

Hierarchical Task and Motion Planning using Logic-Geometric Programming (HLGP)

Danny Driess

Ozgur S. Oguz

Marc Toussaint

Machine Learning & Robotics Lab, University of Stuttgart, Germany

Max Planck Institute for Intelligent Systems, Stuttgart, Germany

Abstract—In this work, we present a hierarchical approach to task and motion planning (TAMP) within an optimization-based framework. Recent work on formulating TAMP as a logic enhanced nonlinear program has shown remarkable capabilities. However, scaling this approach to domains with many discrete decisions or longer horizons implies a computational bottleneck. To overcome this, we introduce hierarchies within this framework, where on coarser levels a problem with less discrete decisions is solved. Formally, the hierarchies are defined in a way that the resulting nonlinear programs on coarser hierarchy levels are lower bounds on the finer hierarchies. We demonstrate the generality of the approach for both a bi-manual manipulation task and a mobile manipulation scenario which includes a “worm” like walking robot.

I. INTRODUCTION

In joint Task and Motion Planning (TAMP), it is common to introduce discrete abstractions for the purpose of reasoning over high level decisions. Given such a discrete decision sequence, a geometric planner must figure out the continuous motions of a robot to fulfill the planned tasks. Logic-Geometric Programming (LGP) is one variant to formalize the relation between discrete decision variables and a continuous nonlinear mathematical program (NLP) that represents the underlying continuous world problem [2]. The purpose of the discrete decisions in LGP is to simplify the underlying mathematical problem – ideally by enumerating sub-problems that can be handled efficiently by a continuous NLP solver. One core property of LGP is that the resulting NLP optimizes for a globally consistent continuous motion sequence conditioned on the discrete decisions. This is important, since, for example, the way an object is grasped greatly influences how it can be placed or used as a tool.

However, when scaling this approach to longer horizons or situations in which there are many possible discrete decisions such as for grasping, the computational demand for solving many NLPs during the search greatly increases. To overcome this, we propose to introduce a hierarchy of LGPs at the discrete decision level for the purpose of providing efficient lower bounds of the underlying LGP problem.

In our approach, a coarse level LGP is solved first that contains less constraints in the underlying NLP. The discrete decisions of this solution conditions a fine level LGP in two respects: Within the coarse LGP they imply an NLP which checks geometric and kinematic feasibility of these coarse decisions, potentially making simplifying assumptions about the kinematics. On the fine level LGP, the decisions of the

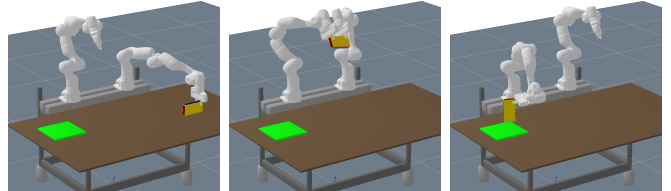


Fig. 1. Box re-orientation task that requires both arms for the solution

coarse level imply constraints on the allowed sequences of decisions of the real manipulation problem. The associated NLP of the fine LGP then solves for a globally consistent motion plan, which is in contrast to previous notions of abstractions in hierarchical TAMP [1], since in LGP feasible abstract decisions are not guaranteed to be refinable.

In Sec. III-A, we demonstrate the method and its computational benefit on a bi-manual manipulation task that involves grasping, re-orientation, handover, and placing. The planning on the fine level plans a motion that includes the detailed grasping geometry of a box shaped object. In Sec. III-B, we consider a novel domain where a “worm” robot can use its two end-effectors to walk, climb and pick-and-place objects, showing interesting sequential manipulation capabilities.

II. HIERARCHICAL LOGIC GEOMETRIC PROGRAMMING

We start by restating the LGP framework of [2] in a form that will allow us to introduce a hierarchy. The goal is to find a global path x in the configuration space as the solution of a logic enhanced NLP, which solves a given TAMP problem, while fulfilling plausibility constraints. The path itself consists of K subpaths $x_k : [T_{k-1}, T_k] \rightarrow \mathcal{X}_k$, $x_k \in C^2$. These K different phases are referred to as kinematic modes, in which the resulting path is smooth. The set $\mathcal{X}_k \subset \mathbb{R}^{l_k} \times SE(3)^{o_k}$, $\dim(\mathcal{X}_k) = n_k$ denotes the n_k -dimensional configuration space of all objects and articulated structures in mode k . The central mathematical object to define the objectives and constraints are so-called feature maps $\phi : \mathcal{X} \times \mathbb{R}^{n_k} \times \mathbb{R}^{n_k} \rightarrow \mathbb{R}^{d_k}$, $\phi \in C^1$, which map the world configuration (and their derivatives) to a d_k -dimensional space like positions of robot links, distances between objects, object poses, etc. The sets $\mathcal{T}_{\text{sos,ineq,eq}}(s_k)$ of active objectives and constraints at a certain phase k of the motion are determined by a discrete decision variable s_k . The possible transitions $s_k \in \text{succ}(s_{k-1})$ between such discrete states from s_{k-1} to s_k is defined by a first order

TABLE I
RUNTIMES IN SECONDS TO FIND THE FIRST FEASIBLE SOLUTION

scenario	no hierarchy	with hierarchy
box re-orientation		
place side 1	1035	122
place side 2	1643	202
place side 3	1596	230
place side 4	750	136
mobile manipulation	92	34

logic language. These considerations lead to the LGP:

$$\min_{K, \{x_k, s_k\}_{k=1}^K} \sum_{k=1}^K \int_{T_{k-1}}^{T_k} \sum_{\phi \in \mathcal{T}_{\text{sos}}(s_k)} \|\phi(x_k(t), \dot{x}_k(t), \ddot{x}_k(t))\|_2^2 dt \quad (1a)$$

$$\text{s.t. } \forall_{k=1, \dots, K} : \quad x_0(0) = \tilde{x}_0, \quad s_0 = \tilde{s}_0 \quad (1b)$$

$$\forall_{\phi \in \mathcal{T}_{\text{eq}}(s_k)} \forall_{t \in [T_{k-1}, T_k]} : \phi(x_k(t), \dot{x}_k(t)) = 0 \quad (1c)$$

$$\forall_{\phi \in \mathcal{T}_{\text{ineq}}(s_k)} \forall_{t \in [T_{k-1}, T_k]} : \phi(x_k(t), \dot{x}_k(t)) \leq 0 \quad (1d)$$

$$h_k(x_{k-1}(T_{k-1}), x_k(T_k), \dot{x}_{k-1}(T_{k-1}), \dot{x}_k(T_k), s_k) = 0 \quad (1e)$$

$$s_k \in \text{succ}(s_{k-1}) \quad (1f)$$

$$s_K \in S_{\text{goal}}, \quad \phi_{\text{goal}}(x_K(T_K)) = y_{\text{goal}} \quad (1g)$$

h_k represents transitional conditions between the subpaths and (1g) defines the symbolic and (optional) continuous goal state. The purpose of the logic is not only to encode high level reasoning like pick and place, but also physical plausibility. The idea to introduce a hierarchy is to have a coarse LGP and a fine LGP. Solving (1) for a logic on a coarse level gives (probably multiple) feasible sequences $\{g_j\}_{j=1}^J$ of discrete decisions. These coarse decisions imply constraints on the allowed decision sequences on the fine level. This leads to the LGP conditioned on $\{g_j\}_{j=1}^J$

$$\min_{K, \{x_k, s_k\}_{k=1}^K} \sum_{k=1}^K \int_{T_{k-1}}^{T_k} \sum_{\phi \in \mathcal{T}_{\text{sos}}(s_k)} \|\phi(x_k(t), \dot{x}_k(t), \ddot{x}_k(t))\|_2^2 dt \quad (2a)$$

$$\text{s.t. } \forall_{k=1, \dots, K} (1b), (1c), (1d), (1e), j_0 = 1$$

$$s_k \in \text{succ}(s_{k-1}, g_{j_{k-1}}) \quad (2b)$$

$$j_k = j_{k-1} + \text{subgoalTrans}(g_{j_{k-1}}, s_{k-1}) \quad (2c)$$

$$j_K = J, \quad (2d)$$

where $\text{subgoalTrans}(g_{j_k}, s_k) = 1$ if s_k reached the subgoal state g_{j_k} , 0 otherwise. The logic on the fine level contains preconditions that depend on g_{j_k} . The coarse decisions g_{j_k} can therefore be seen as symbolic subgoals and the index j_k as a subgoal counter. Since $K \geq J$, the fine level can have much more decisions. In order to simplify the problem on a coarser level while being informative for the search in the finer level, we want the coarser level to be a lower bound on the fine level, which is expressed by the following theorem.

Theorem 1: If the set of active features on the coarser level is contained in the finer level, i.e.

$$\mathcal{T}_{\text{sos,eq,ineq}}(g_{j_k}) \subset \bigcup_{s_k \in \text{succ}(s_{k-1}, g_{j_{k-1}})} \mathcal{T}_{\text{sos,eq,ineq}}(s_k) \quad (3)$$

then feasibility of the coarser level is a necessary condition for the feasibility on the finer level. Furthermore, a solution of a coarser level is a lower bound on the costs of a finer level.

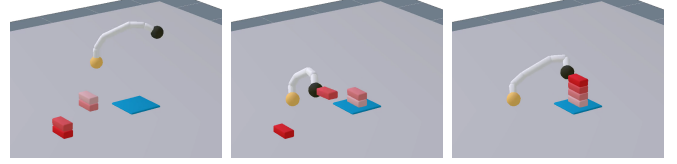


Fig. 2. Box stacking with walking robot

Proof: Clear by construction of (3) and the non-negative additive terms in the objective (2a). ■

By constraining on a subgoal sequence, the combinatorial complexity on the finer level is greatly reduced. Here, we simply alternate between searching on coarse and finer levels.

III. EXPERIMENTS

A. Bi-Manual Box Re-Orientation

In this experiment, we consider the task of placing a box on a target location on a specific side of the box (Fig. 1). The initial position of the box is out of reach of the robot arm that could place it on the target (green square). Therefore, the two arms have to collaborate with each other to solve the task. In this specific experiment, we consider 6 different discrete placement decisions and 8 different grasping decisions. On the coarse level, only the placement on the different sides is part of the decision, while it is assumed that an object can be grasped whenever the end-effector can touch the object. The grippers itself are not part of the collision model on the coarse level. Given a feasible sequence of the coarse level, the problem on the fine level then additionally figures out how the object can be grasped, i.e. from which side and how the grippers are oriented. Tab. I shows the runtimes in seconds for finding the first feasible solution for tasks where the box should be placed on 4 different sides at the green target location. The reported values with the hierarchical approach is the total time for solving on both hierarchy levels. With the hierarchy, the solution is computed between 5.5 and 8.5 times quicker.

B. Mobile Manipulation with Worm Robot

In a second experiment a worm-like walking robot has to stack objects on top of each other in a certain order (Fig. 2). On the coarse level, the robot is represented as a free floating robot which proposes the pick and place sequence for the fine level, where the walking behavior is taken into account as well. Tab. I shows the runtime for this scenario.

IV. CONCLUSION

The hierarchy turned out to greatly reduce the computation time in our experiments. However, theoretically, the worst-case complexity is still the same as without a hierarchy.

REFERENCES

- [1] L. P. Kaelbling and T. Lozano-Perez. hierarchical task and motion planning in the now. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*, 2011.
- [2] Marc Toussaint, Kelsey R. Allen, Kevin A. Smith, and Josh B. Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. In *Proc. of Robotics: Science and Systems (R:SS)*, 2018.