

Manipulation Task Planning and Motion Control Using Task Relaxations

Marcos da Silva Pereira and Bruno Vilhena Adorno

Abstract—When describing high-level manipulation task specifications, it is common to discretize the workspace in multiple locations to define different possibilities for placing the manipulated objects, which usually does not scale well. Hence, we use a kinematic control law based on constrained quadratic programming to relax tasks by defining regions of interest instead of exact locations, with geometric primitives represented by dual quaternions, in conjunction with a multi-layered framework for task and motion planning already available in the literature. The framework consists of a high-level planner, which generates task plans for linear temporal logic (LTL) specifications, and of a low-level motion controller, which executes the task.

I. INTRODUCTION

The goal of manipulation task planning is to automatically generate feasible movement sequences to manipulate objects in order to place them with a desired pose in the workspace. In addition, it is necessary to execute the planned actions, which requires the integration of task and motion planning (ITMP) [1]. Since the high dimensionality of the manipulation problem and the complexity of specifying the tasks make the problem computationally demanding, He *et al.* [2] propose a multi-layered manipulation framework that first generates task plans for pre-grasping and placing of objects between locations of interest. Then, the task plan is transformed to a motion plan by a sampling-based motion planner, which is then executed by a low-level motion controller. However, not every manipulation task can be efficiently described using only one location [3]. Thus, a common approach is to discretize the workspace in multiple locations to define different possibilities for placing the manipulated objects, but this approach does not scale well.

This work proposes the use of a constraint-based motion controller [4], which allows the definition of regions of interest instead of discrete poses, which greatly reduces the computational burden of the overall method, with the additional advantages that there is no need of motion planning and the system is reactive.

II. METHODOLOGY

We use a multi-layered framework [2] to solve the ITMP problem. First, a task φ is specified using linear temporal logic (LTL) [5]. Then, φ is converted into a deterministic finite automaton (DFA) \mathcal{A}_φ that specifies all the ways the robot can execute the task. In addition, an abstraction \mathcal{R} , which

captures all the ways the robot can manipulate the objects, is defined and then combined to \mathcal{A}_φ into a product graph \mathcal{P} that represents how the robot can move the objects to achieve the specified task. Next, we use Dijkstra's algorithm to search for a path on the graph from an initial node to a desired one. Finally, the high-level plan is executed by a task-space constraint-based kinematic controller [6], which is based on an optimization problem that minimizes the joint velocities, $\dot{\mathbf{q}} \in \mathbb{R}^n$, in the ℓ_2 -norm sense while respecting hard constraints, such as obstacles in the workspace, joint limits, etc.

Given a desired task pose $\mathbf{x}_d \in \mathbb{R}^m$, where $\dot{\mathbf{x}}_d = \mathbf{0}$, $\forall t$, the task error $\tilde{\mathbf{x}} \triangleq \mathbf{x} - \mathbf{x}_d$, and a gain $\eta \in (0, \infty)$, the control input \mathbf{u} is obtained as

$$\begin{aligned} \mathbf{u} \in \underset{\dot{\mathbf{q}}}{\operatorname{argmin}} \quad & \|\mathbf{J}\dot{\mathbf{q}} + \eta\tilde{\mathbf{x}}\|_2^2 + \lambda \|\dot{\mathbf{q}}\|_2^2 \\ \text{subject to} \quad & \mathbf{W}\dot{\mathbf{q}} \leq \mathbf{w} \end{aligned} \quad (1)$$

where $\mathbf{J} \in \mathbb{R}^{m \times n}$ is the task Jacobian, $\lambda \in [0, \infty)$ is a damping factor and $\mathbf{W} \in \mathbb{R}^{l \times n}$ and $\mathbf{w} \in \mathbb{R}^l$ are used to impose linear constraints in the control inputs. Since we use the ℓ_2 -norm (i.e., the commonly used Euclidean norm), the control signal is obtained from a local optimization problem. Furthermore, when there are no constraints, the control input generated by the optimization problem (1) is equivalent to the kinematic controller based on the damped least-square inverse (also called singularity-robust inverse) [7].

In order to prevent collisions with the workspace, we use the Vector Field Inequalities (VFI) framework [4], which requires distance functions between two collidable entities and the corresponding Jacobian matrices. We use three plane constraints to prevent end effector collisions with walls in the workspace and four lateral planes to constrain the end effector inside a region of interest—the relaxed task region. Similar to the work of Quiroz-Omaña and Adorno [6], these seven constraints can be written as

$$-\mathbf{J}_{p,n_{\pi_i}}\dot{\mathbf{q}} \leq \eta\tilde{d}_{p,n_{\pi_i}}, \quad (2)$$

where $i \in \{1, 2, \dots, 7\}$ and $\tilde{d}_{p,n_{\pi_i}} = d_{p,n_{\pi_i}} - d_{\pi,\text{safe}}$, with $d_{\pi,\text{safe}}$ being the safe distance to each plane and $d_{p,n_{\pi_i}}$ and $\mathbf{J}_{p,n_{\pi_i}}$ are the point-static-plane distance and its Jacobian, respectively [4]. In addition to the plane constraints, we use conic constraints to prevent saturation of actuators [6].

The VFI approach enables to define desired regions instead of specific poses in the low-level planner, eliminating the need of task-space discretization. As a result, the high-level planner becomes less computationally demanding. The number of valid nodes in the abstraction \mathcal{R} is given by $2(|\mathcal{L}|+1)P_{|\mathcal{O}|}^{(|\mathcal{L}|+1)}$

M. S. Pereira is with the Graduate Program of Electrical Engineering, Federal University of Minas Gerais (UFMG), Belo Horizonte, 31270-901, Belo Horizonte-MG, Brazil, e-mail: marcos-si-pereira@ufmg.br. B.V. Adorno is with the Department of Electrical Engineering - Federal University of Minas Gerais (UFMG) - Av. Antônio Carlos 6627, 31270-901, Belo Horizonte-MG, Brazil. Email: adorno@ufmg.br.

TABLE I

PLANNING DATA FOR φ_1 , φ_2 AND φ_3 . $|\mathcal{L}|$ IS THE NUMBER OF LOCATIONS, $|\mathcal{R}|$ IS THE TOTAL NUMBER OF NODES IN \mathcal{R} , $|\mathcal{R}_{\text{VALID}}|$ IS THE NUMBER OF VALID NODES IN \mathcal{R} , $|\mathcal{P}|$ IS THE NUMBER OF NODES IN \mathcal{P} AND T_{PLANNING} IS THE TOTAL HIGH-LEVEL PLANNING TIME.

Spec. φ	$ \mathcal{L} $	$ \mathcal{R} $	$ \mathcal{R}_{\text{VALID}} $	$ \mathcal{P} $	$T_{\text{PLANNING}}(\text{ms})$
VFI framework					
φ_1	3	128	24	512	470
φ_2	3	768	48	3072	20612
φ_3	3	768	48	2304	8742
Discretization of the task space					
φ_1	6	392	84	1568	3788
φ_2	12	26364	3432	105456	—
φ_3	12	26364	3432	79092	—

[2], where $|\mathcal{L}|$ is the number of locations, $|O|$ is the number of objects and P_n^k is the k -permutation from n elements. Therefore, the size of the product graph \mathcal{P} grows rapidly with the increase in the number of objects and locations and also with the size of the specification.

III. SIMULATIONS & DISCUSSION

To exemplify the decrease of the size of the product graph \mathcal{P} when using the VFI to define regions of interest for the low-level motion controller, we implemented the product graph \mathcal{P} proposed by He et al. [2] in C++ with the Boost Graph Library.¹ The task φ is processed using Spot [8] and we performed simulations on V-REP² using ROS.³ Furthermore, we used the DQ Robotics library [9] for robot modeling and control and to define the geometrical constraints, and constrained convex optimization was implemented using IBM ILOG CPLEX Optimization Studio.⁴

First, we defined the task φ_1 to pick an object from one location and place it on a region of interest on a table. Then we defined the task φ_2 , which consists in swapping the location of objects 1 and 2. Finally, we defined the task φ_3 , which swaps the location of objects 1 and 2 but first placing object 2 on location 2, and only then placing object 1 on location 1. Then we discretized each location into four smaller locations. In the case of φ_1 , only one location was discretized into four smaller locations. Table I shows that even a coarse discretization of the task space greatly increases the size of the graphs and, consequently, the planning time. Therefore, since after 30 minutes the planner did not generate a plan for tasks 2 and 3, the process was manually aborted. Lastly, Fig. 1 illustrates the use of planes to define the region of interest for φ_1 .

IV. CONCLUSION

This work modified the manipulation framework proposed by He et al. [2], which uses sampling-based motion planners as the low-level planner, to use a kinematic control law based on constrained quadratic programming to relax tasks and eliminate the need of a low level motion planner. As a result, there is no need to discretize the task space. This approach

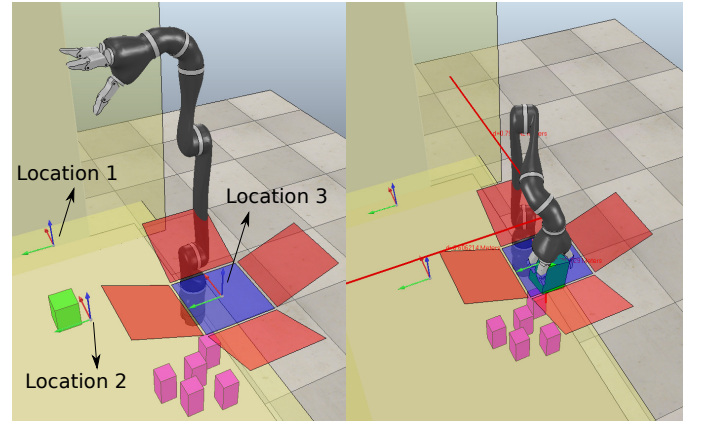


Fig. 1. Scene for φ_1 . On the left, the green cuboid is the object to be manipulated, the pink cuboids are already in the region of interest. The yellow planes are the plane constraints used to prevent collision with the environment. The three possible locations for objects are shown. The red plane constraints define the region of interest indicated by the blue plane. On the right, the manipulator is placing the cuboid on the region of interest.

reduced the number of locations of interest needed in the manipulation framework. Therefore, the number of nodes in the high-level planner graph was kept much lower than in the original framework. We evaluated this approach with three pick-and-place tasks and showed the difference in the number of planning nodes generated. Future works will optimize the construction of the high-level planner graph to contain only the valid nodes for the task to reduce the computational complexity.

REFERENCES

- [1] E. Erdem, K. Haspalamutgil, C. Palaz, V. Patoglu, and T. Uras, "Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, may 2011, pp. 4575–4581.
- [2] K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Towards manipulation planning with temporal logic specifications," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may 2015, pp. 346–352.
- [3] S. Alartsev and F. Ortmeier, "Improving the sequence of robotic tasks with freedom of execution," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2014, pp. 4503–4510.
- [4] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, "Dynamic Active Constraints for Surgical Robots Using Vector-Field Inequalities," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1166–1185, oct 2019.
- [5] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model checking*. MIT Press, 1999.
- [6] J. J. Quiroz-Omana and B. V. Adorno, "Whole-Body Control With (Self) Collision Avoidance Using Vector Field Inequalities," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4048–4053, oct 2019.
- [7] Y. Nakamura and H. Hanafusa, "Inverse Kinematic Solutions With Singularity Robustness for Robot Manipulator Control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 108, no. 3, p. 163, 1986.
- [8] A. Duret-Lutz and D. Poitrenaud, "SPOT: an extensible model checking library using transition-based generalized buchi automata," in *The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004. (MASCOTS 2004). Proceedings*. IEEE, 2004, pp. 76–83.
- [9] B. V. Adorno and M. M. Marinho, "DQ Robotics : a library for robot modeling and control using dual quaternion algebra," preprint available at <https://arxiv.org/abs/1910.11612>.

¹https://www.boost.org/doc/libs/1_71_0/libs/graph/doc/index.html

²<http://www.coppeliarobotics.com/>

³<https://www.ros.org/>

⁴<https://www.ibm.com/products/ilog-cplex-optimization-studio>