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

Shantanu Thakar[1] and Vivek Annem[1] and Ariyan Kabir[1] and Pradeep Rajendran[1] and Satyandra Gupta[1]

***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 for picking-up the part that satisfy the given level of success probability threshold.**

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 is not time-optimal as parts can be grasped 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 underlying 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. Therefore, we will use an active learning method to characterize the probability of successful pick-up operation using simulations.

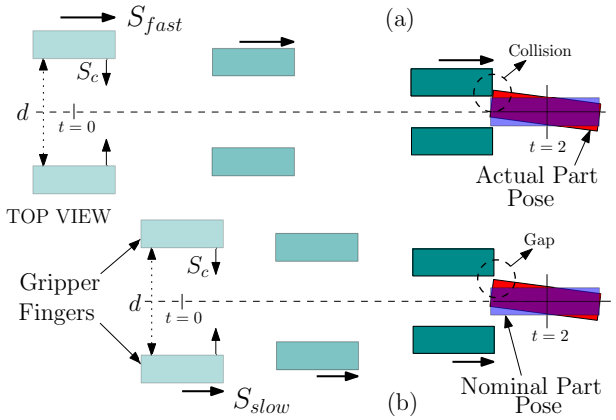

Fig. 1: Two scenarios (a) and (b) with different gripper speeds are shown. In (a), S_{fast} refers to the fast gripper speed and in (b), S_{slow} refers to the slow gripper speed. S_c refers to the gripper closing speed which is constant in both scenarios. The nominal pose of the part is in blue, the 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).

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 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 etc. for grasping of parts [2]–[10]. However,

these studies deal with grasping when the gripper is stationary with respect to the part.

Learning based techniques have been used for grasping of parts using different types of grippers [11]–[22]. Deep learning algorithms have been used for robotic grasping [23]–[27]. Classification of grasps based on the degrees of freedom of the object and the grasp parameters using techniques such as Support Vector Machines (SVM) [28]–[31] and AdaBoost [32] have been used. Moreover, vision based active learning for robotic grasping was demonstrated in [33]–[35]. The focus in these works has been how to grasp the part. However, in our work, we assume that how the part is being grasped is given.

Trajectory generation for grasping using physics-based simulations has been demonstrated in [36], [37]. Combined grasp and manipulation planning using trajectory optimization has been implemented in [38]. The grasping discussed in this paper is with a moving gripper. The interactions with the parts is different as compared to a stationary gripper. Moreover, the gripper velocity is generated through the motions resulting from the motions of the entire body of the mobile manipulator.

Sampling and search-based methods have been studied for motion planning for high-dimensional systems such as mobile manipulators [39]–[50]. Several Optimization based algorithms [51]–[53] 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.

The end-effector is required to move through a set of predefined waypoints for a continuous motion in a constrained trajectory [54], [55]. The solution for each joint angle is found either as a parametric curve using discrete parameter optimization [56], [57] or as a functional using optimal control [58]–[60]. Cubic spline [61] and B-spline [62] approximation of joint trajectories as a function of time, or arc-length parameters have been used with Sequential Quadratic Programming (SQP) to solve the problem. Researchers have also explored sampling-based approach [63] and genetic algorithm [64] to generate path-constrained trajectories for manipulators.

Trajectory generation for high-DOF systems like mobile manipulators [65], [66], humanoids [67]–[69] has been studied. Convex optimization [70], [71] and Quadratic Programming (QP) [59], [72]–[74] have been studied to generate trajectory for high-DOF systems in dynamic environments. Jacobian approximations have been used for joint velocity control-based manipulator trajectory generation [75]–[77] Non-linear programming [75], [78], Constrained Quadratic Programming (cQP) [79] and QP [80] have been used to generate time-optimal trajectory by minimizing pose error. Joint limits, collisions and velocities have been considered as constraints.

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. $\mathcal{B}$ is a frame of reference on the table on which the part is kept. $\mathcal{G}$ 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 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 nominal pose of the part.

The mobile manipulator is an $n+3$ degrees of freedom (DOF) system with configuration variables Θ as $(x, y, \phi, \theta_1, \ldots, \theta_n)$. (x, y, ϕ) are w.r.t. $\mathcal{W}$. 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, which maps the joint rates of the manipulator and the velocities of the mobile base $(\dot{\Theta})$ to the gripper velocity ${}^{\mathcal{W}}\mathbf{V}_g$ (6×1).

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 ${}^{\mathcal{G}}T_{\mathcal{P}}$. A grasping strategy Γ_g is defined by the pair $(\hat{n}_g,\ {}^{\mathcal{G}}T_{\mathcal{P}}(t'))$, 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 $\mathcal{W}$ will be 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 σ and the mean is zero. We define γ as the grasping success probability threshold. In other words, during grasping, the probability of success should be greater than γ.

B. Problem Statement

Given $\Theta_{initial}$, Θ_{goal}, ${}^{\mathcal{W}}T_p$, Γ_g, γ and σ, 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.

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

Where, T_τ is the time required to traverse a trajectory τ. $C(\Theta(t))$ 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 σ (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 in V-REP) 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 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 and estimated an accuracy of about 94% of the simulations in terms of success and failure of grasp.

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 success of a grasp can be viewed as a classification problem. The training example vectors can be 5 dimensional with a 3×1 part pose relative to the gripper (x, y, ϕ; 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 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 3×1 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 [81] 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,Sc), the Grasping Success Boundary is GSB_{S_g,S_c}.

The approach for generating a nonlinear classifier based on SVM [82] is illustrated in Fig. 2. The description henceforth is for a pair of S_g and Sc.

We start with a 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 mid point in the feature space. This midpoint lies close to the boundary as stated by the maximal margin property of SVM [82]. This results in local refinement of the boundary.

B. Constructing Success Probability Meta-Model

Given a part pose uncertainty σ, 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 σ 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 (σ,S_g,S_c). We do this for pairs of feasible S_g and S_c for a particular σ. Furthermore, there is a need to interpolate to get the probability of success for a 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 σ. 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 to converge to a probability value 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

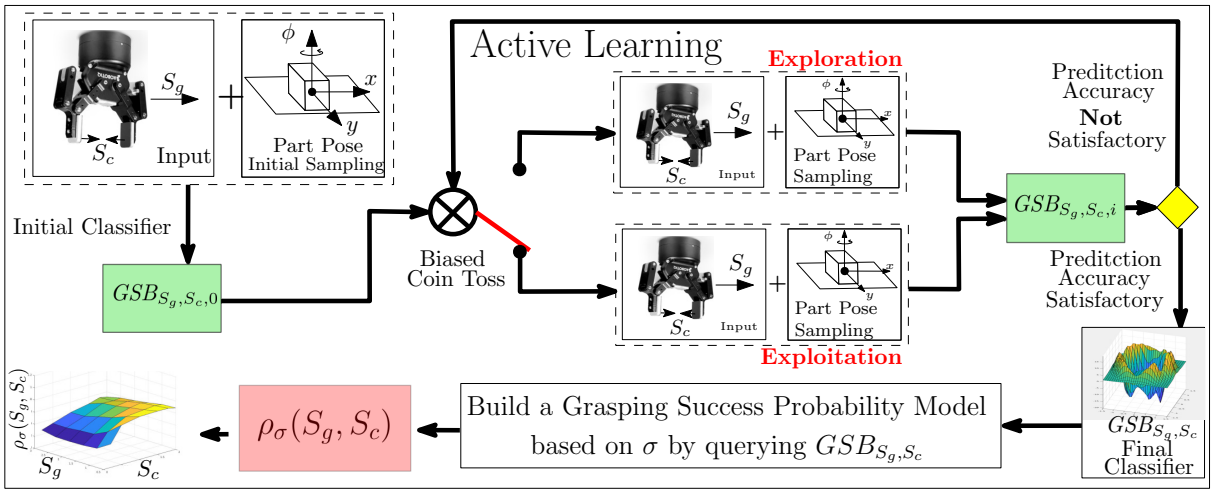

Fig. 2: The flowchart of the offline computation for the Grasping success probability model for one pair of S_g and S_c

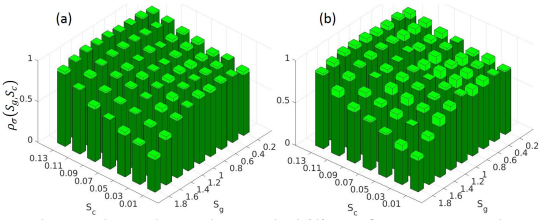

Fig. 3: These plots show the Probability of success vs the gripper speed and the closing speed for a part with (a) low uncertainty in part pose ($\sigma_r = 2mm$ in x & y, $\sigma_\phi = 5^o$ (in orientation)), (b) high uncertainty $\sigma_r = 12mm$, $\sigma_\phi = 15^o$. The light green bars show the 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.

is dependent on σ. Because for a higher σ, the number of samples close to the *GSB* is higher and samples with SD $<$ D_{FT} will be labeled as failure. 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 σ, almost all the samples are away from the GSB, hence their SD $> D_{FT}$ resulting in an accurate prediction of success. For high σ in the worst case, the difference between the active learning and the extensive sampling is about 10%.

The uncertainty in the part pose σ 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 σ. However, the method of extensive sampling for probability generation requires resampling with every new σ. 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 an average is about 350 to 500. This greatly reduces the computation time.

V. Trajectory Generation

A. Definition

The trajectory planning problem is formulated as a non-linear optimization problem. Given, $\Theta_{initial}$, Θ_{goal}, ${}^{\mathcal{W}}T_p$, Γ_g, σ and γ

$$\underset{\Theta}{\text{minimize}}\ T \ s.t. \tag{1}$$

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

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

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

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

Each of the constraints is explained as follows.

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 $\mathcal{A}_g$. 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 (gripper starts closing) and T_2 be the time at which grasping is completed (gripper 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 Γ_g i.e $(\hat{n}_g, {}^{\mathcal{G}}T_{\mathcal{P}}(T_2))$. 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 ($FK(\Theta(T_2) = {}^{\mathcal{G}}T_{\mathcal{P}}(T_2))$.

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 σ 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, γ is a given threshold of success to grasp.

Other Constraints: Eqn. 4 represents the constraints on initial and final configuration of the mobile-base. Eqn. 5 represents the joint position and velocity constraints of 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 initial solution for the next optimization step is generated by the previous step. The problem is set up as before.

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 constraints mentioned above 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, ϕ_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) $q_1 \leftarrow solveNLP(q_0, ObjFunc, Constraints)$ where, $Objfunc = T$; $Constraints : C_{mbpath}$, $C_{i,f}$, C_{joints}. This gives a feasible trajectory for the mobile base.
2) $q_2 \leftarrow solveNLP(q_1, ObjFunc, Constraints)$ where, $Objfunc = T$; $Constraints : C_{mbpath}$, $C_{i,f}$, 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) $q_3 \leftarrow solveNLP(q_2, ObjFunc, Constraints)$ where, $Objfunc = T$; $Constraints : C_{mbpath}$, $C_{i,f}$, 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) $q_4 \leftarrow solveNLP(q_3, ObjFunc, Constraints)$ where, $Objfunc = T$; $Constraints : C_{mbpath}$, $C_{i,f}$, 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 σ and P_p.
5) $q_5 \leftarrow solveNLP(q_4, ObjFunc, Constraints)$ where, $Objfunc = T$; $Constraints : C_{mbpath}$, $C_{i,f}$, 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.

τ is the final trajectory 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 [51], ensuring that the joint and velocity constraints at these points are met.

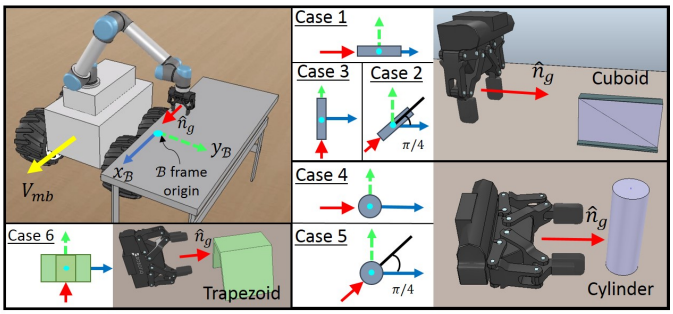

Fig. 4: The description of the test cases, the parts and the grasping strategies

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, which result in different directions of motion for the gripper and the mobile base. We can extend the study to any gripper and part orientation. 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 (0.5 m in position and $\pi/4$ rad in 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 π 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 [83] for fast computation in C++. The physics based simulations were conducted in VREP [84].

TABLE I: This table shows the impact of the uncertainty in part pose on the velocities of the mobile base and the gripper

Test Case	S_{mb} (m/s)		S_g (m/s)		S_c (m/s)	
	Sequential	Without Sequencing	Sequential	Without Sequencing	Sequential	Without 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

We have presented the error comparison of our successive refinement based approach by benchmarking it against the

TABLE II: This table shows the impact of the uncertainty in part pose on the velocities of the mobile base and the gripper

Test Case	Pose Error during Grasping (mm)				Avg Non-holonomic Constraint Error (m)	
	Sequential		Without Sequencing			
	Position	Orientation	Position	Orientation	Sequential	Without Sequencing
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

TABLE III: This table shows the impact of the uncertainty in part pose on the velocities of the mobile base and the gripper

Test Case	Time taken for Grasping (s)		Trajectory Execution Time (s)		Probability of success According to Metamodel	
	Sequential	Without Sequencing	Sequential	Without Sequencing	Sequential	Without 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%

non-linear programming with all the constraints combined together with randomly selected initial seeds satisfying $0 \leq T_1 < T_2 \leq T$. The results in tables I, II, III are for medium level uncertainty in part pose given by $\sigma_r = 7mm$, $\sigma_\phi = 10^o$. γ is 0.96.

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. Moreover, the non-holonomic constraints for the mobile base motion are also lower. From table III, it can also be observed that that the trajectory execution time is lower with our approach. The probability of success values from the model are similar for both the approaches. However, because of the high errors in the end-effector poses, actual successful grasping of the part may not be possible. We observe that the computation time for our method is comparable to the non-linear programming with all the constraints combined together. The use of successive refinement method leads to use of significantly improved seeds for successive stages of the optimization. This reduces the chance of the procedure from returning a poor local minima as the final solution. It also helps reduce the computation time.

Now we will discuss only our approach. We observe from table I that the gripper velocity in cases 1 and 4 is less than that of the mobile base. 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 the 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 the cases 1, 4, and 6 are presented. It can be observed that 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 change the constraints. As it can be seen, the mobile base velocity is high for high joint rates limits, while the gripper velocity is low. However, when we reduce the joint rate 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: This table shows the impact of the uncertainty in part pose on the velocities of the mobile base and the gripper

Test Case	Avg Mobile Base Velocity during grasping (m/s)		Grasping Velocity (Sg) (m/s)		Probability of Success using the Metamodel	
	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%

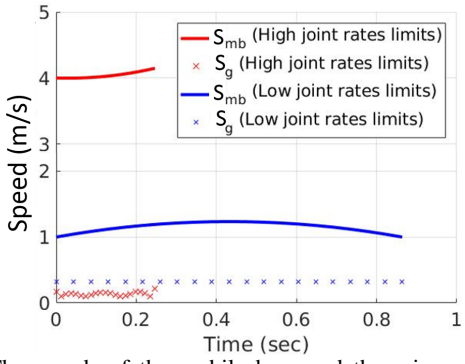

Fig. 5: The speeds of the mobile base and the gripper for high ($3\pi\ rad/sec$) and low limits($\frac{\pi}{2}\ rad/sec$) on joint rates, but same maximum for the mobile base speed limit are shown for Case 1.

VII. Conclusions

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. 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 doing optimization to generate trajectories. We show that this method performs better compared to a one-step optimization in terms of errors in part poses, the final trajectory execution time. It also results in the mobile base moving at higher speeds with the manipulator compensating for the motion of the gripper.

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] S. Dragiev, M. Toussaint, and M. Gienger, "Uncertainty aware grasping and tactile exploration," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 113–119.

[4] 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.

[5] V. N. Christopoulos and P. Schrater, "Handling shape and contact location uncertainty in grasping two-dimensional planar objects," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*. IEEE, 2007, pp. 1557–1563.

[6] R. C. Brost, *Planning robot grasping motions in the presence of uncertainty*. Carnegie-Mellon University, The Robotics Institute, 1985.

[7] J. Kim, K. Iwamoto, J. J. Kuffner, Y. Ota, and N. S. Pollard, "Physically based grasp quality evaluation under pose uncertainty," *IEEE Transactions on Robotics*, vol. 29, no. 6, pp. 1424–1439, 2013.

[8] 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.

[9] 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.

[10] M. Gualtieri, A. ten Pas, K. Saenko, and R. Platt, "High precision grasp pose detection in dense clutter," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 598–605.

[11] 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.

[12] A. Saxena, L. L. Wong, and A. Y. Ng, "Learning grasp strategies with partial shape information." in *AAAI*, vol. 3, no. 2, 2008, pp. 1491–1494.

[13] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," *arXiv preprint arXiv:1709.07857*, 2017.

[14] 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.

[15] C. Choi, W. Schwarting, J. DelPreto, and D. Rus, "Learning object grasping for soft robot hands," *IEEE Robotics and Automation Letters*, 2018.

[16] T. Osa, J. Peters, and G. Neumann, "Experiments with hierarchical reinforcement learning of multiple grasping policies," in *International Symposium on Experimental Robotics*. Springer, 2016, pp. 160–172.

[17] A. Boularias, O. Kroemer, and J. Peters, "Learning robot grasping from 3-d images with markov random fields," in *EEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2011, pp. 1548–1553.

[18] B. Huang, S. El-Khoury, M. Li, J. J. Bryson, and A. Billard, "Learning a real time grasping strategy," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 593–600.

[19] D. Fischinger, M. Vincze, and Y. Jiang, "Learning grasps for unknown objects in cluttered scenes," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 609–616.

[20] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[21] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.

[22] M. Laskey, J. Lee, C. Chuck, D. Gealy, W. Hsieh, F. T. Pokorny, A. D. Dragan, and K. Goldberg, "Robot grasping in clutter: Using a hierarchy of supervisors for learning from demonstrations," in *Automation Science and Engineering (CASE), 2016 IEEE International Conference on*. IEEE, 2016, pp. 827–834.

[23] E. Johns, S. Leutenegger, and A. J. Davison, "Deep learning a grasp function for grasping under gripper pose uncertainty," in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 2016, pp. 4461–4468.

[24] D. Kappler, J. Bohg, and S. Schaal, "Leveraging big data for grasp planning," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4304–4311.

[25] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 769–776.

[26] J. Varley, J. Weisz, J. Weiss, and P. Allen, "Generating multi-fingered robotic grasps via deep learning," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 4415–4420.

[27] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1316–1322.

[28] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An svm learning approach to robotic grasping," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 4. IEEE, 2004, pp. 3512–3518.

[29] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 616–629, 2011.

[30] Q. V. Le, D. Kamm, A. F. Kara, and A. Y. Ng, "Learning to grasp objects with multiple contact points," in *IEEE International Conference Robotics and Automation (ICRA)*. IEEE, 2010, pp. 5062–5069.

[31] H. Dang, J. Weisz, and P. K. Allen, "Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5917–5922.

[32] R. E. Schapire, "Explaining adaboost," in *Empirical inference*. Springer, 2013, pp. 37–52.

[33] M. Salganicoff, L. H. Ungar, and R. Bajcsy, "Active learning for vision-based robot grasping," *Machine Learning*, vol. 23, no. 2-3, pp. 251–278, 1996.

[34] L. Montesano and M. Lopes, "Active learning of visual descriptors for grasping using non-parametric smoothed beta distributions," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 452–462, 2012.

[35] D. Martínez, G. Alenyà, P. Jiménez, C. Torras, J. Rossmann, N. Wantia, E. E. Aksoy, S. Haller, and J. Piater, "Active learning of manipulation sequences," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5671–5678.

[36] N. Kitaev, I. Mordatch, S. Patil, and P. Abbeel, "Physics-based trajectory optimization for grasping in cluttered environments," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3102–3109.

[37] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa, "Physics-based grasp planning through clutter," pp. 78–85, 2012.

[38] M. B. Horowitz and J. W. Burdick, "Combined grasp and manipulation planning as a trajectory optimization problem," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 584–591.

[39] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[40] B. Akgun and M. Stilman, "Sampling heuristics for optimal motion planning in high dimensions," in *International Conference on Intelligent Robots and Systems*, 2011, pp. 2640–2645.

[41] S. M. Lavalle, "Planning Algorithms," *Journal of Chemical Information and Modeling*, vol. 53, no. 9, pp. 1689–1699, 2013.

[42] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 2. IEEE, 2000, pp. 995–1001.

[43] F. Burget, M. Bennewitz, and W. Burgard, "Bi 2 rrt*: An efficient sampling-based path planning framework for task-constrained mobile manipulation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),*. IEEE, pp. 3714–3721.

[44] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.

[45] P. Švec, B. C. Shah, I. R. Bertaska, J. Alvarez, A. J. Sinisterra, K. v. Ellenrieder, M. Dhanak, and S. K. Gupta, "Dynamics-aware target following for an autonomous surface vehicle operating under COLREGs in civilian traffic," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[46] 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.

[47] V. Pilania and K. Gupta, "A hierarchical and adaptive mobile manipulator planner with base pose uncertainty," *Autonomous Robots*, vol. 39, no. 1, pp. 65–85, 2015.

[48] ——, "Mobile manipulator planning under uncertainty in unknown environments," *The International Journal of Robotics Research*, vol. 37, no. 2-3, pp. 316–339, 2018.

[49] 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.

[50] K. Hauser and V. Ng-Thow-Hing, "Randomized multi-modal motion planning for a humanoid robot manipulation task," *The International Journal of Robotics Research*, vol. 30, no. 6, pp. 678–698, 2011.

[51] 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*, 2011.

[52] 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.

[53] 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.

[54] 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.

[55] O. Egeland, "Task-space tracking with redundant manipulators," *IEEE Journal on Robotics and Automation*, vol. 3, no. 5, pp. 471–475, 1987.

[56] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," *IEEE Journal on Robotics and Automation*, vol. 3, no. 2, pp. 115–123, 1987.

[57] 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.

[58] M. Galicki, "Time-optimal controls of kinematically redundant manipulators with geometric constraints," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 1, pp. 89–93, Feb 2000.

[59] M. Giftthaler, F. Farshidian, T. Sandy, L. Stadelmann, and J. Buchli, "Efficient Kinematic Planning for Mobile Manipulators with Non-holonomic Constraints Using Optimal Control," *International Conference on Robotics and Automation*, pp. 3411–3417, 2017.

[60] 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.

[61] T. Chettibi, H. Lehtihet, M. Haddad, and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *European Journal of Mechanics-A/Solids*, vol. 23, no. 4, pp. 703–715, 2004.

[62] 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.

[63] M. Cefalo, G. Oriolo, and M. Vendittelli, "Planning safe cyclic motions under repetitive task constraints," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 3807–3812.

[64] 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.

[65] M. Stilman, "Task constrained motion planning in robot joint space," *IEEE International Conference on Intelligent Robots and Systems (ICRA)*, pp. 3074–3081, 2007.

[66] ——, "Global manipulation planning in robot joint space with task constraints," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 576–584, June 2010.

[67] S. Dalibard, A. Nakhaei, F. Lamiraux, and J. P. Laumond, "Whole-body task planning for a humanoid robot: a way to integrate collision avoidance," in *2009 9th IEEE-RAS International Conference on Humanoid Robots*, Dec 2009, pp. 355–360.

[68] A. Dietrich, "Dynamic Whole-Body Mobile Manipulation with a Torque Controlled Humanoid Robot via Impedance Control Laws," *International Conference on Intelligent Robots and Systems*, pp. 3199–3206, 2011.

[69] D. Leidner, A. Dietrich, F. Schmidt, C. Borst, and A. Albu-Schäffer, "Object-centered hybrid reasoning for whole-body mobile manipulation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1828–1835, 2014.

[70] 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.

[71] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 5495–5502, 2015.

[72] 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 *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 3443–3444.

[73] K. Shankar, J. W. Burdick, and N. H. Hudson, *A Quadratic Programming Approach toQuasi-Static Whole-Body Manipulation*. Cham: Springer International Publishing, 2015, pp. 553–570. [Online]. Available: https://doi.org/10.1007/978-3-319-16595-0_32

[74] A. Escande, N. Mansard, and P.-B. Wieber, "Hierarchical quadratic programming: Fast online humanoid-robot motion generation," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.

[75] A. Reiter, A. Müller, and H. Gattringer, "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.

[76] D. M. Bodily, T. F. Allen, and M. D. Killpack, "Motion planning for mobile robots using inverse kinematics branching," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5043–5050.

[77] 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.

[78] A. Reiter, A. Müller, and H. Gattringer, "Inverse kinematics in minimum-time trajectory planning for kinematically redundant manipulators," in *IECON 2016 - 42nd Annual Conference of the IEEE Industrial Electronics Society*, Oct 2016, pp. 6873–6878.

[79] V. Falkenhahn, F. A. Bender, A. Hildebrandt, R. Neumann, and O. Sawodny, "Online tcp trajectory planning for redundant continuum manipulators using quadratic programming," in *Advanced Intelligent Mechatronics (AIM), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1163–1168.

[80] D. M. Bodily, T. F. Allen, and M. D. Killpack, "Motion planning for mobile robots using inverse kinematics branching," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 5043–5050.

[81] J. Pan, X. Zhang, and D. Manocha, "Efficient penetration depth approximation using active learning," *ACM Transactions on Graphics*, vol. 32, no. 6, 2013.

[82] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[83] 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.

[84] 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.