# One-step Trajectory Planning for Autonomous Pick and Place Tasks

Tarik Kelestemur, Philip Long, and Taskin Padir
Department of Electrical and Computer Engineering
Robotics and Intelligent Vehicles Research Lab, Northeastern University, Boston, MA 02115, USA
{kelestemur.t, p.long, t.padir}@northeastern.edu

## I. Introduction

Trajectory optimization algorithms are actively used for motion planning problems. They are not only outperforming sampling-based algorithms in terms of planning time and success rate but they also provide flexibility to augment task constraints into the motion planning problem. These methods are suitable for integrated task and motion planning such as keeping a cup of coffee upright or fixing robot's links to specified poses for given time windows along the trajectory. Zucker et al. [5] showed that motion planning problem can be formalized as a non-convex and constrained optimization problem and it can outperform well-known sampling-based methods bidirectional RRT by Kuffner and LaValle [2] and RRT* by Karaman and Frazzoli [1]. Later, Schulman et al. [4] introduced TrajOpt algorithm which improved Zucker et al. [5] work by using a numerical optimization algorithm and a different collision checking method.

One of the widely used benchmarks for task and motion planning is the autonomous pick an place task. From a systems integration perspective, it poses several challenges including 3D perception, motion planning, grasp synthesis, and control. Each of these are active research areas, however, the sequential nature of the pick and place task requires that these components should work in a integrated manner. The robot should take secondary task goals into account while planning for primary tasks. For example, Mavrakis et al. [3] showed that reasoning about the place pose to select a grasp pose can improve overall pick and place performance.

As service robots are becoming more available, there is an increasing need for the task and motion planning methods that work in cluttered and dynamic environments in a fast and robust manner. For example, one of the key challenges in developing motion planning pipelines for manipulation in domestic environments is to minimize the amount of time for which the robot remains idle. Imagine a scenario where a human is putting newly bought groceries on a table and the robot is helping her/him by placing them onto a shelf. In this case, the robot needs to keep up with the human for a harmonious human robot collaboration. In this work, we present a formulation to plan pick and place trajectories in one step and compare it with the classical two-step approach. We show how planning in one step can improve the execution time. We also present experimental results from a practical pick and place task.

Fig. 1. Experiment setup: (Left) Environment model (Right) Real world

## II. Methodology

Our development platform used in this study, is the Toyota's Human Support Robot (HSR). The HSR is a mobile service robot equipped with a 5 degrees-of-freedom (DoF) arm and a 3 DoF omni-drive base. The HSR's head has 2 DoF, one pair of stereo camera, a wide angle RGB camera and a depth sensor. The base of the HSR is modeled as two prismatic and one revolute joint and combined with the arm to get the 8-DoF whole body chain.

### A. Motion Planning

The motion planning problem can be formulated as a non-convex optimization problem and can be solved using sequential quadratic programming (SQP) by minimizing a cost function subject to inequality and equality constraints. A simple cost function is used to penalize long-length paths:

$$f(q_1 : T) = \sum_{1}^{T} ||q_{t+1} - q_t||^2 \tag{1}$$

where $q_t \in R^8$, the decision variable of the problem, describes the joint configuration at t-th time step for the 8 DoF kinematic chain. The joint limits are defined as inequality constraints: $(q_t - q^-) > 0$, and $(q^+ - q_t) > 0$, where $q^+$ and $q^-$ are maximum and minimum values, respectively. End-effector poses are defined as equality constrains: $\Delta d(q_t) = 0$ where $\Delta d(q_t)$, ordered as $[\Delta x, \Delta y, \Delta z, \Delta roll, \Delta pitch, \Delta yaw]$, is the Cartesian deviation between the end-effector pose at the robot state $q_t$ and its desired pose. Finally, Gilbert-Johnson-Keerthi algorithm is used to compute collision distance and hinge-loss function is used to set up the constraint as introduced in [4]. Once a trajectory is planned, it is smoothed using tension spline interpolation based on velocity and acceleration limits.

### B. Perception

In order to plan collision-free trajectories, a complete model of the environment needs to be generated. We used RANSAC algorithm to segment planes in the point cloud e.g., shelf racks, table. The tabletop is then clustered using Euclidean Clustering algorithm to segment individual objects. Finally, Principal Component Analysis is applied to tabletop object to get their poses. The OpenRAVE simulation platform is used to model the environment of the robot.

## III. Experiments

### A. Experimental Setup

The overall experiment is picking an object from a table and placing it onto a shelf within the robot's workspace. The experimental setup is shown in Figure 1. The pick position is selected 3cm away from the object's center towards the robot and the orientation is aligned with the object's principal axes. The place pose is selected as the middle of the top shelf rack. For a fair comparison, the environment is modeled in the beginning and used for all tests. Two cases are tested:

*1) 1. Two-step Planning:* A classical two-step approach in which two separate trajectories are planned for each of picking and placing poses. The robot executes the first plan, grasp the object, executes the second plan and places the object. For each planning 5 waypoints are used and the end-effector goals are given as pose constraint at the last waypoint to the optimization.

*2) 2. One-step Planning:* The whole task is planned in one step where 10 waypoints are used and the pick and place pose are given as pose constraints at the 5th and 10th waypoints, respectively. Since the robot will be moving at the picking waypoint, an external controller is used to check whether the end-effector is reached to picking pose by looking at the difference of current end-effect pose and goal pose. If the error is less than $10^{-4}$, the fingers are closed immediately. Each case is executed 15 times.

### B. Results

The two cases are compared in terms of planning time, total path length and the execution time. No notable improvements have been observed for the planning time (around 1 second for both cases) and the path length, however, a significant improvement has been recorded for the execution time. For one-step planning, the mean of the execution time over 15 pick-and-place tasks is found to be 20.15 seconds whereas for the two-step planning, it is 26.96 seconds. The standard deviation and mean of execution times are depicted in Figure 2. Since we are using the sum of joint displacements for each consecutive waypoints as the cost function (1), the following joint configurations after picking configuration are close to picking configuration, when it is planning for one-step. Thus, it is shorter for the robot to reach from picking configuration to placing configuration. Also, one-step planning produces a complete trajectory which results in a better velocity profiling during trajectory smoothing.

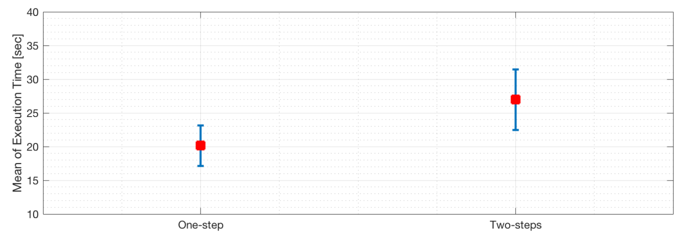


Fig. 2. Mean and standard deviation of execution times

## IV. Conclusion

This report describes how a pick and place task can be planned in one step and produce shorter execution times compared to the two-step planning. We show our experimental setup with a mobile manipulator that performs the task in the real world. Although this work focuses on sequential manipulation, since we can define arbitrary links to reach certain poses at any time along the trajectory, this method can easily be extended to augment different task constraints into motion planning problem. Therefore, the same framework can be applied to more general task and motion planning problems. One drawback of one-step planning is that it is prone to unstable grasping, we are planning to address this problem as a future work and generalize this method for more objects.

## References

[1] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

[2] James J Kuffner and Steven M LaValle. Rrt-connect: An efficient approach to single-query path planning. In *International Conference on Robotics and Automation (ICRA)*, volume 2, pages 995–1001. IEEE, 2000.

[3] Nikos Mavrakis, Marek Kopicki, Rustam Stolkin, Ales Leonardis, et al. Task-relevant grasp selection: A joint solution to planning grasps and manipulative motion trajectories. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 907–914. IEEE/RSJ, 2016.

[4] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014.

[5] Matt Zucker, Nathan Ratliff, Anca D Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M Dellin, J Andrew Bagnell, and Siddhartha S Srinivasa. Chomp: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9-10): 1164–1193, 2013.