

# Accounting for Part Pose Estimation Uncertainties during Trajectory Generation for Part Pick-Up Using Mobile Manipulators

Shantanu Thakar<sup>1</sup> Pradeep Rajendran<sup>1</sup> Vivek Annem<sup>1</sup> Ariyan Kabir<sup>1</sup> Satyandra Gupta<sup>1</sup>

**Abstract**—To minimize the operation time, mobile manipulators need to pick-up parts while the mobile base and the gripper are moving. The gripper speed needs to be selected to ensure that the pick-up operation does not fail due to uncertainties in part pose estimation. This, in turn, affects the mobile base trajectory. This paper presents an active learning based approach to construct a meta-model to estimate the probability of successful part pick-up for a given level of uncertainty in the part pose estimate. Using this model, we present an optimization-based framework to generate time-optimal trajectories that satisfy the given level of success probability threshold for picking-up the part.

## I. INTRODUCTION

Mobile manipulators can be used for pick-and-transport operations for parts. Traditionally, this is done by positioning the mobile base near the part, then moving the manipulator to grasp the part for the pick-up operation. However, this takes a longer time as compared to grasping parts while the mobile base and the gripper are in motion. This behavior is routinely demonstrated by humans who can pick up objects with their arms while walking or running. Therefore, we are interested in moving the mobile base and the manipulator simultaneously during the pick-up operation. We have demonstrated the feasibility of this idea when there is no uncertainty in part pose estimates [1].

In most cases, there is some uncertainty in part pose estimates when the mobile manipulator attempts to pick up the part. This uncertainty affects the speed of the pick-up operation. For example, Fig. 1 illustrates a case where the gripper is not aligned well with the part during the pick-up operation due to the uncertainty in the part pose estimate. If the gripper moves at a fast speed as shown in 1(a), the gripper finger will collide with the part resulting in a failure to grasp. On the other hand, as shown in 1(b) if it moves at a slower speed, the gripper fingers will align with the part when they close resulting in a successful grasp. However, a slower gripper speed may require the mobile manipulator to slow down resulting in increased operation time. This illustrates the need for adjusting the gripper speed based on the part pose estimate uncertainty.

The first main problem investigated in this paper is the effect of part pose estimation uncertainties on gripper velocities. We are interested in understanding and characterizing how the probability of successful grasping depends upon gripper speed and part pose estimation uncertainties. This characterization helps us in selecting the appropriate gripper speed based on the expected uncertainties in the part pose estimates. Characterizing the probability of successful grasps through exhaustive simulation of various combinations of un-

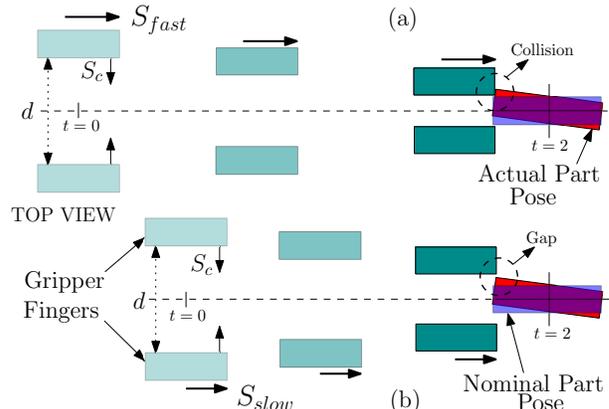


Fig. 1: Two scenarios (a) and (b) with different gripper speeds are shown. In (a),  $S_{fast}$  refers to the fast gripper speed and in (b),  $S_{slow}$  refers to the slow gripper speed.  $S_c$  refers to the gripper closing speed which is constant in both scenarios. The nominal pose of the part is in blue, the actual part pose is in red. The grasping time is dictated by the gripper closing time, which remains constant in both cases. Hence, in (a), the gripper starts closing further away from the part as compared to in (b).

derlying operation parameters is computationally not viable. Moreover, the contact physics between the gripper and the part with a moving gripper is complex and hence it is difficult to model the grasp success probability analytically. We use an active learning method to characterize the probability of successful pick-up operation using simulations.

Given the desired success probability, we are interested in moving the mobile base and the manipulator such that the pick-up operation time is minimized. This is done by accounting for the part pose estimation uncertainties and its effect on gripper speed. The second problem investigated in this paper is the problem of picking up the part using the manipulator while moving the mobile base from a given initial location to the goal location. The method will be designed for problems where the initial and goal locations of the mobile base are near the part location. We assume that a state space search will be performed to identify the mobile base's initial location and the goal location when we need to solve a large distance part transport problem [1]. The state space search is not the focus of this paper. Mobile manipulator trajectories computed using the method described in this paper can be used as motion primitives in the state space search to solve large distance part transport problems.

## II. RELATED WORK

Significant work has been done to study the effects of uncertainties in shape, part pose, contact, physics, dimensions, environment and perception for grasping of parts [2]–[7]. These studies deal with grasping when the gripper is

stationary with respect to the part.

Deep learning algorithms have been used for robotic grasping [8]–[10]. Classification of grasps based on the degrees of freedom of the object and the grasp parameters using techniques such as Support Vector Machines (SVM) [11] and AdaBoost [12] have been used. The focus in these works has been to generate techniques to determine the grasping strategies for objects. In our work, we determine the grasp strategies using existing grasp planners [13]. We focus on determining the approach velocity of the gripper during grasping with the appropriate grasping strategy.

Physics-based trajectory generation for grasping using optimization has been demonstrated in [14], [15]. Combined grasp and manipulation planning using trajectory optimization has been implemented in [16]. The grasping discussed in this work is with a moving gripper. The physical interactions of the object with a moving gripper are different as compared to a stationary gripper. The velocity of the gripper approaching the object plays an important role, which has not been the focus of most of the previous works in the area of robotic grasping.

Sampling and search-based methods have been studied for motion planning for high-dimensional systems such as mobile manipulators [17]–[21]. In [22], the pick of moving objects using manipulators is studied using a search-based approach by defining specialized motion primitives and heuristics. The final solution is resolution optimal which may result in non-smooth trajectories. Several Optimization based algorithms [23]–[25] have been developed to generate smooth trajectories for high dimensional systems. However, most of these techniques are used for point-to-point path planning and not for continuous trajectory generation as desired for specific end-effector trajectories.

The end-effector is required to move through a set of pre-defined waypoints for a continuous motion in a constrained trajectory [26], [27]. The solution for each joint angle is found either as a parametric curve using discrete parameter optimization [28]–[30] or as a functional using optimal control [31], [32]. Spline approximation of joint trajectories as a function of time or arc-length parameters have been used followed by Sequential Quadratic Programming (SQP) [33]. Researchers have also explored genetic algorithms [34] to generate path-constrained trajectories for manipulators.

Trajectory generation for high-DOF systems like mobile manipulators [35], humanoids [36] has been studied. Convex optimization [37], [38] and Quadratic Programming (QP) [39], [40] have been used to generate trajectories for high-DOF systems in dynamic, environments and for end-effector path tracing by minimizing pose error. Joint limits, collisions and velocities are also considered via constraints. Jacobian approximations can be used for joint velocity control-based manipulator trajectory generation [41], [42]. In these methods having an appropriate seed for computing subsequent configurations is important.

### III. PROBLEM FORMULATION

#### A. Definitions

We define four frames of reference.  $\mathcal{W}$  is the world frame of reference. The mobile manipulator poses and velocities are with respect to  $\mathcal{W}$ .  $\mathcal{P}$  is the frame of reference attached to the part. Frame  $\mathcal{B}$  is such that the location of its origin is the same as the origin of  $\mathcal{P}$ , but the axes are aligned with  $\mathcal{W}$ . The origin of  $\mathcal{B}$  changes with a change in the location of the part.  $\mathcal{G}$  is the frame of reference attached to the gripper. A pose is defined by the homogeneous transformation matrix  $T$ . The poses of the part, the mobile base and the gripper in any frame  $\mathcal{F}$  are denoted by  ${}^{\mathcal{F}}T_p$ ,  ${}^{\mathcal{F}}T_{mb}$  and  ${}^{\mathcal{F}}T_g$  respectively.

The mobile manipulator is an  $n + 3$  degrees of freedom (DOF) system with configuration variables  $\Theta$  as  $(x, y, \phi, \theta_1, \dots, \theta_n)$ .  $(x, y)$  is the location and  $\phi$  is the orientation of the mobile base w.r.t.  $\mathcal{W}$ .  $\theta_1, \dots, \theta_n$  are the joint angles of the manipulator. The kinematic model includes the forward kinematics (FK) which maps the mobile manipulator DOFs to the gripper (attached to the end-effector) pose  ${}^{\mathcal{W}}T_g$  and the Jacobian  $J$  ( $6 \times (n + 3)$ ), which maps the joint velocities of the manipulator and the velocities of the mobile base ( $\dot{\Theta}$ ) to the gripper velocity  ${}^{\mathcal{W}}\mathbf{V}_g$  ( $6 \times 1$ ).

The approach vector of the gripper towards the part in the frame  $\mathcal{B}$  is  $\hat{n}_g$ . The gripper orientation does not change while it is approaching the part. The pose of the gripper with respect to the frame  $\mathcal{P}$  is  ${}^{\mathcal{G}}T_p$ . A grasping strategy  $\Gamma_g$  is defined by the pair  $(\hat{n}_g, {}^{\mathcal{G}}T_p(t'))$ , where  $t'$  is the time instance at which grasping ends. In essence, it gives the direction from which the gripper will approach the part and the orientation of the gripper.

The gripper velocity is denoted by  $\mathbf{V}_g$  in  $\mathcal{W}$ . The gripper velocity in the frame  $\mathcal{W}$  is the same as in  $\mathcal{B}$ . The gripper speed is denoted by  $S_g$  (this includes only the translation velocity magnitude). As the gripper orientation does not change during grasping, the angular velocity components of  $\mathbf{V}_g$  are zero. The gripper closing speed is denoted by  $S_c$ .

The standard deviation in the estimated part pose uncertainty is denoted by  $\sigma$  and the mean is zero. We define  $\gamma$  as the grasping success probability threshold. In other words, during grasping, the probability of success should be greater than  $\gamma$ .

#### B. Problem Statement

Given  $\Theta_{initial}$ ,  $\Theta_{goal}$ ,  ${}^{\mathcal{W}}T_p$ ,  $\Gamma_g$ ,  $\gamma$  and  $\sigma$ , the local trajectory generation can be represented as an optimal control problem. Our formulation is one that transforms the basic optimal control problem into one of nonlinear programming using direct transcription.

$$\begin{aligned} & \underset{\Theta(t)}{\text{minimize}} && T_\tau \\ & \text{subject to} && C(\Theta(t)) \leq 0, \quad 0 \leq t \leq T_\tau \end{aligned}$$

Where,  $T_\tau$  is the time required to traverse a trajectory  $\tau$ .  $C(\Theta(t))$  is a vector representation of the constraints on the mobile manipulator at time  $t$  which include, the end-effector or gripper pose constraint while picking up the part, the Jacobian constraints for end-effector (or gripper) velocity, the mobile base non-holonomic constraints, grasping success probability threshold constraint, joint limit, joint rate,

velocity constraints and also the self and external collisions constraints.

To express the grasping success probability threshold constraint, we develop a meta-model that estimates the success probability as a function of  $S_g$  and  $S_c$  for a given  $\sigma$  (see Sec. IV for details).

#### IV. A META MODEL FOR ESTIMATING PART GRASPING SUCCESS PROBABILITY

To build a meta-model for estimating the grasping success probability, we use a real-time physics engine (Bullet Physics [43] in V-REP [44]) for simulating the grasping of parts with a moving gripper. We define grasping success conservatively, by measuring the gripper overlap on the part and checking whether the part is held inside the gripper while it is moving for a predefined period of time. Also, the distance between a fixed point on the gripper and a fixed point on the part is measured for any changes throughout the gripper motion. If the part moves more than a threshold amount after being grasped, we label that as a failure. Using this notion of grasping success, we use only the simulation data to build a classification model. Physical experiments with a gripper mounted on a manipulator for various gripper speeds and closing speeds have been conducted to verify the accuracy of the physics engine used for simulation. By placing three parts in different poses, we performed 90 physical experiments resulting in 94% match with simulations in terms of success and failure of grasps.

The success of a grasp is dependent on the part pose relative to the gripper, the gripper speed and the gripper closing speed. Given these inputs, determining the success of a grasp can be viewed as a classification problem. The training example vectors can be 5 dimensional with a  $3 \times 1$  part pose relative to the gripper ( $x, y, \phi$ ; this definition of part pose will be used in this section), the gripper speed ( $S_g$ ) and the gripper closing speed ( $S_c$ ) (Fig. 2). Some of these variables have a predictable effect on the success of grasp and we can reduce the dimension of the input vector for the classifier. For example, for a certain pair of  $S_g$  and  $S_c$ , if the grasp is successful for a  $S_c$ , it is also successful for a higher  $S_c$  for the same  $S_g$  and the part pose. Instead of using 5D input vectors, we use training example vectors consisting of only the  $3 \times 1$  part pose relative to the gripper. So instead of having one classification model in 5D we have  $k$  models in 3D and perform interpolation between them.

We take inspiration from [45] where active learning was used to determine the contact surface for collision detection in high dimensional configuration space. We proceed with a physics simulator based grasp success evaluation function.

##### A. Active Learning for Generating Classification Model

Our goal is to understand how much deviation from the nominal part pose will start causing grasping failures for a pair of  $S_g$  and  $S_c$ . We denote the classification surface as *Grasping Success Boundary (GSB)*. For a given  $(S_g, S_c)$ , the Grasping Success Boundary is  $GSB_{S_g, S_c}$ .

The approach for generating a nonlinear classifier based on SVM [46] is illustrated in Fig. 2. The description henceforth is for a pair of  $S_g$  and  $S_c$ .

We start with an initial classifier surface ( $GSB_0$ ) with few part pose samples and refine it using active learning. Our goal is to actively select pose samples so that a better approximation of the grasping success boundary  $GSB_1$  can be obtained subsequently. We start with equal weight on exploration and exploitation. During exploration after  $GSB_i$ , we bias random sampling of new poses in areas that were not explored before. This refines the *GSB* in places which initially had only a few samples. If the prediction accuracy after exploration increases, we increase the bias towards exploitation. At every step, new pose samples are added and the *GSB* is updated. This procedure is repeated until the prediction accuracy of the generated model is greater than a predefined threshold or when the total number of pose samples generated is larger than a given threshold.

For refining the generated *GSB*, we sample near the boundary by choosing a pair of support vectors of opposite labels and finding their midpoint in the feature space. This midpoint lies close to the boundary as stated by the maximal margin property of SVM [46]. This results in local refinement of the boundary.

##### B. Constructing Success Probability Meta-Model

Given a part pose uncertainty  $\sigma$ , our goal is to build a model that will predict success probability as a function of  $S_g, S_c$  using the classifier from the previous section.

We define Success Depth (SD) as the distance from the *GSB* inside the success region. It is approximated as,  $SD(p_0, GSB) = \min_{p \in GSB} dist(p_0, p)$ .

We define Failure Threshold Distance ( $D_{FT}$ ) to be the distance from *GSB* into the success region, such that any point  $p_0$  with  $SD(p_0, GSB) \geq D_{FT}$  is always a success. At the *GSB*, there is high uncertainty in the success classification because of the physics engine approximations. Hence, we set a  $D_{FT}$  to overcome this uncertainty resulting in a conservative definition of grasping success.

Once a  $GSB_{S_g, S_c}$  is generated satisfying the termination conditions, we query a large number of poses generated by the standard deviation  $\sigma$  in the part pose. They are labeled by taking into account their SD. The ratio of the number of successful grasps and the total number of query points is the probability of success for the tuple  $(\sigma, S_g, S_c)$ . We do this for pairs of feasible  $S_g$  and  $S_c$  for a particular  $\sigma$ . Furthermore, there is a need to interpolate to get the probability of success for an  $S_g, S_c$  pair for which this probability was not computed. For this, we fit a surface and use its lower bound approximation to generate a conservative probability of success for any  $S_g$  and  $S_c$  pair. This surface is an analytical function of  $S_g$  and  $S_c$  denoted by  $\rho_\sigma(S_g, S_c)$ . Fig. 3 shows an example of the generated  $\rho_\sigma(S_g, S_c)$  for two values of  $\sigma$ . This  $\rho_\sigma(S_g, S_c)$  will be used in the grasping success probability constraint in Sec. V.

The Fig. 3 also shows the results for this method and compares it with the probability computations using extensive sampling for two different levels of uncertainty in part pose. Extensive sampling required about 2000 samples (and hence that many simulation runs) to converge to a probability value

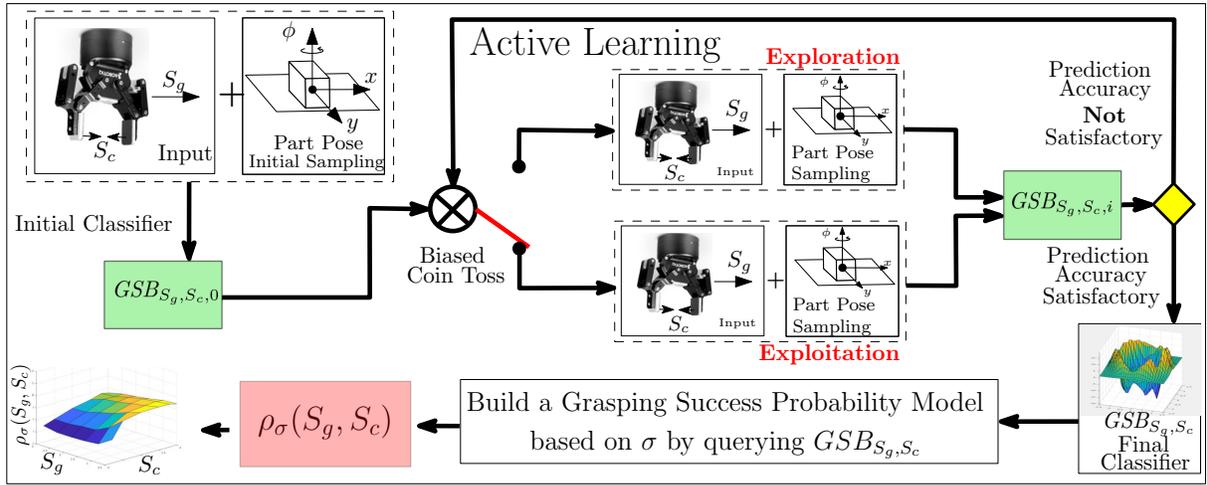


Fig. 2: The flowchart of the offline computation for the Grasping success probability model for one pair of  $S_g$  and  $S_c$

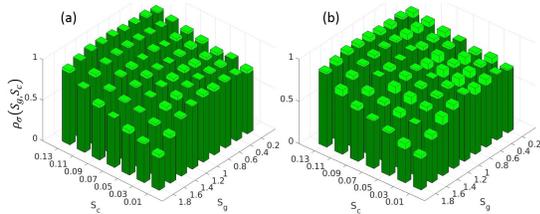


Fig. 3: These plots show the Probability of success vs the gripper speed and the closing speed for a part with (a) low uncertainty in part pose ( $\sigma_r = 2mm$  in  $x$  &  $y$ ,  $\sigma_\phi = 5^\circ$  (in orientation)), (b) high uncertainty  $\sigma_r = 12mm$ ,  $\sigma_\phi = 15^\circ$ . The light green bars show the probability using extensive sampling of the part pose with the given uncertainty for the pairs of  $S_g$  and  $S_c$ . The dark green bars show the probability measured with active learning for the same  $S_g$  and  $S_c$ .

for every pair of  $S_g$  and  $S_c$ . It can be observed that in all cases, the probability computed using active learning is lower than the probability computed using extensive sampling. The extent to which it is lower is dependent on  $\sigma$ . Because for a higher  $\sigma$ , the number of samples close to the GSB is higher and samples with  $SD < D_{FT}$  will be labeled as failures. It can also be seen from Fig. 3 that for lower uncertainty levels, the difference between the active learning method and the extensive sampling method is less. The reason for this is that because of low  $\sigma$ , almost all the samples are away from the GSB, hence their  $SD > D_{FT}$  resulting in an accurate prediction of success. For high  $\sigma$  in the worst case, the difference between the active learning and the extensive sampling probabilities is about 10% of the active learning probability. The uncertainty in the part pose  $\sigma$  typically depends on the vision system used for part detection. Generating the above  $GSB_{S_g, S_c}$  is a one-time process for a particular part, grasping strategy, a grasping speed and a gripping closing speed. It can be queried with any  $\sigma$ . However, the method of extensive sampling for probability generation requires resampling with every new  $\sigma$ . Each simulation run for a sample is about 1.3 seconds and hence it is infeasible to simulate such a large number of poses. To train the model using active learning with exploration and exploitation, the number of samples that we need on average is about 350 to 500 for 96% prediction accuracy. This greatly reduces the computation time.

## V. TRAJECTORY GENERATION

### A. Definitions

The trajectory planning problem is formulated as a non-linear optimization problem. Given,  $\Theta_{initial}$ ,  $\Theta_{goal}$ ,  ${}^W T_p$ ,  $\Gamma_g$ ,  $\sigma$  and  $\gamma$

$$\text{minimize } T \text{ s.t.} \quad (1)$$

$$C_{mbpath}(\Theta, \dot{\Theta}) \leq 0, C_{grasping}(\Theta, \dot{\Theta}) \leq 0 \quad (2)$$

$$C_{uncertainty} \leq 0, C_{collision}(\Theta) \leq 0 \quad (3)$$

$$C_{initial}(\Theta, \dot{\Theta}) \leq 0, C_{final}(\Theta, \dot{\Theta}) \leq 0 \quad (4)$$

$$C_{joint}(\Theta) \leq 0, C_{joint-rates}(\dot{\Theta}) \leq 0 \quad (5)$$

**Path Constraint for the Mobile Base:** This includes the non-holonomic constraint ( $\dot{x} \sin \phi - \dot{y} \cos \phi = 0$ ) as well as any other path constraint. We also have a constraint that during grasping, the mobile base should lie inside the grasping area  $A_g$ . Grasping area is an area around the part within which when the mobile base is located, the end-effector can reach the part and grasp it with a specific grasping strategy [1]. These constraints are represented as  $C_{mbpath}$ .

**Grasping Constraints:** Let  $T_1$  be the time at which grasping starts (grasper starts closing) and  $T_2$  be the time at which grasping is completed (grasper is at the grasping location and closed). Therefore,  $0 \leq T_1 \leq T_2 \leq T$ . This constraint is specific to a grasping strategy  $\Gamma_g$  i.e ( $\hat{n}_g, {}^G T_p(T_2)$ ). During the time interval  $T_2 - T_1$ , the process of grasping is executed, i.e., this is the time required for the gripper to close. The pose of the gripper should follow the velocity constraint on configuration variables ( $J(\Theta(t))\dot{\Theta}(t) = \mathbf{V}_g$ ) and pose constraint during grasping ( $FK(\Theta(T_2)) = {}^G T_p(T_2)$ ).

**Grasping Success Probability Constraints:**  $S_g$  and  $S_c$  determine the probability of success as described in Sec. IV. Hence, in order to make sure that we are moving the gripper such that even in the presence of uncertainty the resulting grasping is successful, we use the function  $\rho_\sigma(S_g, S_c)$  generated in Sec. IV for an uncertainty level  $\sigma$  to get additional constraints on  $S_g$  and  $S_c$ . The  $C_{uncertainty}$  can be written as  $\gamma - \rho_\sigma(S_g, S_c) \leq 0$ . Where,  $\gamma$  is a given threshold of success to grasp.

**Other Constraints:** Eq. 4 represents the constraints on initial and final configuration of the mobile-base. Eqn. 5 represents the joint position and velocity constraints on the manipulator.

### B. Successive Refinement Procedure

We represent each DOF of the mobile manipulator as a polynomial in time ( $\Theta_i = \sum_{k=0}^m a_{i,k} t^k$ ). The degree ( $k$ ) of this polynomial depends on the expected motion. Our exploratory investigation demonstrated that a cubic polynomial is sufficient to represent motions for the mobile base and the wrist joints and a quintic polynomial is needed for the base and shoulder joints of the manipulator.

Let,  $q$  be the vector of optimization variables. It includes  $a_{1-9,k}$  (parameters of the polynomials),  $T$ ,  $T_1$ ,  $T_2$ , and  $S_g$ . We have developed a successive refinement approach to solve the optimization problem. In this approach, the seed for the next optimization is the solution to the current optimization.

We discretize the time in the following manner for evaluating constraints. Time intervals  $[0, T_1]$ ,  $[T_1, T_2]$ , and  $[T_2, T]$  are uniformly sampled for  $m$  time instances in each interval. The mentioned constraints are satisfied at each time instance.

$q_0$  is the initial value (seed) for the optimization variable with  $a_{1,1} = x_i$ ,  $a_{2,1} = y_i$  and  $a_{3,1} = \phi_i$  ( $x_i, y_i, \phi_i$  is initial pose of mobile-base).  $T_1, T_2$ , and  $T$  are initialized such that  $0 \leq T_1 < T_2 \leq T$ . Other elements of  $q_0$  are assigned randomly. The *solveNLP* function takes in the seed, the objective function and the constraints and uses non-linear programming to determine a locally optimal solution. The following steps describe our approach.

- 1)  $q_1 \leftarrow \text{solveNLP}(q_0, \text{ObjFunc}, \text{Constraints})$  where,  $\text{ObjFunc} = T$ ;  $\text{Constraints} : C_{mbpath}, C_{i,f}, C_{joints}$ . This gives a feasible trajectory for the mobile base.
- 2)  $q_2 \leftarrow \text{solveNLP}(q_1, \text{ObjFunc}, \text{Constraints})$  where,  $\text{ObjFunc} = T$ ;  $\text{Constraints} : C_{mbpath}, C_{i,f}, C_{joints}, C_{gT_2}$ . This step results in a trajectory such that the end-effector is at the grasping pose at time  $T_2$ .
- 3)  $q_3 \leftarrow \text{solveNLP}(q_2, \text{ObjFunc}, \text{Constraints})$  where,  $\text{ObjFunc} = T$ ;  $\text{Constraints} : C_{mbpath}, C_{i,f}, C_{joints}, C_{gT_2}, C_{jacobian}$ . This step results in a trajectory such that the end-effector follows the Jacobian constraints for  $t \in [T_1, T_2]$  and it is at the grasping pose at time  $T_2$ .
- 4)  $q_4 \leftarrow \text{solveNLP}(q_3, \text{ObjFunc}, \text{Constraints})$  where,  $\text{ObjFunc} = T$ ;  $\text{Constraints} : C_{mbpath}, C_{i,f}, C_{joints}, C_{gT_2}, C_{jacobian}, C_{uncertainty}$ . This step ensures that the gripper speed is such that the probability of grasping success threshold is met for a given  $\sigma$  and  $P_p$ .
- 5)  $q_5 \leftarrow \text{solveNLP}(q_4, \text{ObjFunc}, \text{Constraints})$  where,  $\text{ObjFunc} = T$ ;  $\text{Constraints} : C_{mbpath}, C_{i,f}, C_{joints}, C_{gT_2}, C_{jacobian}, C_{uncertainty}, C_{coll}$ . Finally we add the collision constraints for generating a feasible trajectory.

The non-linear programming in each step terminates when the improvement in the objective function or the step size falls below a critical tolerance.

The final trajectory  $\tau$  is generated from  $q_5$ . Since the parametric equations in time are polynomials and the constraints on the manipulator are essentially between  $T_1$  and  $T_2$ , it may

exhibit undesirable motions between 0 to  $T_1$  and between  $T_2$  to  $T$ . Therefore, we refine the manipulator trajectory in those intervals separately. We use STOMP [23], ensuring that the joint and velocity constraints at these points are met.

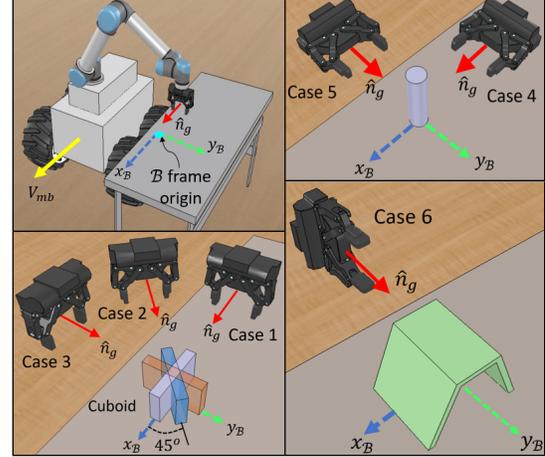


Fig. 4: The description of the test cases, the parts and the grasping strategies. The direction of  $V_{mb}$  (solid yellow arrow) is the same in all cases.  $\hat{n}_g$  is the direction of the gripper speed  $S_g$  (solid red arrow). Blue and green dashed arrows represent the  $x$  and  $y$  axes of frame  $B$ .

## VI. RESULTS

We have considered a 9 DOF mobile manipulator with a differentially driven mobile base (InspectorBot), a UR5 manipulator, and a Robotiq 2-fingered gripper. We tested the planner on 6 different test cases, some of which are shown in Fig. 4. The three parts shown require three different gripper orientations for grasping. The test cases include different grasping strategies for different directions of relative motion between the gripper and the mobile base. The method can be extended to any gripper and part orientations without loss of generality. In each test case, we vary the desired  $\hat{n}_g$  while the mobile base follows similar paths. The initial location of the mobile base in each case is  $(0m, 0m)$  and its orientation is  $\pi/6$  rad and the final location and orientation is  $(8m, 0m)$  and  $0$  rad respectively. We allow a tolerance of  $0.5$  m in its position and  $\pi/4$  rad in its orientation at the goal location. The nominal part location was  $(3m, 4m)$  for all test cases and the nominal part orientations were  $0, \pi/4, \pi/2, 0, \pi/4,$  and  $\pi/2$  for the 6 cases respectively. The grasping direction  $\hat{n}_g$  is determined according to the nominal part orientations. The maximum linear velocity of the mobile robot is  $2$  m/s and the maximum joint velocity for the manipulator is  $\pi$  rad/sec. We consider the same uncertainty levels as mentioned in Sec. IV. The algorithm was implemented using MATLAB on a computer with an Intel Xeon 3.50GHz processor and 32GB of RAM. *Interior point* method from Matlab's *fmincon* library was used as the optimization algorithm. For SVM, we have used Thundersvm [47] for fast computation in C++. The physics-based simulations were conducted in VREP.

We have presented the error comparison of our successive refinement based approach by benchmarking it against the non-linear programming with all the constraints combined together (called No Sequencing in the tables) with randomly selected initial seeds satisfying  $0 \leq T_1 < T_2 \leq T$ . The

TABLE I

Test Case	Smb (m/s)		Sg (m/s)		Sc (m/s)	
	Sequential	No Sequencing	Sequential	No Sequencing	Sequential	No Sequencing
1	0.82	0.28	0.26	0.28	0.13	0.15
2	0.71	0.27	0.42	0.48	0.12	0.15
3	0.22	0.19	0.21	0.3	0.15	0.15
4	0.87	0.2	0.27	0.22	0.11	0.15
5	0.94	0.13	0.29	0.21	0.14	0.15
6	0.38	0.07	0.38	0.09	0.14	0.15

TABLE II

Test Case	Pose Error during Grasping (mm)				Avg Non-holonomic Constraint violation (m)	
	Sequential		No Sequencing		Sequential	No Sequencing
	Position	Orientation	Position	Orientation		
1	2.00E-04	3.10E-05	2.80E-03	3.30E-03	1.70E-03	9.08E-02
2	6.30E-04	2.40E-05	6.00E-03	7.40E-04	2.10E-03	7.30E-02
3	4.10E-04	2.10E-04	1.70E-03	1.60E-03	1.38E-02	7.10E-02
4	2.40E-04	1.01E-04	8.70E-03	1.40E-04	1.43E-03	7.00E-03
5	3.00E-04	8.00E-04	1.34E-03	9.80E-03	6.70E-03	5.50E-03
6	2.50E-04	2.90E-03	7.65E-03	1.20E-03	7.10E-03	8.60E-03

results in tables I, II, III are for medium level uncertainty in part pose given by  $\sigma_r = 7mm$ ,  $\sigma_\phi = 10^\circ$ .  $\gamma$  is 0.96. The simulation and physical experiments can be seen in the video at <https://www.youtube.com/watch?v=8fWLFA97TC0>

We observe from table I the mobile base speed is higher in our approach. Since the trajectory time depends on the mobile base motion, a high speed plays a significant role in reducing overall time. In table II, we observe that the pose errors during grasping are significantly lower for our approach as compared to the ones with no sequencing. Moreover, the non-holonomic constraint violations for the mobile base motion are also lower. From table III, it can also be seen that the trajectory execution time is lower with our approach. The probability of grasping success values from the model are similar for both the approaches. We observe that the computation time for our method is comparable to the non-linear programming with all the constraints combined together (No sequencing). The use of successive refinement method leads to the use of significantly improved seeds for successive stages of the optimization. This reduces the chances of returning a poor local minimum as the final solution. It also helps reduce the computation time.

We observe from table I that the gripper velocity in cases 1 and 4 is less than that of the mobile base for in approach. However, since the motions of both are in the same directions, the manipulator compensates for the motion of gripper so as to enable the mobile base to move faster resulting in a decrease in the time taken to reach the goal. For cases 2 and 3, the mobile base velocity decreases because the manipulator has to compensate not only along the motion of mobile base but also perpendicular to it. Cases 4 and 5 can be explained similarly.

In table IV the results of low and high uncertainties for cases 1, 4, and 6 are presented. The velocity of the gripper is higher for the case of low uncertainty and lower in the case of high uncertainty. Subsequently, the mobile base velocities in the two cases are higher and lower respectively as well. Also, the probability of grasping success is higher for low uncertainty and lower for high uncertainty.

Fig. 5 shows the capabilities of the planner when we

TABLE III

Test Case	Time taken for Grasping (s)		Trajectory Execution Time (s)		Probability of success	
	Sequential	No Sequencing	Sequential	No sequencing	Sequential	No Sequencing
1	0.65	0.57	17.24	30.14	96.80%	96.01%
2	0.71	0.57	26.71	29.71	96.36%	96.01%
3	0.57	0.57	29.69	30.17	96.20%	96.84%
4	0.77	0.57	13.00	13.54	96.30%	96.97%
5	0.61	0.56	17.54	19.54	96.16%	95.44%
6	0.61	0.57	24.04	25.12	96.48%	96.91%

change the joint velocity constraints. As can be seen, the mobile base velocity is high for high joint velocity limits, while the gripper velocity is low. However, when we reduce the joint velocity limits on the manipulator, the mobile base velocity also decreases as it has to slow down for the gripper to move with the desired velocity.

TABLE IV: The impact of uncertainty in part pose on the mobile base and gripper speeds and probability of success

Test Case	Avg Smb (m/s)		Sg (m/s)		Probability of Success	
	Low Uncertainty	High Uncertainty	Low Uncertainty	High Uncertainty	Low Uncertainty	High Uncertainty
1	1.32	0.24	1.05	0.26	98.97%	86.20%
4	1.21	0.45	0.87	0.27	98.15%	87.15%
6	1.18	0.31	0.76	0.17	98.10%	84.15%

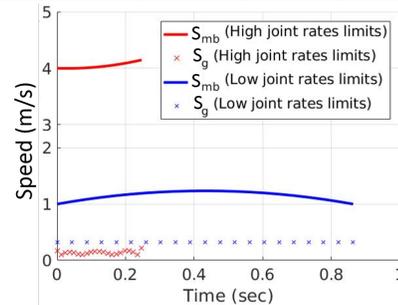


Fig. 5: The speeds of the mobile base and the gripper for high ( $3\pi$  rad/sec) and low limits ( $\frac{\pi}{2}$  rad/sec) on joint rates, but same the mobile base speed limit are shown for Case 1.

## VII. CONCLUSIONS AND FUTURE WORK

In this paper, we present an approach for solving two challenging problems encountered during part pick-up with a moving gripper. The first one addresses the effect of the uncertainty in the part pose on the gripper speed. We present an approach for constructing a meta-model for estimating the success probability of grasping a part as a function of the gripper speed and gripper closing speed. One of the limitations of using a physics engine for simulating success of grasps is that all the grasp failure modes may not be represented through it. The second problem addressed in the paper is generating a trajectory for the mobile manipulator to satisfy the previously generated success probability of grasping constraints. We use a sequential refinement method for optimization to generate trajectories and show that its performance is superior as compared to one without sequencing. Future work includes extending the method for more complex parts and cluttered environments. Also, the behavior of different sequencing of constraints for different environments will be investigated.

**Acknowledgment:** This work is supported in part by National Science Foundation Grant #1634431. Opinions expressed are those of the authors and do not necessarily reflect the views of the sponsor.

## REFERENCES

- [1] S. Thakar, L. Fang, B. C. Shah, and S. K. Gupta, "Towards time-optimal trajectory planning for pick-and-transport operation with a mobile manipulator," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Munich, Germany, Aug 2018.
- [2] M. Li, K. Hang, D. Kragic, and A. Billard, "Dexterous grasping under shape uncertainty," *Robotics and Autonomous Systems*, vol. 75, pp. 352–364, 2016.
- [3] Y. Zheng and W.-H. Qian, "Coping with the grasping uncertainties in force-closure analysis," *The International Journal of Robotics Research*, vol. 24, no. 4, pp. 311–327, 2005.
- [4] V. N. Christopoulos and P. Schrater, "Handling shape and contact location uncertainty in grasping two-dimensional planar objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 1557–1563.
- [5] R. C. Brost, *Planning robot grasping motions in the presence of uncertainty*. Carnegie-Mellon University, The Robotics Institute, 1985.
- [6] N. B. Kumbla, S. Thakar, K. N. Kaipa, J. Marvel, and S. K. Gupta, "Handling perception uncertainty in simulation-based singulation planning for robotic bin picking," *Journal of Computing and Information Science in Engineering*, vol. 18, no. 2, p. 021004, 2018.
- [7] N. B. Kumbla, S. Thakar, K. N. Kaipa, J. Marvel, and S. K. Gupta, "Simulation based on-line evaluation of singulation plans to handle perception uncertainty in robotic bin picking," in *ASME 12th International Manufacturing Science and Engineering Conference (MSEC)*, 2017.
- [8] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.
- [9] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4461–4468.
- [10] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1316–1322.
- [11] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An svm learning approach to robotic grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, 2004, pp. 3512–3518.
- [12] R. E. Schapire, "Explaining adaboost," in *Empirical inference*. Springer, 2013, pp. 37–52.
- [13] A. T. Miller and P. K. Allen, "Graspt! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, 2004.
- [14] N. Kitaev, I. Mordatch, S. Patil, and P. Abbeel, "Physics-based trajectory optimization for grasping in cluttered environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 3102–3109.
- [15] M. Moll, L. Kavraki, J. Rosell *et al.*, "Randomized physics-based motion planning for grasping in cluttered and uncertain environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 712–719, 2018.
- [16] M. B. Horowitz and J. W. Burdick, "Combined grasp and manipulation planning as a trajectory optimization problem," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 584–591.
- [17] J. Zhou, R. Paolini, A. M. Johnson, J. A. Bagnell, and M. T. Mason, "A probabilistic planning framework for planar grasping under uncertainty," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2111–2118, 2017.
- [18] S. M. Lavalle, "Planning Algorithms," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2013.
- [19] F. Burget, M. Bennewitz, and W. Burgard, "Bi 2 rrt\*: An efficient sampling-based path planning framework for task-constrained mobile manipulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.
- [20] A. M. Kabir, B. C. Shah, and S. K. Gupta, "Trajectory planning for manipulators operating in confined workspaces," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Munich, Germany, Aug 2018.
- [21] V. Paliana and K. Gupta, "Mobile manipulator planning under uncertainty in unknown environments," *The International Journal of Robotics Research*, vol. 37, no. 2-3, pp. 316–339, 2018.
- [22] A. Menon, B. Cohen, and M. Likhachev, "Motion Planning for Smooth Pickup of Moving Objects," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [23] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic Trajectory Optimization for Motion Planning," in *International Conference on Robotics and Automation (ICRA)*, 2011.
- [24] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.
- [25] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [26] J. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The International Journal of Robotics Research*, vol. 4, no. 3, pp. 3–17, 1985.
- [27] P. M. Bhatt, P. Rajendran, K. McKay, and S. K. Gupta, "Context-dependent compensation scheme to reduce trajectory execution errors for industrial manipulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019.
- [28] D. Constantinescu and E. A. Croft, "Smooth and time-optimal trajectory planning for industrial manipulators along specified paths," *Journal of robotic systems*, vol. 17, no. 5, pp. 233–249, 2000.
- [29] A. Kabir, A. Kanyuck, R. K. Malhan, A. V. Shembekar, S. Thakar, B. C. Shah, and S. K. Gupta, "Generation of synchronized configuration space trajectories of multi-robot systems," in *IEEE International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019.
- [30] R. K. Malhan, A. M. Kabir, B. C. Shah, and S. K. Gupta, "Identifying feasible workpiece placement with respect to redundant manipulator for complex manufacturing tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, Montreal, Canada, May 2019.
- [31] M. Giffthaler, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient Kinematic Planning for Mobile Manipulators with Non-holonomic Constraints Using Optimal Control," *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3411–3417, 2017.
- [32] Q. Li and S. Payandeh, "Optimal-control approach to trajectory planning for a class of mobile robotic manipulations," *Journal of Engineering Mathematics*, vol. 67, no. 4, pp. 369–386, 2010.
- [33] A. Gasparetto and V. Zanotto, "A new method for smooth trajectory planning of robot manipulators," *Mechanism and machine theory*, vol. 42, no. 4, pp. 455–471, 2007.
- [34] M. Tarokh and X. Zhang, "Real-time motion tracking of robot manipulators using adaptive genetic algorithms," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, pp. 697–708, 2014.
- [35] H. G. Tanner and K. J. Kyriakopoulos, "Nonholonomic motion planning for mobile manipulators," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 2000, pp. 1233–1238.
- [36] A. Dietrich, "Dynamic Whole-Body Mobile Manipulation with a Torque Controlled Humanoid Robot via Impedance Control Laws," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3199–3206, 2011.
- [37] P. Lehner, A. Sieverling, and O. Brock, "Incremental, sensor-based motion generation for mobile manipulators in unknown, dynamic environments," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 4761–4767, 2015.
- [38] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," *IEEE International Conference on Robotics and Automation (ICRA)*, no. June, pp. 5495–5502, 2015.
- [39] B. Bäuml, F. Schmidt, T. Wimböck, O. Birbach, A. Dietrich, M. Fuchs, W. Friedl, U. Frese, C. Borst, M. Grebenstein, O. Eiberger, and G. Hirzinger, "Catching flying balls and preparing coffee: Humanoid rollin' justin performs dynamic and sensitive tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 3443–3444.
- [40] V. Falkenhahn, F. A. Bender, A. Hildebrandt, R. Neumann, and O. Sawodny, "Online tcp trajectory planning for redundant continuum manipulators using quadratic programming," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 2016, pp. 1163–1168.
- [41] A. Reiter, A. Müller, and H. Gatttringer, "On higher order inverse kinematics methods in time-optimal trajectory planning for kinematically

- redundant manipulators,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1681–1690, 2018.
- [42] S. R. Buss, “Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods,” *IEEE Journal of Robotics and Automation*, vol. 17, no. 1-19, p. 16, 2004.
- [43] E. Coumans, “Bullet physics simulation,” in *ACM SIGGRAPH 2015 Courses*, 2015, p. 7.
- [44] M. F. E. Rohmer, S. P. N. Singh, “V-rep: a versatile and scalable robot simulation framework,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.
- [45] J. Pan, X. Zhang, and D. Manocha, “Efficient penetration depth approximation using active learning,” *ACM Transactions on Graphics*, vol. 32, no. 6, 2013.
- [46] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [47] Z. Wen, J. Shi, Q. Li, B. He, and J. Chen, “Thundersvm: A fast svm library on gpus and cpus,” *Journal of Machine Learning Research*, vol. 19, no. 21, 2018.