Integrated Task and Motion Planning Using Physics-based Heuristics

Aliakbar Akbari, Muhayyuddin, and Jan Rosell
Institute of Industrial and Control Engineering (IOC), Universitat Politècnica de Catalunya (UPC), Barcelona, jan.rosell@upc.edu

***Abstract*—This work presents a knowledge-based task and motion planning framework based on a version of the Fast-Forward task planner. A reasoning process on symbolic literals in terms of knowledge and geometric information about the workspace, together with the use of a physics-based motion planner, is proposed to evaluate the applicability and feasibility of manipulation actions and to compute the heuristic values that guide the search. The proposal results in low-cost physically-feasible plans.**

I. Introduction

Mobile manipulation planning requires, at task level, the finding of a sequence of actions, and at motion level, the finding of the way to execute them. Therefore, the efficient combination of both planning levels has currently emerged as a substantial challenge. In this line, some approaches like [8, 9, 21, 18] considered various mechanisms to interface between the symbolic and the geometric reasoning processes. Other approaches look for hierarchical planning solutions based on hierarchical planning, such as [14, 17, 6, 11, 5], that evaluate task-level decisions with low-level geometric-reasoning modules. Some other approaches like [4, 10] propose different ways to integrate geometric information within the Fast-Forward (FF) task planner [13] or use task planning based on linear temporal logic (LTL) like [15, 12]. Finally, some other approaches use the GraphPlan task planning algorithm [3], like [1, 2]. These approaches used physics-based motion planning to evaluate the feasibility of all potential actions in the planning graph and, complementarily, incorporated knowledge in terms of ontologies, which has previously been proposed as a useful way to incorporate knowledge in solving human-like tasks [23, 7].

Contributions We present a new framework, based on the FF planner, for the efficient combination of knowledge-based task planning and physics-based motion planning guided by reasoning process. The reasoning process is performed on symbolic literals in terms of knowledge and geometric information about the workspace (offline reasoning), as well as using high-level reasoning and physics-based motion planning to determine the applicability of actions and to determine the feasibility of applicable actions along their effects (online reasoning). As a consequence, the proposed method is able to prevent or discard some unnecessary actions while planning. Moreover, the computation of the heuristic cost that guides the search of the solution plan takes into account the physical properties of objects and the actual cost of selected actions computed by a physics-based motion planner. The proposal, therefore, aims to make the planning more efficient and to find physically-feasible low-cost plans.

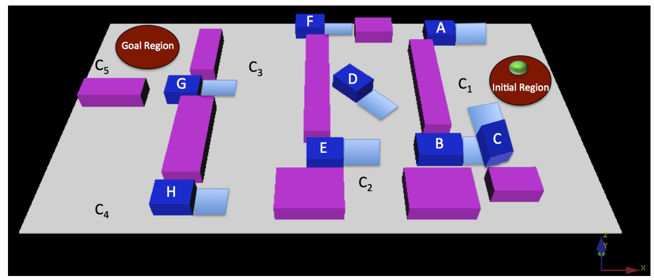

Fig. 1: Manipulation problem: the robot (green sphere) has to move from the initial to the goal region among fixed and manipulatable obstacles (labeled according to their weight in a decreasing order). Workspace regions are labeled with the names of the associated configuration space disconnected regions.

II. Problem Formulation

Consider a manipulation problem like that shown in Fig. 1 where a mobile robot is required to move from an initial region towards a goal one in an indoor environment cluttered with obstacles that can be either fixed or manipulatable. The existence of manipulatable obstacles may partition the free subspace of the configuration space of the robot, C_{free}, into disconnected regions C_i. If the initial and the goal regions of the query to be solved lie in different disconnected regions, then the robot will need to move some manipulatable obstacles away in order to connect them and find a solution path. It is assumed that the robot is able to perform three type of actions: *push* and *pull* actions to change the position of manipulatable objects, and the *transit* action to move freely along a collision-free path. It is also assumed that there are two types of obstacles in the environment, *fixed obstacles* which the robot must avoid, and *manipulatable obstacles* (*MObs*), labeled from A to H, that can be pushed or pulled by the robot along a given direction. In order to interact with a *MObs*, the robot must be located in the corresponding manipulatable region *MRgn* (highlighted in light blue).

Moreover, *MObs* have different physical features and some of them, like object A, may be beyond the robot manipulation capacity. Also, it must be noted that there can be some actions which do not provide fruitful effects to solve the problem. For

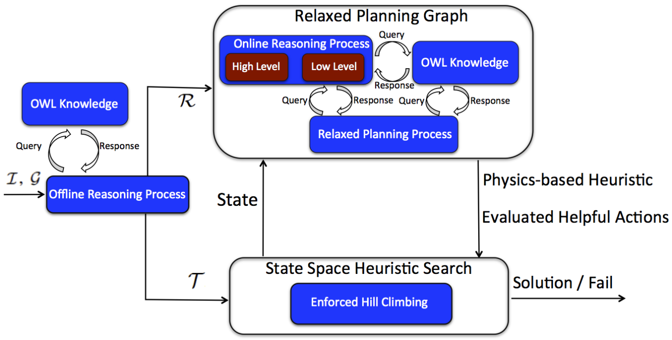

Fig. 2: Solution Overview

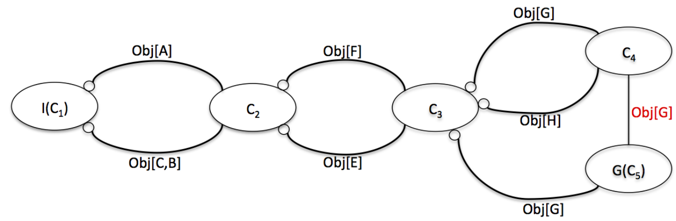

Fig. 3: Disjoint components of C_{free}. Edges are labeled with the critical objects whose removal changes the connectivity. Small circles at the end of the edges illustrate from where the objects can be manipulated.

example, pulling object F does not provide the access from C_2 towards C_3. Finally, note that a number of potential possible plans may exist and the least-cost one is the one sought. These aforementioned issues pose interesting challenges that can be properly solved by considering an efficient combination of task and physics-based motion planning.

Let a manipulation planning problem $\mathcal{T}$ be defined as the tuple $\mathcal{T} = \langle \mathcal{I}, \mathcal{G}, \mathcal{K} \rangle$ where $\mathcal{I}$ and $\mathcal{G}$ are, respectively, the initial and goal states, and $\mathcal{K}$ is the ontological knowledge.

A state is the tuple $S = \langle L, \mathcal{W} \rangle$ comprising a conjunction of literals L and the geometric information of the workspace $\mathcal{W}$. A state changes when an action is applied. An action a can be defined by a tuple $a = \langle$*name, pre, effect*$^+$, *effect*$^-$, $\mathcal{Q}\rangle$ where: *name* is a symbolic name; *pre* is preconditions; *effect*$^+$ and *effect*$^-$ are, respectively, positive and negative effects of the action; and $\mathcal{Q}$ is a query to a physics-based motion planner acting on $\mathcal{W}$, that computes a path and its actual cost, and returns the new state of the workspace. However, $\mathcal{Q}$ is called once required and not for all actions. For a given action a, the literals defining the successor state are computed as:

$$Succ(S.L, a) = S.L \cup \mathit{effect}^+(a) \backslash \mathit{effect}^-(a)$$

The geometric information $\mathcal{W}$ is updated with $\mathcal{Q}$, or left unchanged if $\mathcal{Q}$ is not called.

The following literals are used to define states, being some of them evaluated based on a reasoning process:

- *HasAcc(FromRgn, ToRgn)*: Captures the result of geometric reasoning and evaluates to true if a trajectory may exist for the robot to move between *FromRgn* and *ToRgn*.
- *At(Robot, Rgn)*: Holds if *Robot* has reached region *Rgn*.
- *IsCritical(MObs)*: Holds if *MObs* is a manipulatable obstacle whose removal makes two disjoint configuration space regions to be connected.
- *Located(MObs, Position)*: Holds if *MObs* is located at *Position* after displacement.
- *IsManipulatable(MObs)*: Informs whether *MObs* is a manipulatable obstacle.

The three type of actions considered and their preconditions and positive and negative effects are:

Transit(Robot, FromRgn, ToRgn):
Pre: *At(Robot, FromRgn), HasAcc(FromRgn, ToRgn)*
Add: *At(Robot, ToRgn)*
Delete: *At(Robot, FromRgn)*
Push/Pull(Robot, MObs, FromPos, ToPos, MRgn, ToRgn):
Pre: *At(Robot, MRgn), IsManipulatable(MObs), IsCritical(MObs), Located(MObs, FromPos)*
Add: *Located(MObs, ToPos), HasAcc(MRgn, ToRgn)*
Delete: *Located(MObs, FromPos), IsCritical(MObs)*

III. Proposed Solution

In order to solve the aforementioned problem, physics-based heuristics manipulation planning using knowledge is proposed as illustrated in Fig. 2, based on the Fast Forward task planning procedure (FF). The FF planning procedure does an heuristic search in the state space, where the heuristic used to estimate the cost to reach the goal from the state being evaluated is done using the *Relaxed Planning Graph* (RPG), which is a version of the Planning Graph that does not consider the negative effects. The selection of the next state in the exploration is done with *Enforced Hill Climbing* (EHC).

The variant of FF that we propose uses, on the one hand, an offline reasoning process to incorporate knowledge to the manipulation problem, related to the workspace and to the manipulation of objects. Knowledge is represented by ontologies using the Web Ontology Language (OWL). On the other hand, the proposal considers an online reasoning including high-level and low-level reasoning process.

- The *offline reasoning process* is responsible of using the knowledge to set the literals defining the initial state, as well as to build a graph, called $\mathcal{R}$, to define the connectivity of the workspace and that will be used for the online reasoning process (Fig. 3).
- The *Relaxed Planning Graph* module contains the procedure to build the RPG, extract the relaxed plan and compute the heuristic value. The construction of the RPG uses an online high-level reasoning process, based on $\mathcal{R}$, that selects the potential applicable actions and considers costs of actions based on the physical properties of the objects to compute literals cost. Once the RPG is built, the plan is extracted using the standard backward procedure considering the cheapest actions but enhanced with calls to the physics-based motion planner in order to accept only those that are feasible (for the push and pull actions the feasibility also includes the verification of the accessibility

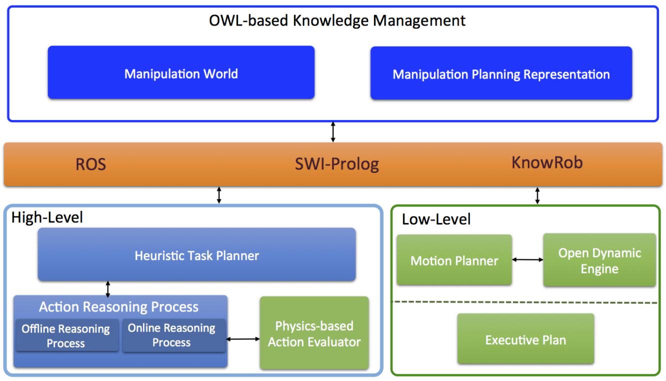

Fig. 4: Implementation framework

of the robot to the required regions after their execution). Once the plan is found, the heuristic value is set to the sum of the costs of the actions appearing in the plan, which have been obtained from the physics-based motion planner.

- The *State Space Heuristic Search* module gets $\mathcal{T}$ as input and returns either a solution plan of manipulation actions or reports failure. It keeps exploring the states using the EHC strategy and calling, for each state being explored, the RPG module to estimate the physics-based heuristic cost to reach the goal, as well as the feasible helpful actions to follow.

The completeness of the proposal is the same as that of the original FF planner, that depends on the EHC process. The FF is complete on tasks in which no fatally wrong decision is made in the EHC as it cannot take this decision back. This reason motivates us to consider physics-based motion planning where the heuristic value is computed aiming to find appropriate helpful actions. Therefore, the result of motion planning is used to guide the EHC process that significantly effects to the selection process of helpful actions from the current state.

IV. Implementation

The proposed framework implementation consists of three major layers as depicted in Fig. 4: ontological knowledge, task-level and motion-level layers. The knowledge layer is coded in the form of an OWL ontology and it consists of: a) Knowledge about the manipulation world, i.e. information on the type of objects (manipulatable or fixed), manipulatable regions, their poses (position and orientation), and physical properties (objects masses, friction coefficients, etc.). b) Manipulation knowledge representation involving all requirements for task planning (such as actions and their conditions). The main purpose of representing the knowledge as the form of OWL is to provide it on the world-wide accessible database that can be shared by multiple systems.

The task-level layer embraces: a) Heuristic task planner including the modified FF planner (implemented using the Prolog language); b) Action reasoning process whose purpose is to determine actions conditions by calling online along offline reasoning process; c) Physics-based action evaluator

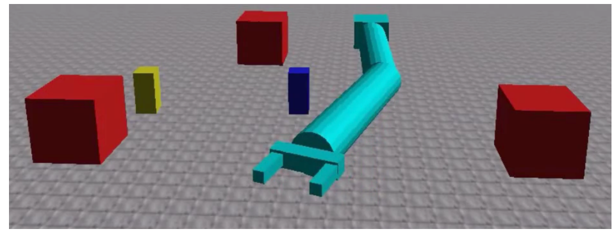

Fig. 5: Manipulation example involving a kinematic chain. The solution involves a push motion to remove the blue object followed by a move object to reach the target yellow object.

integrated with the online reasoning process that calls the motion planner when required. To fetch the OWL knowledge for task planning, the Knowrob software which enables the access of stored knowledge by Prolog predicates is employed. All task planning components are encapsulated inside the ROS environment [19] using SWI-Prolog library.

The motion-level layer is encapsulated as a ROS service that comprises The Kautham Project (https://sir.upc.edu/projects/kautham/) [20] that enables to plan under kinodynamic and physics-based constraints. It uses Open Motion Planning Library (OMPL) [22] as its core of planning algorithms, and is integrated with the Open Dynamic Engine (ODE) for the dynamic simulations [16].

V. Simulation Results

The proposal has been applied to the example of Fig. 1 and the solution sequence of the cheapest feasible actions is: *MoveC, PullC, MoveB, PullB, MoveF, PushF, MoveG, PullG, MoveGoal*. Also, a manipulation problem including the planar chain 2D robot shown in Fig. 5 has been considered. In this example, the goal is to grasp the yellow object while avoiding the fixed objects (shown in red). Since there is no direct solution, the robot pushes the manipulatable object (shown in blue) in order to reach the goal. The solutions can be visualized in https://sir.upc.edu/projects/kautham/videos/manip.mp4.

VI. Conclusion

A manipulation framework that interleaves knowledge-based task planning with physics-based motion planning has been presented. The approach is based on a version of the FF task planner, and to determine the applicability and the feasibility of actions and be able to heuristically guide the search in an efficient way, different types of reasoning processes (online and offline) integrated with ontological knowledge, the geometry of workspace, and physics-based motion planning have been proposed. The framework has been implemented and simulated for two manipulation problems involving a mobile robot and a planar 2D manipulator. The proposal is able to find physically-feasible low-cost plans.

Acknowledgments

This work was partially supported by the projects DPI2013-40882-P and DPI2014-57757-R and by FI-DGR 2014 and FPI 2015 PhD. grants.

REFERENCES

[1] Aliakbar Akbari, Jan Rosell, et al. Task and motion planning using physics-based reasoning. In *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*, pages 1–7. IEEE, 2015.

[2] Aliakbar Akbari, Muhayyuddin Gillani, and Jan Rosell. Reasoning-based evaluation of manipulation actions for efficient task planning. In *Robot 2015: Second Iberian Robotics Conference*, pages 69–80. Springer, 2016.

[3] Avrim L Blum and Merrick L Furst. Fast planning through planning graph analysis. *Artificial intelligence*, 90(1):281–300, 1997.

[4] Stephane Cambon, Rachid Alami, and Fabien Gravot. A hybrid approach to intricate motion, manipulation and task planning. *The International Journal of Robotics Research*, 28(1):104–126, 2009.

[5] Lakdeepal de Silva, Akhilesh Kumar Pandey, and Rachid Alami. An interface for interleaved symbolic-geometric planning and backtracking. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 232–239. IEEE, 2013.

[6] Lavindra de Silva, Amit Kumar Pandey, Mamoun Gharbi, and Rachid Alami. Towards combining HTN planning and geometric task planning. *arXiv preprint arXiv:1307.1482*, 2013.

[7] Daniel Di Marco, Paul Levi, Rob Janssen, René van de Molengraft, and Alexander Perzylo. A deliberation layer for instantiating robot execution plans from abstract task descriptions. In *Proceedings of the International Conference on Automated Planning and Scheduling: Workshop on Planning and Robotics (PlanRob). AAAI Press*, 2013.

[8] Christian Dornhege, Patrick Eyerich, Thomas Keller, Sebastian Trüg, Michael Brenner, and Bernhard Nebel. Semantic attachments for domain-independent planning systems. In *Towards Service Robots for Everyday Environments*, pages 99–115. Springer, 2012.

[9] Esra Erdem, Kadir Haspalamutgil, Can Palaz, Volkan Patoglu, and Tansel Uras. Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4575–4581. IEEE, 2011.

[10] Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. FFrob: An efficient heuristic for task and motion planning. In *Algorithmic Foundations of Robotics XI*, pages 179–195. Springer, 2015.

[11] Mamoun Gharbi, Raphael Lallement, and Rachid Alami. Combining symbolic and geometric planning to synthesize human-aware plans: toward more efficient combined search. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 6360–6365. IEEE, 2015.

[12] Keliang He, Morteza Lahijanian, Lydia E Kavraki, and Moshe Y Vardi. Towards manipulation planning with temporal logic specifications. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 346–352. IEEE, 2015.

[13] Jörg Hoffmann and Bernhard Nebel. The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, pages 253–302, 2001.

[14] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1470–1477. IEEE, 2011.

[15] Savvas G Loizou and Kostas J Kyriakopoulos. Automated planning of motion tasks for multi-robot systems. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 78–83. IEEE, 2005.

[16] Muhayyudin, Aliakbar Akbari, and Jan Rosell. Ontological physics-based motion planning for manipulation. In *Proc. of IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2015.

[17] Amit Kumar Pandey, Jean-Philippe Saut, Daniel Sidobre, and Rachid Alami. Towards planning human-robot interactive manipulation tasks: Task dependent and human oriented autonomous selection of grasp and placement. In *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, pages 1371–1376. IEEE, 2012.

[18] Erion Plaku and Gregory D Hager. Sampling-based motion and symbolic action planning with geometric and differential constraints. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 5002–5008. IEEE, 2010.

[19] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, volume 3, page 5, 2009.

[20] Jan Rosell, Alexander Pérez, Akbari Aliakbar, Muhayyuddin, Leopold Palomo, and Néstor García. The Kautham Project: A teaching and research tool for robot motion planning. In *Proc. of the IEEE Int. Conf. on Emerging Technologies and Factory Automation*, 2014.

[21] Sanjeev Srivastava, Eugene Fang, Lorenzo Riano, Rohan Chitnis, Stephen Russell, and Pieter Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 639–646. IEEE, 2014.

[22] Ioan Sucan, Maciej Moll, Lydia E Kavraki, et al. The open motion planning library. *Robotics & Automation Magazine, IEEE*, 19(4):72–82, 2012.

[23] Moritz Tenorth and Michael Beetz. A unified representation for reasoning about robot actions, processes, and their effects on objects. In *International Conference on Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ*, pages 1351–1358. IEEE, 2012.