Within-hand Manipulation Planning and Control Approaches for Variable Friction Fingers

Gokul Narayanan[1], Joshua Amrith Raj[1], Abhinav Gandhi[1], Aditya A. Gupte[1], Adam J. Spiers[2] and Berk Calli[1]

***Abstract*— The ability to conduct within-hand manipulation provides significant dexterity and flexibility advantages, which would greatly benefit robots operating in unstructured and dynamic environments. Yet, robotic within-hand manipulation is challenging to implement, even for highly articulated, sensorized and expensive robotic hands, mainly due to the lack of accurate hand-object and contact models. In our recent prior work, we developed a novel 2-DOF robot gripper that can change the effective friction of its finger surfaces with a simple actuated mechanism of interlaced materials. This allowed controlled sliding and rotation actions of objects. In this paper, we present automatic object-pose planning and control schemes to accompany this unique and open-source manipulation platform. Here, we propose three strategies to re-position and reorient an object within the workspace of the hand: 1) an offline motion planner that is utilized in a feedforward manner, 2) an online vision-based control algorithm, 3) a hybrid approach that combines the offline and online algorithms. These methods are designed considering the non-holonomic and switching nature of the system, while achieving path smoothness, high accuracy and efficiency. Our experimental results on simulated and physical systems show that each method has their use cases, i.e. the offline planner is the fastest, the online planner is prompt and accurate, and the hybrid planner is efficient and leads to smooth trajectories. These algorithms will be released as an accompaniment to the VF hand hardware, thus providing to a complete open-source, low-cost platform for investigating within-hand manipulation.**

I. INTRODUCTION

Humans heavily rely on their Within-Hand Manipulation (WIHM) capabilities for performing daily activities [1]. These subconsciously conducted actions for re-orienting and re-positioning objects within the hand help people to streamline manipulation operations by changing the initial grasp pose into task-oriented hand-object configurations (e.g. re-positioning a pen, scissors or keys after picking-up to make the objects ready for use).

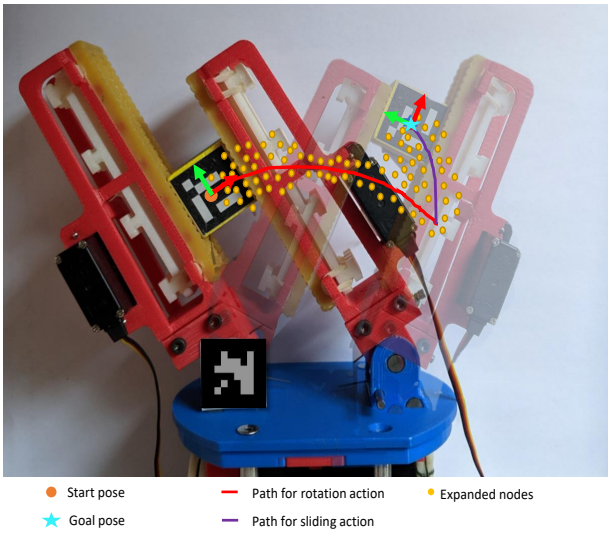

Fig. 1. Within-Hand Manipulation of a cubic object from an arbitrary start pose (position and orientation) to a desired end pose using an automatically determined sequence of sliding and rolling actions based on a weighted A* search algorithm with modified cost function. The 2-DOF hand platform utilizes variable friction (VF) finger surfaces.

Current robotics technology lacks similar dexterous manipulation capabilities, which are essential for the service robots that are expected to efficiently operate in environments engineered for humans. These capabilities can further extend the use of industrial robots to less structured environments by relaxing the constraints of their initial grasps and allowing them to re-position grasped objects within the gripper (without relying on re-grasping), even in confined spaces that restrict gross arm motions.

Successful WIHM warrants controlled translation, re-orientation and fine positioning of an object within the hand workspace. These actions are greatly challenging to achieve even with sensorized, high-DOF, expensive robotic hands operating in structured settings with detailed hand-object models [2]–[4].

Our research focuses on enabling robotic WIHM

[1]Authors are with Robotics Engineering Program, Worcester Polytechnic Institute, 85 Prescott Street, Worcester, MA-01605, USA {gsathyanarayanan;jcalebchanthiraj;agandhi2;aagupte;bcalli}@wpi.edu

[2]Author is with the Dept. Haptic Intelligence, Max Planck Institute for Intelligent System, Stuttgart, 70372, Germany a.spiers@is.mpg.de

by facilitating essential WIHM actions of sliding and rotating of objects along a robot's fingers with practical and effective combinations of mechanical design, control strategies and minimal system models. In our previous work, we have proposed a simple 2-finger, 2-DOF (1-DOF per finger) robot gripper with variable friction (VF) surfaces [5]. The actuated contact surfaces of each finger (which consist of interlaced high and low friction materials) allows us to easily switch between sliding and pivoting motions of the object. To the best of our knowledge, the gripper is unique for being able to achieve large sliding and reorientation actions, allowing operations such as 180 degree object rotations within-hand, and sliding-gating the object from the center of the grasp to the tip of the fingers. However, these capabilities were only previously demonstrated with manually determined, pre-programmed actions. While the design provides various mechanical advantages, it is a non-holonomic, switching system with uneven action outcomes, all of which pose challenges for the development of an automatic motion planning scheme.

In this work, we investigate strategies for motion planning and control of WIHM of a grasped object using the setup proposed in [5]. Several algorithms have been proposed to accurately perform accurate object pose control via sliding and re-orientation (rolling) using the VF hand, via finger motion and automatic friction-mode switching. These options are as follows 1) An offline motion planner, 2) an online vision-based control method and 3) a hybrid approach that hierarchically combines the offline planner and the vision-based controller. The algorithms that utilize vision-based control (2 and 3) demonstrate robustness to gripper and object model uncertainties, in addition to such phenomenon as undesired slippage between the object and robot finger surfaces.

This paper is organized as follows: Section II gives a brief review of related work. The system description and kinematics model is presented in Section III. The motion planner and visual servoing approach are discussed in Sections IV, V respectively. Section V-B explains the hybrid approach and the experimental results are showcased in section VI.

II. Related Work

Traditional WIHM methods such as those in [6], [7] are based on the modelling of friction forces between the object and robotic hand, along with kinematic analysis of the grasp and suggested control strategies. In [2], [8]–[10] anthropomorphic designs are presented for WIHM of objects. Non-anthropomorphic manipulator designs

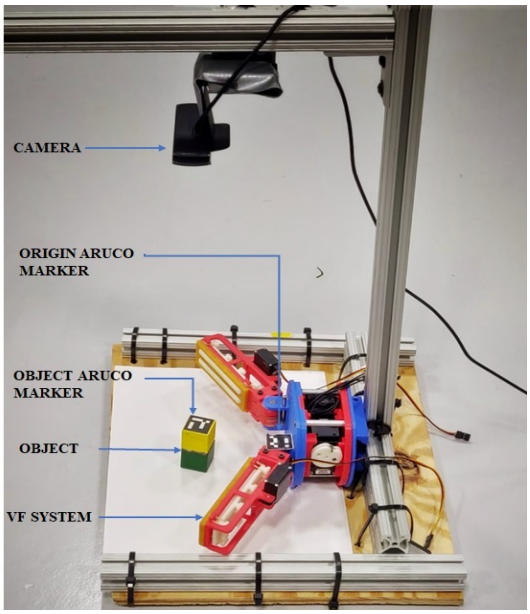

Fig. 2. The experimental setup used in this work.

showing a variety of within hand manipulation capabilities have also been proposed in [11], [12], but the VF finger design offers an open source and inexpensive alternative capable of WIHM. Methods including rolling [13] as well as sliding the objects [14] to dexterously manipulate objects have been tested and we explore the same for the VF finger design.

Multiple works like [5], [15] use learning based methods for WIHM but due to ease of kinematic modelling of the mechanism used for manipulation the motion planning schemes presented in the paper need not be learning based. In [16] the authors study the capabilities of a simple gripper to dexterously manipulate the object using external forces and several other methods used for in hand manipulation are summarized in [17], but none of them use the unique VF hardware setup proposed in [5].

Heuristic based motion planning algorithms that pertain not specifically to WIHM but are applicable to non-holonomic motion are compiled in [18].The feasibility of applying heuristic based search to high dimensional state space problem like manipulation by modifying heuristics and using motion primitives is explained in [19] . The proposed motion planner uses modified heuristics similar to [19]to obtain smoother paths efficiently.

Combination of tactile and visual feedback forms the basis of the control framework proposed in [20] for dexterous manipulation. The papers [21], [22] use vision based feedback to make the WIHM control strategies more robust to inaccuracies in modelling. The fundamental techniques used in [20]–[22] are explained in

[23]. We are using an image based visual servoing (IBVS) scheme as given in [23] with help of ArUco markers.

III. System Description and Model

In this section, the design and functionalities of the VF gripper are summarized. Then, a kinematic model of the system is presented, which will later be utilized in motion planing and vision-based control schemes in Sections IV and V.

A. Description of the VF Gripper

Friction plays an important role in WIHM. Effective frictional forces between the finger and object surfaces can change the outcome of a particular finger motion on the object. The VF gripper is designed to allow modification of the effective surface friction coefficient at the contacts on demand. The VF finger hardware is open-source as part of the Yale OpenHand project. CAD files and assembly instructions may be downloaded from *https://www.eng.yale.edu/grablab/openhand*.

Each finger's body and low friction surface were 3D-printed in ABS (using a Stratsys Fortus 3D printer), whereas the high friction surfaces were molded using urethane rubber (Vytafelx 30, manufactered by Smooth-On). The ABS inserts offer lower friction and can be retracted using servo motors to expose the high friction urethane rubber. This capability of changing friction allows switching between sliding and rolling of the object.

By setting the $\{$*left finger*, *right finger*$\}$ friction states appropriately, we are able to achieve different WIHM behaviours. As such, moving the fingers during the following friction conditions 1) $\{high, low\}$, 2) $\{low, high\}$ and 3) $\{high, high\}$ leads to 1) sliding of the object along the right finger, 2) sliding of the object along the left finger and 3) rotation of the object.

Dynamixel Model-X actuators are used to control the motion of the fingers while miniature hobby servos (HiTec HS-85MG) toggle the variable friction surfaces between high and low conditions. As previously described in [5], at any moment, one of the Dynamixels is in velocity mode to lead the motion, while the other Dynamixel is in torque mode, to maintain finger contact with the object. The role of each Dynamixel actuator changes depending on the friction state of the gripper and direction of object motion, as also detailed in [5].

B. Kinematics Model

In this section we present a kinematics model of our hand-object system, which expands on the model presented in [5] to give sufficient detail for closed loop object pose control. The kinematics provides a mapping between actuator space, finger space and Cartesian space. Note that this kinematic model assumes a square object, which is the focus of the planning algorithm in this work, however the model may be easily expanded to other object shapes.This system is non-holonomic, because the motion of the object depends on the friction states of the finger. We therefore need a kinematic model for each configuration.

The model has four variables, d_L, d_R, θ_L and θ_R, as illustrated in Fig. 3. Here, d_L is the distance from the left finger joint to the object's bottom corner and d_R is the distance from the right finger joint to the object's bottom corner. θ_L and θ_R are the angles of the fingers with respect to the positive X-axis in the gripper frame.

When the object slides, one of its surfaces maintains contact with the high-friction surface while the opposite surface slides against the low-friction finger. This constraint, of the object being effectively stationary against one finger, is used to find the position of the center of the object in terms of the state variables. Only one finger is responsible for controlling the position of the object due to the torque-control contact-maintaining function on the other finger (as was outlined in Section III-A). The hand-object system forms a closed kinematic chain that bridges both fingers with the object. Therefore, the object position can be expressed in terms of any one of the actuator angles. We use $[d_L, \theta_L]$ or $[d_R, \theta_R]$ to formulate the object position depending on which finger the object is sliding against. The position of the center of the object while it slides against the right finger is given as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} (d_L + w_o/2)cos(\theta_L) + (w_o/2 + f_w)sin(\theta_L) \\ (d_L + w_o/2)sin(\theta_L) - (w_o/2 + f_w)cos(\theta_L) \end{bmatrix} \tag{1}$$

Similarly the equations for sliding on left finger is given in equation 2.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} w_p + (d_R + w_o/2)cos(\theta_R) - (w_o/2 + f_w)sin(\theta_R) \\ (d_R + w_o/2)sin(\theta_R) + (w_o/2 + f_w)cos(\theta_R) \end{bmatrix} \tag{2}$$

Here, w_p is the palm width (the distance between the finger joints), w_o is the width of the object and f_w is the finger width, which is the distance between the center line of a finger to the fingerpad, as illustrated in Fig. 3. While the object is sliding along the right finger, the position of the object in the Cartesian space is dependent on d_L and θ_L. From the two equations in the X and Y directions, we can find d_L, θ_L from the objects position. The relation between the state variables d_L, d_R, θ_L and

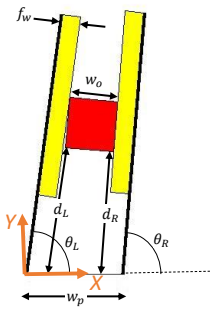

Fig. 3. Parameters of the kinematic model for the hand-object system (for a square object of width w_o).

θ_R is already derived in [5]. It must be noted that in (1), the actuator angle, θ_L alone would vary when the object slides on the right finger, whereas d_L remains constant throughout the sliding operation.

Equation (1) is differentiated with respect to time to get the velocity of the object in terms of the velocity of the actuator. The velocity of the object is related to the finger angles with a Jacobian matrix, $v = J\dot{\theta}$. We thus determine the Jacobian, which we will use later for visual servoing. The velocity of the object when it is sliding on the right finger can be written as follows:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}-(d_L + w_o/2)sin(\theta_L) + (w_o/2 + f_w)cos(\theta_L)\\ (d_L + w_o/2)sin(\theta_L) + (w_o/2 + f_w)sin(\theta_L)\end{bmatrix}\dot{\theta_L} \tag{3}$$

The velocity of the object sliding in left finger can be obtained by differentiating equation 2 with respect to time.

IV. Motion Planning

The purpose of the WIHM motion planning is to find the required finger actions that will lead the object to any given feasible within-hand goal pose from an arbitrary initial grasp pose. In this section, we determine approaches to plan the motion of the hand and object, within the grasp of the VF hand.

The WIHM motion planning task can be formulated as a graph search problem of finding the shortest path between nodes. Here, the nodes of the graph are the poses of the object and edges are the actions of the VF system. This graph search problem can be solved using various techniques [24]. We are using the A* algorithm with modifications in the cost function to get a smoother and faster solution. The details of the search algorithm are discussed in the following sections.

A. State Space, Action Space and Cost Functions

For the WIHM planning problem, the state variables are expressed in finger space, as this provides an intuitive way to formulate cost functions for the actions. An alternative way would be using Cartesian space, which has a one-to-one mapping to the finger space for the feasible object poses, and a very similar formulation would follow. The state variables are $s = (d_L, d_R, \alpha)$, where α is the object orientation and d_L, d_R are defined in Section III-B. We discretize d_L and d_R by 1 mm displacements, which we think as a reasonable position resolution for this system. The discretization of α depends on the object geometry.

TABLE I

State Action Transition Table, for Moving the Object from State s to State s'

Action	Next State s'
LEFT SLIDE UP	$(d_L + \gamma_{slide}, d_R, \alpha)$
LEFT SLIDE DOWN	$(d_L - \gamma_{slide}, d_R, \alpha)$
RIGHT SLIDE UP	$(d_L, d_R + \gamma_{slide}, \alpha)$
RIGHT SLIDE DOWN	$(d_L, d_R - \gamma_{slide}, \alpha)$
ROTATE CLOCKWISE	$(d_L + l, d_R - l, \alpha - \gamma_{rotate})$
ROTATE ANTI-CLOCKWISE	$(d_L + l, d_R - l, \alpha + \gamma_{rotate})$

The action space A comprises of a set of sliding and rotation actions as follows:

$$\begin{aligned} A = \{&LEFT\_SLIDE\_UP,\\ &LEFT\_SLIDE\_DOWN,\\ &RIGHT\_SLIDE\_UP,\\ &RIGHT\_SLIDE\_DOWN,\\ &ROTATE\_CLOCKWISE,\\ &ROTATE\_ANTICLOCKWISE\} \end{aligned} \tag{4}$$

In this notation, $LEFT\_SLIDE\_UP$ means sliding the object along the left finger in the positive (distal) direction, and so on. The magnitude of the sliding and rotating actions can be varied with the parameter γ_{slide} and γ_{rotate} respectively.

In our notation, we denote the current state with s and the immediate next state after an action is taken as s'. The state action transitions from s to s' for various actions is expressed in the Table I. Since a rotation action is achieved by pivoting the object on its corner though 90^o, the object also gets translated by one body length, l.

The cost function $c(s, s')$ gives the cost of performing the state transition from state s to s'.

$$c_{sliding}(s, s') = \gamma_{slide} \tag{5}$$

$$c_{rotation}(s, s') = 2 * l + \gamma_{rotate} \tag{6}$$

The heuristic cost $h(s)$ is the estimated cost to move from the current state to the goal state:

$$h(s) = h_{position}(s) + h_{orientation}(s) \tag{7}$$

where $h_{position}(s)$ is the position correction cost, which is equal to the Manhattan distance between the current position and goal position. $h_{orientation}(s)$ is the orientation correction cost, which is equal to the orientation difference between the current pose and the goal pose in absolute values.

B. A* Search Algorithm

The conventional A* [18] algorithm finds the shortest path from the start state to the goal state by expanding nodes in the order of increasing cost. The cost function is given as

$$f(s') = g(s') + h(s') \tag{8}$$

$$g(s') = g(s) + c(s, s') \tag{9}$$

where $f(s')$ is the estimated cost from the start state to goal state, through s'. $g(s)$ is the accumulated cost of all traversed states from the start state to s. $g(s')$ is the cost of the state s' from start state to s'. $h(s')$ is the estimated cost to reach goal state from s'. Applying the conventional A* algorithm to our problem generates a sequence of actions connecting the start state to the goal state with the lowest possible cost.

A path generated by the A* planner for the initial pose $(X, Y, \alpha) = (7.0, 7.5, -90)$ and goal pose $(12, 12, 0)$ is shown in the Fig. 4a together with the expanded nodes for the search. It is important to note that, for the given initial position, it is not possible to directly perform the rotation action due to workspace limitations on finger angles. Nevertheless, the path planner successfully leads the system to a configuration where the rotation becomes possible, and then performs that rotation. After the rotation, the object is slid on the fingers with consecutive sliding actions until it reaches the goal pose.

The sliding actions after the rotation includes high frequency chattering. The reason for this non-smooth behavior is that the sliding actions need to switch between the left and right fingers frequently in order to reach the goal position in the shortest possible way. This is due to the non-holonomic nature of the VF finger system. Such chattering is highly undesired for our system, since it requires frequent switching between finger modes (i.e. $\{high\_friction, low\_friction\}$ to $\{low\_friction, high\_friction\}$ and vice versa), which does not only take time, but also causes wear and tear of the hardware and energy inefficiency. To solve this problem, we are unable to utilize conventional trajectory smoothing techniques [25], which are based on polynomial interpolation and curve fitting, as our system performs discrete rotations (in orders of 90^o) that are difficult to model in those frameworks. Therefore, we propose the following modifications in the heuristic function.

C. Modified Heuristic Cost Function

Generally, a node in the A* search expansion tree comprises of the following information: state s, action a, cost $f(s)$, and a pointer to the parent node P. The parent node is the node from which the current node is expanded, and the parent node action is given by P_a. Smooth paths can be achieved by relaxing the optimality constraint, and directing the search algorithm to balance between the path smoothness and length. This is achieved by assigning low costs to nodes which are expanded by the same action as their parent node. We define the heuristic cost $h(S)$ of a node as follows:

$$h(S) = \beta * h(S) \tag{10}$$

where

$$\beta = \begin{cases} 0 < \beta < 1, & \text{if } a == P_a \\ 1, & \text{otherwise} \end{cases} \tag{11}$$

Since the A* search algorithm expands the nodes with lowest cost first, the modified heuristic cost function will make the algorithm choose a smoother trajectory by favoring the parent node action, sometimes at the expense of the shortest path. The balance between the path length and trajectory smoothness is determined by the parameter β, with a lower β meaning a smoother path. This heuristic function is applied to the same initial and goal positions as previously defined in Section IV-B. Results are given for $\beta = 0.7$ in Fig.4b. It can be observed that the path is smooth, but the number of nodes expanded are more than the conventional A* search in Fig. 4a which makes the search algorithm significantly slower.

D. Weighted A* Search

In order to speed up the search algorithm, the modified heuristic cost $h(s)$ is multiplied with a constant ϵ and then added to $g(s)$ to calculate the final cost as shown in (12). This results in a weighted A* search [26].

$$f(S) = g(s) + \epsilon * h(S) \quad where\ \epsilon > 1 \tag{12}$$

This creates a bias towards the goal state by expanding more nodes, which are closer to the goal to speed up the search. However, again, the solution doesn't guarantee an optimal path, whereas it guarantees a ϵ sub-optimal path. Fig. 4c shows the path from the weighed A* without modification in the cost function. Fig. 4d shows the path from the weighed A* with modified cost function, and the solution is smoother and faster than the conventional A* approach shown in Fig. 4a. The parameters ϵ and β can be tuned as per requirements.

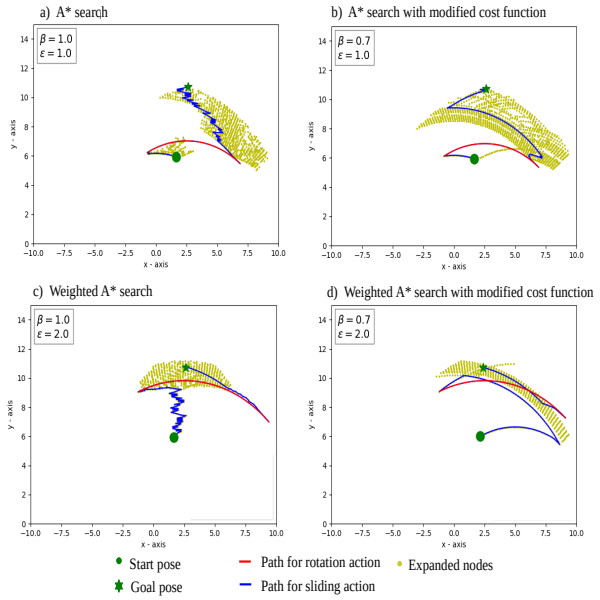

Fig. 4. Planned path using motion planner from start state (X, Y, α = 7.0, 7.5, -90) to goal state (12, 12, 0) (a) Conventional A* search (not smooth). (b) A* search with modified cost function (smoother) (d) Weighed A* search (not smooth but fast solution). (d) Weighed A* search with modified cost function (smooth and fast solution). Axis units are cm.

V. Introducing Visual Feedback for WIHM

In our framework, we assume an eye-to-hand system, i.e. a camera, is observing the hand-object system. We propose two vision-based control schemes: a pure vision-based control, and a hybrid approach that utilize both the offline plan and the online feedback.

A. Pure Vision-based Approach

We use a classical Image-based Visual Servoing scheme [27], which tracks the features on the object, and generates velocity references that minimize the error between the current and desired feature locations as follows

$$v_{ref} = -\lambda {J_i}^{\dagger} e \tag{13}$$

where ${J_i}^{\dagger}$ is the pseudo-inverse of the image Jacobian, e is the feature error vector, and v_{ref} is the Cartesian velocity reference for the object. We cannot directly apply this velocity reference to our system due to two reasons. The first reason is that our system is non-holonomic, and cannot directly follow the linear velocities generated by the visual servoing algorithm. The second reason is that our system can only perform discrete rotations in the order of 90^o, and therefore cannot follow continuous rotational velocities. Instead, we decouple the position error and orientation error and apply the following procedures:

1) Position correction step: We correct the positioning errors by sliding the object either along the left finger or along the right finger. As explained in Section III-B, there are separate Jacobians for each of these cases. In each time step, we calculate both of these Jacobians, calculate associated object velocity vectors, and choose the one that is closest to v_{ref}.

2) Orientation correction step: A complete $90°$ rotation of the object cannot be initiated from all points in the workspace, since this action requires large object displacements that may lead the object outside the workspace. The motion planner in Section IV finds a path that brings the object to a location, where rotation is feasible (e.g. Fig. 4), but visual servoing algorithm is essentially a local optimizer, and lacks the mechanisms to do so. In order to overcome this disadvantage, when an orientation error exists in the finger space, we first move the system towards an extreme position (right most location for clockwise rotation of the object and left most location for anti-clockwise rotation of the object) and then execute the rotation action. Once the rotation is corrected, we continue executing position correction steps. The object trajectories obtained with this strategy is far from ideal, but ensure the object to stay in the operable workspace. Nevertheless, combining the offline planner and the visual servoing algorithm provides an alternative solution as follows.

B. Hybrid Approach

Executing the path of the offline motion planner directly on the real hardware results in large inaccuracies (as shown in the Fig. 5-b) due to the inaccuracies of the hand-object model and undesired/uncontrolled slip happening during the process. While online vision-based method does not have these advantages, using it in a standalone manner may not be preferable either since 1) we need to utilize some rules to manage the discrete rotation motions, which results inaccurate trajectories, 2) high frequency chattering happen during convergence (Fig. 5-a). Therefore, we propose a hybrid controller, which sends intermediate goal poses based on the offline plan generated by the motion planner in a feedforward manner, and uses the visual servoing controller in the feedback loop to correct for the position errors while reaching these intermediate goals. Such an approach results in both smooth and accurate WIHM Fig. 5-c. In the next section, we analyze the performance of motion planner, pure visual servoing and hybrid approach with test cases.

VI. Experimental Results

We conducted experiments with the VF system and compared the performance of the three approaches for conducting the within-hand manipulation task: 1) offline motion planning, pure visual servoing, and the hybrid method. The experimental setup is constructed as shown in the Fig.2. The camera was set to 640x480 pixels resolution and 30 f/s capture rate. The distance of the camera to the object top surface was 20 cm. The size of the cube is 2.5 cms. Each method was tested on three different tasks:

1) Perform only position correction.
2) Perform both position and orientation correction.
3) Perform position and orientation correction with modelling inaccuracy.

For each task, 5 randomly chosen sets of feasible start and goal states are generated. These set points are applied to each method, and mean and variance of the performance metrics are presented in Table II. The parameters for the motion planner described in Section IV are set as $\epsilon = 2$ and $\beta = 0.1$ to get a smooth solution. The evaluation of the methods were based on four criteria: accuracy, path length, time taken, and the number of switching actions. Here, the path length is the total distance travelled by the object in the Cartesian space. The measure of smoothness is the the number of times the system changes its friction configuration.

The results of Task 1 is presented in Table II-a. It can be seen that, while offline motion planner gives the smoothest path (thanks to the β parameter) and the fastest execution time (due to the feedforward execution), it is significantly less accurate than the vision-based methods. Visual servoing method gives the most accurate results, but the least efficient method in time consumption and path length. This is due to large amount of switching caused by trying to follow a straight path with the non-holonomic system. However, it is important to note that the visual servoing algorithm does not require any offline planning, which can be an advantage in some scenarios.

The results of Task 2 is given in Table II-b. Here we observe a large inefficiency in visual servoing algorithm caused by its orientation correction strategy. The hybrid method becomes the most accurate one, since it can handle both position and orientation correction precisely by combining the advantages of the offline and online methods.

For Task 3, the size of the cube is given to the planner as 2 cm instead of 2.5 cm. In Table II-c, we can see that using vision-based methods maintain the high accuracy even in the face of modelling errors.

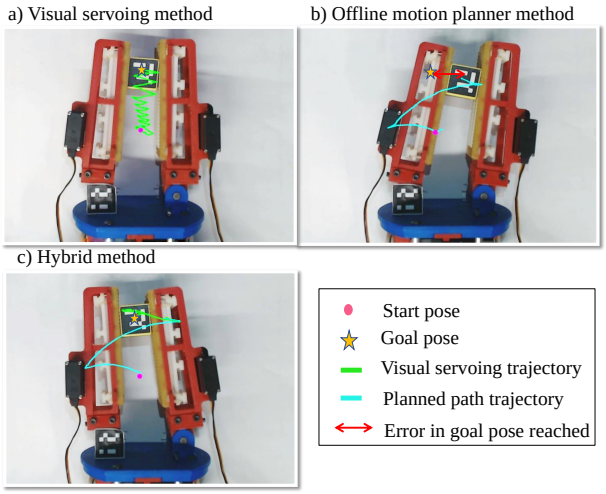

Fig. 5. Trajectory tracked by the object from start pose (7,7,0) to goal pose (12,12,0) for different methods a) Visual servoing b) Offline motion planning c) Hybrid method

Surprisingly, the offline motion planner also performs good in this task, even though modeling inaccuracy is expected to drop its performance further. We believe that this results is due to hand and object modeling inaccuracies coincidentally cancelling out each other, resulting in a more accurate overall model. Nevertheless, such object model inaccuracies are expected to drop the offline method's performance in general.

VII. Conclusion

In the present work, we showcase different frameworks to perform in-hand manipulation of objects using the VF finger gripper system. The frameworks were tested with modelling inaccuracies and with uncertainties in the motion execution. The experimental results show that the hybrid framework performs faster and gives smoother paths as it combines the advantage of a pre-defined path from the motion planner with online correction from visual servoing. This proves that even with inaccurate models of the system it is possible to perform in-hand manipulation of objects with our approach.

In our future work, we propose to develop and integrate an online motion planner along with visual servoing in the feedback correction loop to address the uncertainties in the environment more efficiently. Additionally, we plan to develop algorithms, by incorporating reinforcement learning methods, that will learn policies for WIHM. Thus making the VF gripper system more robust to uncertainties and allowing us to extend this approach to objects of different shapes and sizes.

To co-incide with the publication of this manuscript, we shall release a ROS library of the differ-

TABLE II

PERFORMANCE METRICS FOR DIFFERENT TASKS

a. TASK1: POSITION CORRECTION

	Accuracy (cm)	Time taken (s)	Path length (cm)	Switching actions
Visual servoing	0.14 ± 0.1	46 ± 22	38 ± 29	11 ± 3
Offline Motion planning	3.1 ± 2.3	10 ± 1	13 ± 3	2 ± 0
Hybrid	0.36 ± 0.2	27 ± 15	20 ± 3	6 ± 2

b. TASK2: POSITION AND ORIENTATION CORRECTION

	Accuracy (cm)	Time taken (s)	Path length (cm)	Switching actions
Visual servoing	0.35 ± 0.1	76 ± 50	90 ± 56	16 ± 11
Offline Motion planning	2 ± 1.8	28 ± 3	33 ± 5	4 ± 0
Hybrid	0.2 ± 0.1	54 ± 10	57 ± 9	12 ± 4

c. POSITION AND ORIENTATION CORRECTION WITH MODELLING INACCURACIES

	Accuracy (cm)	Time taken (s)	Path length (cm)	Switching actions
Visual servoing	0.2 ± 0.02	92.7 ± 14.74	82 ± 145	19 ± 8
Offline Motion planning	1.0 ± 0.8	20 ± 2.65	32 ± 5	5 ± 2
Hybrid	0.3 ± 0.2	37.3 ± 15.4	35 ± 10	7 ± 2

ent algorithms described in this work, along with necessary functions for integration with the open-source and easy-to-assemble VF hand hardware (CAD files and instructions may be found at *https://www.eng.yale.edu/grablab/openhand*).

REFERENCES

[1] I. M. Bullock and A. M. Dollar, "Classifying human manipulation behavior," in *IEEE Int Conf on Rehabilitation Robotics*, 2011, pp. 1–6.

[2] J. Ueda, M. Kondo, and T. Ogasawara, "The multifingered naist hand system for robot in-hand manipulation," *Mechanism and Machine Theory*, vol. 45, no. 2, pp. 224–238, 2010.

[3] N. Y. Chong, D. Choi, and I. H. Suh, "A generalized motion/force planning strategy for multifingered hands using both rolling and sliding contacts," in *Proceedings of IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, 1993, pp. 113–120.

[4] Y. Karayiannidis, K. Pauwels, C. Smith, D. Kragic *et al.*, "In-hand manipulation using gravity and controlled slip," in *IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, 2015, pp. 5636–5641.

[5] A. J. Spiers, B. Calli, and A. M. Dollar, "Variable-friction finger surfaces to enable within-hand manipulation via gripping and sliding," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4116–4123, 2018.

[6] J. K. Salisbury and M. Mason, *Robot hands and the mechanics of manipulation*. MIT press Cambridge, MA, 1985.

[7] R. M. Murray, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[8] S. C. Jacobsen, J. E. Wood, D. Knutti, and K. B. Biggers, "The utah/mit dextrous hand: Work in progress," *The Int Journal of Robotics Research*, vol. 3, no. 4, pp. 21–50, 1984.

[9] L. Bologni, S. Caselli, and C. Melchiorri, *Design Issues for the UB Robotic hand*. Univ., School of Engineering, 1988.

[10] F. Lotti, P. Tiezzi, G. Vassura, L. Biagiotti, and C. Melchiorri, "Ubh 3: an anthropomorphic hand with simplified endo-skeletal structure and soft continuous fingerpads," in *IEEE Int Conf on Robotics and Automation (ICRA), 2004. Proceedings*, vol. 5, pp. 4736–4741.

[11] J. K. Salisbury and J. J. Craig, "Articulated hands: Force control and kinematic issues," *The Int journal of Robotics research*, vol. 1, no. 1, pp. 4–17, 1982.

[12] D. Osswald and H. Wörn, "Mechanical system and control system of a dexterous robot hand," in *Proceedings of the IEEE-RAS Int Conf on Humanoid Robots*. Citeseer, 2001, p. 8.

[13] A. Bicchi and R. Sorrentino, "Dexterous manipulation through rolling," in *IEEE Int Conf on Robotics and Automation*, vol. 1, 1995, pp. 452–457.

[14] J. C. Trinkle and R. P. Paul, "Planning for dexterous manipulation with sliding contacts," *The Int Journal of Robotics Research*, vol. 9, no. 3, pp. 24–48, 1990.

[15] H. Van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *IEEE-RAS 15th Int Conf on Humanoid Robots (Humanoids)*, 2015, pp. 121–127.

[16] N. C. Dafle, A. Rodriguez, R. Paolini, B. Tang, S. S. Srinivasa, M. Erdmann, M. T. Mason, I. Lundberg, H. Staab, and T. Fuhlbrigge, "Extrinsic dexterity: In-hand manipulation with external forces," in *IEEE Int Conf on Robotics and Automation (ICRA)*, 2014, pp. 1578–1585.

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

[18] D. Ferguson, M. Likhachev, and A. Stentz, "A guide to heuristic-based path planning," in *Proceedings of the Int workshop on planning under uncertainty for autonomous systems, Int Conf on automated planning and scheduling (ICAPS)*, 2005, pp. 9–18.

[19] B. J. Cohen, S. Chitta, and M. Likhachev, "Search-based planning for manipulation with motion primitives," in *IEEE Int Conf on Robotics and Automation (ICRA)*, 2010, pp. 2902–2908.

[20] C. Jara, J. Pomares, F. Candelas, and F. Torres, "Control framework for dexterous manipulation using dynamic visual servoing and tactile sensors' feedback," *Sensors*, vol. 14, no. 1, pp. 1787–1804, 2014.

[21] B. Calli and A. M. Dollar, "Vision-based model predictive control for within-hand precision manipulation with underactuated grippers," in *IEEE Int Conf on Robotics and Automation (ICRA)*, 2017, pp. 2839–2845.

[22] ——, "Vision-based precision manipulation with underactuated hands: Simple and effective solutions for dexterity," in *IEEE/RSJ Int Conf on Intelligent Robots and Systems (IROS)*, 2016, pp. 1012–1018.

[23] S. Hutchinson and F. Chaumette, "Visual servo control, part i: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[24] A. Madkour, W. G. Aref, F. U. Rehman, M. A. Rahman, and S. M. Basalamah, "A survey of shortest-path algorithms," *ArXiv*, vol. abs/1705.02044, 2017.

[25] A. Ravankar, A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, p. 3170, 2018.

[26] E. A. Hansen and R. Zhou, "Anytime heuristic search," *J. Artif. Int. Res.*, vol. 28, no. 1, pp. 267–297, Mar. 2007.

[27] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *IEEE Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.