91.4 ± 8.46	-	0/12
	Rectangle	2.1 ± 0.14	90.4 ± 13.09	-	2/12
Limiting Control	Naive (omit Alg. 2)				
	Large Cir.	4.8 ± 1.78	78.0 ± 22.64	-	2/12
	Pear	7.7 ± 1.25	81.9 ± 10.13	-	0/12
	Rectangle	8.9 ± 0.95	103.5 ± 5.78	-	0/12
	Open Loop (omit Alg. 2 and Alg. 3)				
	Large Cir.	3.8 ± 1.09	53.47 ± 9.78	-	0/12
Pear	5.1 ± 1.62	61.96 ± 4.01	-	0/12	
Rectangle	5.1 ± 1.22	65.45 ± 5.18	-	0/12	
Noisy Sensing	Uniform Noise – 5mm / 5°				
	Large Cir.	5.9 ± 0.99	127.5 ± 18.11	50.6 ± 18.23	7/12
	Pear	10.7 ± 2.29	138.9 ± 12.46	40.2 ± 7.99	4/12
	Rectangle	9.9 ± 1.70	137.1 ± 6.28	49.7 ± 4.19	4/12
	Uniform Noise – 10mm / 10°				
	Large Cir.	11.7 ± 2.29	148.5 ± 12.46	39.6 ± 11.19	1/12
Pear	12.6 ± 0.33	134.1 ± 8.82	34.9 ± 7.19	0/12	
Rectangle	13.4 ± 1.97	212.6 ± 13.58	51.9 ± 11.79	0/12	

B. System Analysis

Several ablation studies were performed using three of the five evaluated objects – the large circle, the pear, and the rectangle – in order to better understand how different components of the system contribute to task success. In particular, these tests evaluate the effects of: 1) reducing system

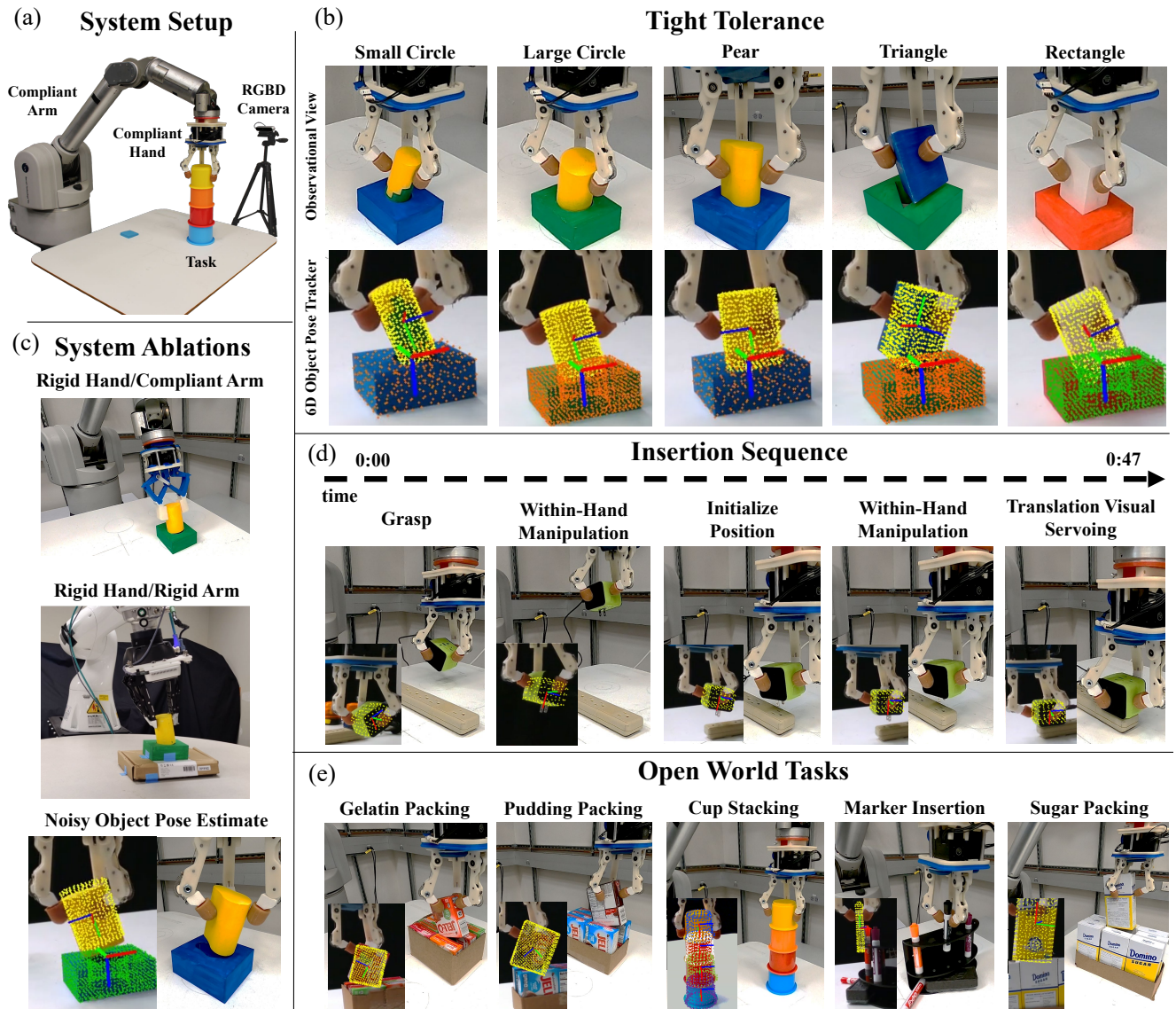


Fig. 7: (a) System setup overview. (b) Tight tolerance insertion of 5 peg geometries, highlighting the observational/external view and the tracked 6D pose via the RGBD camera. (c) System ablations include reducing compliance of the hand and the arm, in addition to deliberately adding noise to the pose estimate. (d) The open world task of plug insertion is highlighted, showcasing the sequence of actions taken from grasp to insertion. (e) Five other open world tasks were also evaluated – box packing, marker insertion, and cup stacking. Please refer to the supplementary video for a complete overview of evaluated tasks.

compliance via the hand, the arm and the environment; 2) performing insertion with differing levels of feedback, i.e., naive and open loop control; and, 3) deliberately introducing noise into the object tracker’s pose estimation to simulate higher perception uncertainty. (Table II). The Appendix describes how the system compensated for disturbances.

Reducing system compliance: To test the role of compliance for task success, three altered system configurations were developed: 1) A system with a rigid, parallel Yale Openhand Model GR2 gripper [62] with custom fingertips as to immobilize the object to the end of a low-impedance manipulator; 2) A system with a rigid 3-fingered Robotiq hand affixed to the end of a rigid Kuka IIWA manipulator but allowing compliance in the environment by attaching a standard packing box at the base of the hole; and, 3) The same rigid robot setup with the Robotiq hand and the Kuka

manipulator but with a fixed hole, removing any presence of compliance. Fig. 9 in the Appendix highlights the different variations considered in terms of compliance. Twelve insertions for each of the three test objects were performed in each case while following Alg. 1. During task execution, all object rotations that were performed by the hand originally, were now controlled solely by the manipulator. Results indicate, as presented in Table II, that compliance at some level in the arm/hand/object system is beneficial for task completion. The worst success rate arises for a fully rigid system, task, and environment. The Kuka setup with a compliant hole performed well, especially for the difficult rectangle insertion task, assisted by the precision of the object tracker and the higher accuracy of the rigid Kuka manipulator. Planning time between systems varied due to differences in computational resources, and thus it is not directly compared.

Naive and open loop control: These tests evaluate the effects of relinquishing feedback by limiting the amount of information transferred from the tracker to the insertion algorithm. This is tested both via naive control and an open loop configuration of the system. Naive control attempted to perform the same sequence of actions as in Alg. 1, but did not use any within-hand manipulation of the object after grasping, i.e., visual servoing was purely translational with the manipulator and Alg. 2 was not used. The compliant hand in this case effectively acted as a remote center of compliance [29, 63]. In open loop testing, a single plan was computed and executed from the starting grasp configuration up until object insertion without utilizing in-hand manipulation or any visual feedback during the task, i.e., neither Alg. 2 or Alg. 3 were used. Notably, both of these ablations performed very poorly in testing, as the imprecision of the manipulator and lack of a controlled object rotation to aid in insertion, drastically impeded task success (Table II).

Noisy pose estimate: The final ablation experiment evaluated how the accuracy of pose estimation played a role in task success. While following the same sequence of actions as in Alg. 1, uniformly sampled noise was introduced into the pose output of the tracker at two different levels. The first test introduced uniform sampled noise within $5mm$ and 5° to the pose estimate. The second test introduced uniform noise within the range of $10mm$ and 10° . As indicated by the results in Table II, the system was largely able to compensate for this noise at the $5mm/5^\circ$ level, successfully completing 7 for the large circle, 4 for the pear, and 4 for the rectangle out of 12 executions. The additional noise from the $10mm/10^\circ$ test drastically decreased success. The total task time for both noise level increased significantly as compared to the baseline trials in Sec. V-A. This increase in execution time is attributed to Alg. 3, where noise continually moves \mathcal{M} outside of the γ threshold insertion region, so the manipulator slowly oscillates back and forth until δ_c is achieved. Conclusively, the ability of the algorithm to complete sub- mm accuracy insertion tasks with purposefully added noise indicates the robustness of the overall framework.

C. Open World Tasks

Finally, a series of open world tasks were attempted to highlight the utility of the proposed system in complex manipulation scenarios: marker insertion, plug insertion, box packing, and cup stacking (Fig. 7(e)). These tests mostly utilize objects contained in the YCB Object and Model Set [55], which provides object CAD models for tracking. The charging plug, however, was 3D scanned using [53] with an inexpensive RGBD sensor, providing a coarse representation of the true object model (Fig. 6). The goal for each task was to grasp, manipulate, and insert the object so as to reach its desired hole configuration, which was predefined depending on task requirements. Of the six tasks, three – gelatin box, pudding box, and sugar box – came in the form of packing, where a single box was removed from the case and had to be replaced. Two other tasks – marker insertion and plug insertion – were

performed such that the marker was placed into a holder and the plug was inserted into an outlet. The most difficult of the open world tasks corresponded to cup stacking. This task requires sequential tracking of both the cup that is to be stacked on top of and the cup being manipulated. The proposed system is able to complete this task placing four cups successfully on top of one another. These evaluations showcase the tracking and manipulation capabilities of the proposed system, which are also highlighted in the supplementary video.

VI. DISCUSSIONS AND FUTURE WORK

This manuscript presented a vision-driven servoing framework that tackles the problem of controlling compliant, passively adaptive mechanisms for precision manipulation. The framework is able to perform 5 tight tolerance and 6 open world tasks – regardless of whether the object CAD model was imperfect or if pose estimate feedback was noisy (Sec. V-C, V-B). In summary, the contributions of this work are fourfold:

- **Precision control with minimal on-board sensing:** The framework can reliably complete an array of insertion tasks – including those with tight tolerances – without force-torque sensors on the manipulator or tactile sensors/joint encoders on the compliant hand.
- **Passively compliant within-hand manipulation for insertion:** The system utilizes controlled hand actions to extend the task’s workspace and provides an added layer of compliance to limit object insertion jamming.
- **Vision-driven feedback controller:** The servoing insertion algorithm is able to generalize to different object geometries, and can be easily utilized for other robot assembly and insertion tasks given compliance.
- **Utilizing the environment to solve the task:** The control strategy intentionally leverages premature contacts by relaxing the rigid constraints of the task, which is possible and effective given compliant systems. In this way, this work applies the principle of “extrinsic dexterity” [19] in the context of insertion tasks with tight tolerances.

There are several aspects of the proposed system that are worth pursuing in future work: 1) Adapt the insertion algorithm so as to better leverage compliance for complex out-of-plane geometry, such as in cases of sharp edges or non-convex objects; 2) Optimize a passively compliant hand design that is capable of performing both finger gaiting and in-hand manipulation with a large workspace; 3) Develop and integrate a model-free perception tracker into the visual feedback framework to reduce dependence on the object’s CAD model; 4) Incorporate learning into the insertion algorithm so as to first automatically tune any object-specific hyperparameters and then incrementally develop more effective strategies in a RL fashion; and finally, 5) Increase difficulty of the tasks by performing more advanced and sequential assembly procedures, such as assembling toys or simple pieces of household furniture.

REFERENCES

- [1] C. C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments [Grand Challenges of Robotics]," *IEEE Robotics & Automation Magazine*, vol. 14, no. 1, pp. 20–29, 2007.
- [2] O. Kroemer, S. Niekum, and G. D. Konidaris, "A Review of Robot Learning for Manipulation: Challenges, Representations, and Algorithms," *CoRR*, vol. abs/1907.03146, 2019. [Online]. Available: <http://arxiv.org/abs/1907.03146>
- [3] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and Observations From the First Amazon Picking Challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2018.
- [4] J. van den Berg, S. Miller, K. Goldberg, and P. Abbeel, *Gravity-Based Robotic Cloth Folding*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 409–424.
- [5] T. Bhattacharjee, G. Lee, H. Song, and S. S. Srinivasa, "Towards Robotic Feeding: Role of Haptics in Fork-Based Food Manipulation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1485–1492, 2019.
- [6] N. Hogan, "Impedance Control: An Approach to Manipulation," in *1984 American Control Conference*, 1984, pp. 304–313.
- [7] T. Wimböck, B. Jahn, and G. Hirzinger, "Synergy level impedance control for multifingered hands," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 973–979.
- [8] 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.
- [9] T. Laliberte, L. Birglen, and C. Gosselin, "Underactuation in robotic grasping hands," *Machine Intelligence & Robotic Control*, vol. 4, no. 3, pp. 1–11, 2002.
- [10] 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.
- [11] H. van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 121–127.
- [12] N. Elangovan, A. Dwivedi, L. Gerez, C. Chang, and M. Liarokapis, "Employing IMU and ArUco Marker Based Tracking to Decode the Contact Forces Exerted by Adaptive Hands," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 525–530.
- [13] H. Tjaden, U. Schwanecke, E. Schömer, and D. Cremers, "A region-based gauss-newton approach to real-time monocular multiple object tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1797–1812, 2018.
- [14] Y. Li, G. Wang, X. Ji, Y. Xiang, and D. Fox, "Deepim: Deep iterative matching for 6d pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 683–698.
- [15] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "se(3)-TrackNet: Data-driven 6D Pose Tracking by Calibrating Image Residuals in Synthetic Domains," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10367–10373.
- [16] F. Chaumette, S. Hutchinson, and P. Corke, *Visual Servoing*. Cham: Springer International Publishing, 2016, pp. 841–866. [Online]. Available: https://doi.org/10.1007/978-3-319-32552-1_34
- [17] B. Calli and A. M. Dollar, "Robust Precision Manipulation With Simple Process Models Using Visual Servoing Techniques With Disturbance Rejection," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 406–419, 2019.
- [18] 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.
- [19] N. Chavan-Dafle, A. Rodriguez, R. Paolini, B. Tang, S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic Dexterity: In-Hand Manipulation with External Forces," in *Proceedings of (ICRA) International Conference on Robotics and Automation*, May 2014, pp. 1578 – 1585.
- [20] H. Park, J. Park, D. Lee, J. Park, M. Baeg, and J. Bae, "Compliance-Based Robotic Peg-in-Hole Assembly Strategy Without Force Feedback," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 8, pp. 6299–6309, 2017.
- [21] T. Tang, H. Lin, Yu Zhao, Wenjie Chen, and M. Tomizuka, "Autonomous alignment of peg and hole by force/torque measurement for robotic assembly," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, 2016, pp. 162–167.
- [22] T. Tang, H. Lin, Y. Zhao, Y. Fan, W. Chen, and M. Tomizuka, "Teach industrial robots peg-hole-insertion by human demonstration," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2016, pp. 488–494.
- [23] J. Xu, Z. Hou, W. Wang, B. Xu, K. Zhang, and K. Chen, "Feedback Deep Deterministic Policy Gradient With Fuzzy Reward for Robotic Multiple Peg-in-Hole Assembly Tasks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1658–1667, 2019.
- [24] Y. Fie and X. Zhao, "An Assembly Process Modeling and Analysis for Robotic Multiple Peg-in-hole," *Journal of Intelligent and Robotic Systems*, vol. 36, pp. 175–189, 2003.
- [25] K. Zhang, J. Xu, H. Chen, J. Zhao, and K. Chen, "Jamming Analysis and Force Control for Flexible Dual Peg-in-Hole Assembly," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 1930–1939, 2019.
- [26] G. Schoettler, A. Nair, J. A. Ojea, S. Levine, and E. Solowjow, "Meta-Reinforcement Learning for Robotic Industrial Insertion Tasks," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9728–9735.
- [27] M. P. Polverini, A. M. Zanchettin, S. Castello, and P. Rocco, "Sensorless and constraint based peg-in-hole task execution with a dual-arm robot," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 415–420.
- [28] S. Levine, N. Wagener, and P. Abbeel, "Learning contact-rich manipulation skills with guided policy search," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 156–163.
- [29] C. Choi, J. Del Preto, and D. Rus, "Using Vision for Pre- and Post-grasping Object Localization for Soft Hands," in *2016 International Symposium on Experimental Robotics*, D. Kulić, Y. Nakamura, O. Khatib, and G. Venture, Eds. Cham: Springer International Publishing, 2017, pp. 601–612.
- [30] W.-C. Chang, "Robotic assembly of smartphone back shells with eye-in-hand visual servoing," *Robotics and Computer-Integrated Manufacturing*, vol. 50, pp. 102–113, 2018.
- [31] Y. She, S. Wang, S. Dong, N. Sunil, A. Rodriguez, and E. Adelson, "Cable Manipulation with a Tactile-Reactive Gripper," in *Robotics: Science and Systems*, 2020.
- [32] Z. Liu, L. Song, Z. Hou, K. Chen, S. Liu, and J. Xu, "Screw Insertion Method in Peg-in-Hole Assembly for Axial Friction Reduction," *IEEE Access*, vol. 7, pp. 148313–148325, 2019.
- [33] J. Xu, Z. Hou, Z. Liu, and H. Qiao, "Compare Contact Model-based Control and Contact Model-free Learning: A Survey of Robotic Peg-in-hole Assembly Strategies," 2019.
- [34] K. Van Wyk, M. Culleton, J. Falco, and K. Kelly, "Comparative Peg-in-Hole Testing of a Force-Based Manipulation Controlled Robotic Hand," *IEEE Transactions on Robotics*, vol. 34, no. 2, pp. 542–549, 2018.
- [35] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making Sense of Vision and Touch: Self-Supervised Learning of Multimodal Representations for Contact-Rich Tasks," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8943–8950.
- [36] K. Zakka, A. Zeng, J. Lee, and S. Song, "Form2Fit: Learning Shape Priors for Generalizable Assembly from Disassembly," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9404–9410.
- [37] G. Thomas, M. Chien, A. Tamar, J. A. Ojea, and P. Abbeel, "Learning Robotic Assembly from CAD," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3524–3531.
- [38] C. C. Beltran-Hernandez, D. Petit, I. G. Ramirez-Alpizar, and K. Harada, "Variable Compliance Control for Robotic Peg-in-Hole Assembly: A Deep-Reinforcement-Learning Approach," *Applied Sciences*, vol. 10, no. 19, p. 6923, Oct 2020. [Online]. Available: <http://dx.doi.org/10.3390/app10196923>
- [39] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.
- [40] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

- [41] U. Viereck, A. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *Conference on Robot Learning*. PMLR, 2017, pp. 291–300.
- [42] 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.
- [43] F. Ebert, C. Finn, S. Dasari, A. Xie, A. Lee, and S. Levine, "Visual foresight: Model-based deep reinforcement learning for vision-based robotic control," *arXiv preprint arXiv:1812.00568*, 2018.
- [44] S. Bechtle, N. Das, and F. Meier, "Learning Extended Body Schemas from Visual Keypoints for Object Manipulation," *arXiv preprint arXiv:2011.03882*, 2020.
- [45] L. Manuelli, Y. Li, P. Florence, and R. Tedrake, "Keypoints into the Future: Self-Supervised Correspondence in Model-Based Reinforcement Learning," *CoRL*, 2020.
- [46] A. Byravan, F. Leeb, F. Meier, and D. Fox, "Se3-pose-nets: Structured deep dynamics models for visuomotor control," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3339–3346.
- [47] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.
- [48] D. Kappler, F. Meier, J. Issac, J. Mainprice, C. G. Cifuentes, M. Wüthrich, V. Berenz, S. Schaal, N. Ratliff, and J. Bohg, "Real-time perception meets reactive motion generation," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1864–1871, 2018.
- [49] C. Mitash, R. Shome, B. Wen, A. Boularias, and K. Bekris, "Task-Driven Perception and Manipulation for Constrained Placement of Unknown Objects," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5605–5612, 2020.
- [50] W. Gao and R. Tedrake, "kPAM 2.0: Feedback Control for Category-Level Robotic Manipulation," *IEEE Robotics and Automation Letter (RA-L)*, 2020.
- [51] J. Issac, M. Wüthrich, C. G. Cifuentes, J. Bohg, S. Trimpe, and S. Schaal, "Depth-based object tracking using a robust gaussian filter," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 608–615.
- [52] M. Wüthrich, P. Pastor, M. Kalakrishnan, J. Bohg, and S. Schaal, "Probabilistic object tracking using a range camera," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3195–3202.
- [53] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011, pp. 559–568.
- [54] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 23–30.
- [55] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-CMU-Berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.
- [56] B. Galerne, Y. Gousseau, and J.-M. Morel, "Random phase textures: Theory and synthesis," *IEEE Transactions on image processing* 2010, vol. 20, no. 1, pp. 257–267, 2011.
- [57] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1–4.
- [58] B. Wen, C. Mitash, S. Soorian, A. Kimmel, A. Sintov, and K. E. Bekris, "Robust, occlusion-aware pose estimation for objects grasped by adaptive hands," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6210–6217.
- [59] K. Tahara, S. Arimoto, and M. Yoshida, "Dynamic object manipulation using a virtual frame by a triple soft-fingered robotic hand," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 4322–4327.
- [60] I. A. Sucas, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.
- [61] R. Ma and A. Dollar, "Yale OpenHand Project: Optimizing Open-Source Hand Designs for Ease of Fabrication and Adoption," *IEEE Robotics Automation Magazine*, vol. 24, no. 1, pp. 32–40, 2017.
- [62] W. G. Bircher, A. M. Dollar, and N. Rojas, "A two-fingered robot gripper with large object reorientation range," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3453–3460.
- [63] J. Loncaric, "Normal forms of stiffness and compliance matrices," *IEEE Journal on Robotics and Automation*, vol. 3, no. 6, pp. 567–572, 1987.
- [64] S. Sajjan, M. Moore, M. Pan, G. Nagaraja, J. Lee, A. Zeng, and S. Song, "Clear Grasp: 3D Shape Estimation of Transparent Objects for Manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3634–3642.

System Disturbance Compensation

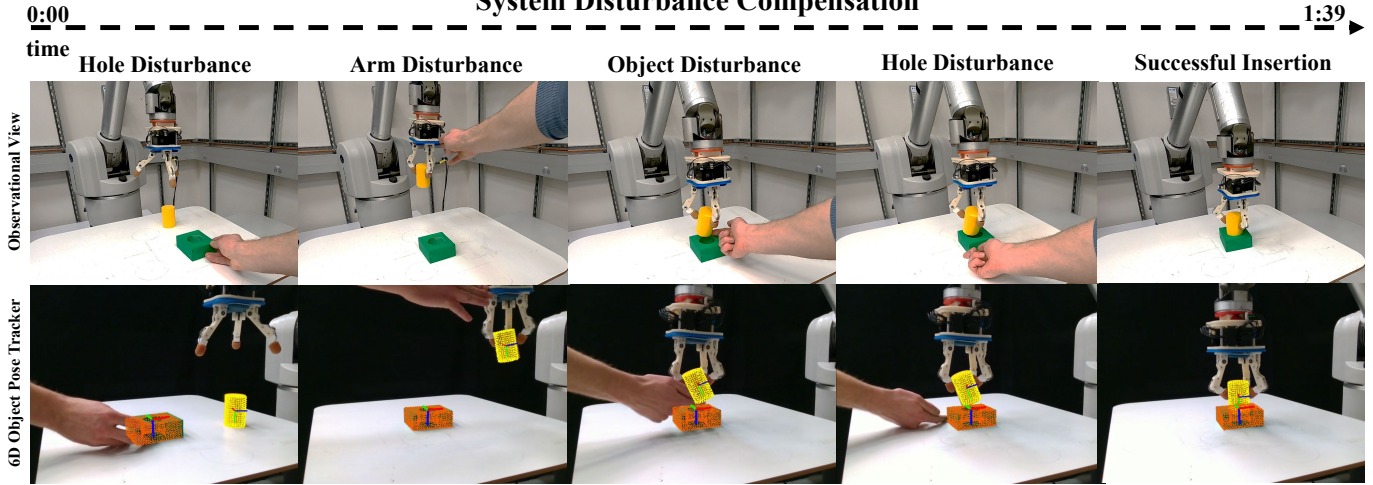


Fig. 8: The proposed system is robust to external disturbances imposed during task execution. In this task sequence, the pose of the hole, arm, and object are all manually perturbed and the system recovers such that the object successfully reaches its goal configuration. Refer to the supplementary video for the complete results.

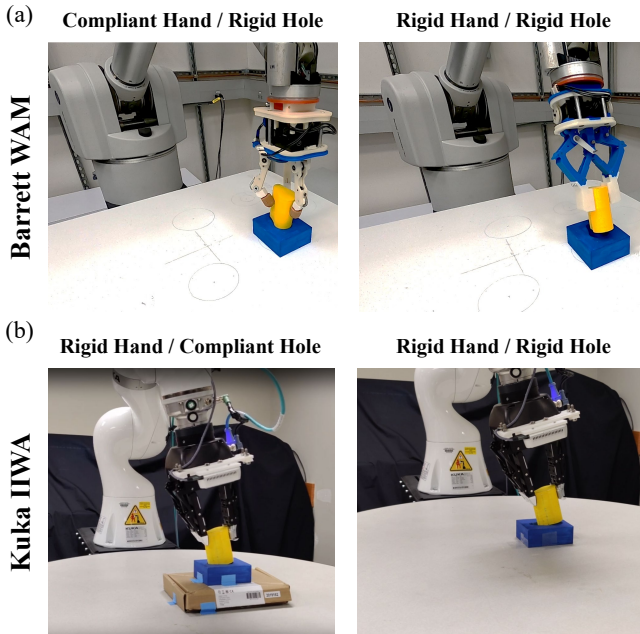


Fig. 9: The ablations make use of two manipulators with different levels of compliance at the arm, hand, or environment level. (a) A Barrett WAM serves as a low-impedance, compliant manipulator and has been tested with a compliant (left) and a rigid (right) hand; while the (b) Kuka IIWA is an example of a rigid manipulator, which has been tested with a rigid hand and a compliant (left) or rigid (right) hole.

APPENDIX

A. Selecting a Manipulation Frame

Selecting the best manipulation frame for object insertion, as eluded to in Sec. IV-A, is important as the antipodal grasp that defines \mathcal{M} , limits the amount of torque that can be applied by the contacts onto the object. Let’s consider two objects that define different antipodal contact widths, d_o^1 and d_o^2 , respectively. These distances fundamentally define the lever arm by which contact forces can be applied by the manipulator, or in the case of this work a hand, to the object. Assuming that

friction coefficients and contact force applications are identical, the grasp with $\max(d_o^1, d_o^2)$ will have a larger bound in the torques than can be imparted onto the object to aid in insertion.

In addition to the case of supplying a larger torque to the object, selecting the cross section of Γ that nearly maximizes d_o is also advantageous, according to the defined condition of insertion (Sec. IV-A). Fundamentally, while keeping the tolerance between the hole and all candidate object cross sections constant, as one increases d_o , β_f will subsequently decrease. By selecting d_o such that it is (near) maximized, this condition is attempting to control the most difficult axis of the object to insert while relying on compliance to account for any out-of-plane reorientation of the other axis, which has a larger β_f . While the proposed approach for picking \mathcal{M} is largely heuristic, it suits well for insertion with a dexterous hand that can apply much smaller forces than that of a manipulator.

More formally, in the case of selecting $\mathcal{M} \in SE(3)$, we want to choose an insertion plane inside of an arbitrary 3D object geometry, Γ , such that we maximize the projection of T back onto the vector connecting the two antipodal contact points. Assuming a given Γ , we can compute T and thus \mathcal{M} by analyzing properties of the object’s face. More formally, consider that the outer geometry of a peg’s face is sufficiently represented by a 2D point cloud of n points, i.e. $F \in \mathbb{R}^{n \times 2}$. It is possible to calculate the centroid of the point cloud, F_c , by taking its dimension-wise mean. Using Principal Component Analysis (PCA), we can compute the two principal axes of F , forming $P = [\pi_1, \pi_2]$. We solve for the outermost point, m , of the point cloud geometry by solving the optimization problem,

$$m = \arg \max_{f \in F} \pi_1 \cdot (f - F_c) \quad (9)$$

The location of m , which lies, or almost lies, along π_1 , appropriately maximizes the distance, d_o , between two antipodal contacts on the 3D object geometry. We can then compute the transformation T from m back to the object frame, $\mathcal{X} \in SE(3)$, and thus providing our manipulation frame $\mathcal{M} \in SE(3)$ that will be used for guiding object insertion.

B. System Recovery to Disturbances

For robots that act in unknown, and contact-rich environment, various forms of disturbances can arise depending on the robot's operating scenario. As aforementioned, these disturbances can be self-induced, where for instance, a robot's perception and control noise causes the manipulated object to move differently than the internal model predicted, and thus the system must react. Similarly, disturbances can occur that are not caused directly by the robot's actions but by other objects or robots in its environment, as in the case of highly cluttered scenarios. We simulate such occurrences by deliberately disturbing the arm, object, and hole during an execution (Fig. 8). More specifically, during the task and in four separate occasions, we move the location of the hole on the support surface, causing our system to reactively adapt to the new goal configuration via the continuous feedback from the object tracker. During this sequence, we also push on the arm and the object, perturbing the state of the hand-object configuration, as to require the system to overcome such disturbances for successful insertion. These task evaluations are included in this paper's supplementary video.

C. System Observations and Limitations

This subsection is part of a discussion on system limitations (Sec. VI), and what can be improved about the system for future advancements in robot assembly tasks.

Visual Perception: While the RGBD-based 6D pose tracker is robust to a variety of objects with challenging properties such as textureless, reflective, geometrical featureless, or thin/small shapes, it struggles to track severely shiny, glossy, or transparent objects, due to the degenerated depth sensing of the camera. In future work, we hope to explore extending this framework to these other types of scenarios with the techniques of depth enhancement and completion [64]. In addition, the current framework requires an object CAD model beforehand to perform 6D pose tracking and for reasoning about the task of peg-in-hole insertion. In future work, reconstructing the model of novel objects on-the-fly [49] while with sufficient precision to perform high-tolerance tight insertion tasks is of interest.

Inaccuracy of the Low-Impedance Manipulator for Grasp Acquisition: While the manipulator leveraged in this work was largely beneficial for task completion, it also introduced difficulties in acquiring an initial grasp. Soft, compliant, and underactuated hands are well suited for grasping, but in cases of robot assembly where future within-hand manipulation is necessary (especially with 40+ finger actions as we saw with our tasks), acquiring a well-intended and stable grasp at the fingertips is necessary from the start. Such grasps would be especially possible with an accurate manipulator that is able to appropriately position the hand over the object for grasping. Though, for cases in this work where we saw task failure, it was largely due to starting with a poor initial fingertip grasp, which was directly attributed to the end effector's deviation from its commanded initial pose.

ACKNOWLEDGEMENTS

The authors would like to thank V. Patel for his assistance in designing object-forming fingertips for the GR2 hand, in addition to the layered two-tone object geometries.