A knowledge-based planning framework for smart and autonomous manipulation robots

Jan Rosell, Aliakbar Akbari, Muhayyuddin and Mohammed Diab

Abstract— **Manipulation planning in human environments is one of the challenging areas in robotics research. It is focused on making the robot capable of performing complex manipulation tasks, which requires manipulation planning capabilities in cluttered and unstructured environments. These capabilities need, on the one hand, a rich semantic description of the scene and knowledge about the manipulation actions, and on the other, a smart combination of task and motion planning levels. A method, called K-TMP, is proposed here where: *a)* knowledge is coded as an ontology framework with information about objects, robots, sensors and actions, the workspace and the context, as well as an inference mechanism for reasoning over this knowledge; *b)* planning is done with an heuristic task planner based on the Feed Forward method that uses a physics-base motion planner to guide the search and find a feasible sequence of actions to perform the task. Knowledge is used at motion level to evaluate the potential interactions between the robot and the objects, resulting in robust paths in cluttered scenarios with the robot possibly interacting with some objects. At task level knowledge is used to take into consideration all the constraints affecting the actions, making the planning efficient. A table-top example with a bi-manual robot is included, as well as a discussion on challenges and future works.**

I. INTRODUCTION

Challenging robotic problems, like manipulation task in cluttered environments, require the planning at task and motion levels. Motion planning is devoted to finding collision-free paths for the robot between two configurations and is usually done in the configuration space using sampling-based approaches. Several variants may allow to include different constraints and even allow collision with some movable objects, like physics-based motion planning. Task planning is devoted to find a sequence of actions to fulfill a task. It is usually modeled using a state-transition system, and can be solved for instance with methods based on constraint solving or based on heuristic state-space search, like the Fast forward method.

Different tasks may have different grades of complexity, both at symbolic and geometric levels, as well as regarding the dependence between them. Logic states and action have to be mapped to geometric instances, and a state transition can only occur if the geometric instances satisfy the pre-conditions of the action and if the action is geometrically feasible. Therefore, a smart combination of task and motion planning capabilities is required to make the process efficient.

The authors are with the Institute of Industrial and Control Engineering (IOC), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain (jan.rosell@upc.edu). This work was partially supported by the Spanish Government through the project DPI2016-80077-R.

Also, the use of knowledge may enhance the planning capabilities at both levels, giving more autonomy to the robots. Knowledge can be modeled using ontologies, that represent it in the form of concepts along relations. Ontologies can be encoded using the Web Ontology Language (OWL), which makes it a world-wide accessible database, and can be queried and used for reasoning by means of Prolog predicates.

We are mainly interested in table-top manipulation tasks for bimanual robots. These types of tasks must cope with blocking objects, require a correct selection of grasps and placements, and must analyze the feasibility of actions. Also, depending on the type of problem, goal ordering must be carefully handled and even large task spaces may be possible, requiring objects to be moved more than once for achieving the goals.

Contributions: This paper proposes an ontology framework to model manipulation knowledge, and a knowledge-based task and motion planning method, called K-TMP. The proposed method combines the heuristic-based Fast Forward task planner with a physics-based motion planner (that allows push actions) and the ontological knowledge. Both the motion planner and the knowledge are used in off-line and on-line reasoning processes on symbolic literals to determine the actions feasibility and applicability that guide the search of the plan, making the process efficient. The implemented physics-based motion planner has also been used for grasp-in-the-clutter tasks, where the robot can interact with the obstacles obstructing the path towards the goal, moving them away.

After this introduction the paper is structured as follows. Sec. II presents the related work and Sec. III the framework of the proposed approach. Sec. IV summarizes the ontological manipulation framework and Sec. V and Sec. VI the planning at motion and manipulation levels, respectively. Finally, Sec. VII gives some implementation details, and Sec. VIII some results and a discussion on challenges and future work.

II. RELATED WORK

A. Knowledge representation and reasoning

Knowledge representation is concerned with structuring concepts and relations such that they are usable for reasoning tasks done by artificial systems (e.g. robots). Several techniques are proposed for knowledge representation such as ontologies. Formally, "Ontology is defined as an explicit, formal specification of a shared conceptualization" [1]. The conceptualization refers to the abstract models of entities in a certain domain. These models are achieved by defining their relevant concepts along with their relations.

Many efforts are proposed to represent knowledge in a standardized way for robotics and automation field such as Ontologies for Robotics and Automation Working Group (ORA WG), sponsored by the IEEE Robotics & Automation Society [2]. ORA WG has proposed the Core Ontology for Robotics and Automation (CORA) [3]. It provides a conceptual structure that can be used and integrated with other specific ontologies developed for the robotics and automation domain. CORA is extended by integration of specific ontologies such as [4], [5], [6], [7] that are used to describe the robotic world from top-level ontological categorization to the low-level information related the environments' entities, i.e. from the abstract concepts that describe the world, such as the concept of *object* used to describe physical objects in terms of names and functionality, to the position and orientation of the physical objects. This standardized way of representing knowledge facilitates the robot to reason about the way of executing tasks.

Abundant studies have investigated the use of knowledge-based reasoning approaches in planning. These approaches define terminologies and the inference mechanism (way of querying and reasoning over the knowledge) to facilitate the planning process in many domains. For instance, [8], [9] categorize knowledge about the world into terminological knowledge (TBOX), and assertional knowledge (ABOX). The former contains definitions of concepts, such as interaction and action. These concepts and their relations are arranged in a hierarchy. Whereas the latter contains individuals that are instantiations of these concepts. In the navigation area, some works such as [10], [11] use a metric map and a topological map to define the robot environment. The metric map is used for the geometrical representation of the robot workspace in terms of free and occupied areas, while the topological map is used to capture the topology of the workspace. Moreover, In manipulation planning domain, works such as [12], propose an ontological framework to organize the knowledge needed for physics-based manipulation planning, allowing to derive manipulation regions and behaviors. [13] propose ontologies that classify the knowledge into manipulation world and manipulation planning. The former is used to describe the robot environment, while the latter is used to facilitate the planning process, by retrieving the information about the manipulation world.

B. Motion planning

Motion planning problem deals with computing collision-free trajectory to move a robot from the start to the goal state in the configuration space. To plan in higher dimensional configuration spaces, sampling-based motion planning algorithms such as RRT [14] and KPIECE [15] are proposed. These algorithms do not required the explicit representation of the obstacles in the configuration space and plan efficiently for the systems with kinodynamic (geometric and differential) constraints. However these approaches focus on computing collision-free trajectory. Physics-based motion planning has emerged as new class of planning algorithms that allows purposeful manipulations of the objects by considering dynamic interactions (robot-object and object-object) while planning. The results of these dynamic interactions influence the planning process. Physics-based planning approaches employ sampling-based motion planners for sampling the states and constructing the solution path. Whereas physics engines such as Open Dynamic Engine (ODE-http://www.ode.org/) are used for the state propagation that takes care of kinodynamic and physics-based constraints.

Physics-based motion planning is computationally intensive due to the higher dimensional state space, large planning search space and highly constraint solution set. A few approaches are proposed to overcome these challenging issues. For instance, to reduce the search space [16] proposed *Behavioral Kinodynamic RRT*s that define non-deterministic tactics using a finite state machine along with skills to control the sampling. A hybrid approach is proposed in [17] that integrates knowledge of the robot's workspace (represented in the form of ontologies) with physics-based motion planning. The knowledge-based reasoning process is used to reduce the planning search space and to guide the motion planner by defining the manipulation constraints (that determine the way of manipulation) to interact with the objects in the robot's workspace. RRT and KPIECE are used as kinodynamic motion planners and state propagation is performed using ODE. This approach is extended in [13] by incorporating low-level geometric reasoning that determines the appropriate bounds for sampling controls in order to obtain a power-efficient solution. This geometric reasoning process determines whether the robot is in contact with an object(s) or moving freely, and in case of interaction, the control bounds are computed according to the properties of the target object.

C. Task planning

There are various approaches in Artificial Intelligence (AI) planning based on different search strategies that have been popular among robotics researchers.

Search can be done in the plan space, like the *GRAPHPLAN* task planner [18], which uses the *Planning Graph* technique that interleaves state-levels (representing a set of facts) and action-levels (representing a set of actions whose preconditions are satisfied in the previous state level, including maintenance actions that keep literals unchanged for the next level). Mutual exclusion relations among actions and facts may exist in the *Planning Graph*. The search space procedure that builds the planning graph continues until all the goal conditions appear in the last state-level. A plan is then looked for by backtracking from the last state-level towards the initial one taking into account mutual exclusions.

Alternatively, search can be done in state space. In this direction, one of the most efficient task planning approaches is the *FastForward* (FF) [19], which performs a heuristic search. This is the task planner used in this paper. It has two main components: the *Enforced Hill Climbing* (EHC) module devoted to select the more promising successor state using the heuristic values, and the Relaxed *GRAPHPLAN* module that computes the heuristic value in terms of the estimated number of actions. This later module also computes the set of helpful actions (i.e. those actions that executed from that state have a high probability of being in the solution plan), which allows making the exploration more efficient. The *Relaxed GRAPHPLAN* module is based on a relaxed version of the *Planning Graph*. The relaxed version of the *Planning Graph* (called RPG) ignores the delete lists of the actions, so mutual exclusion relations do not take effect in the planning phase. From tthe RPG, the relaxed plan is computed including the sequence of cheapest actions connecting the initial state to the final one. The heuristic value is then computed as the number of actions in the relaxed plan, and the helpful actions are those actions of the RPG that appear in the first action level. If EHC fails, everything done so far is skipped and the FF restarts considering *Best-First Search* (BFS).

Another technique, called hierarchical-based planning, decomposes tasks into sub-tasks and grow in a search space. Searching is performed till all sub-task conditions are met by achieving a set of executable actions, being the dependency among actions modeled by means of networks called *Hierarchical Task Networks* (HTN) [20].

D. Combining task and motion planning

With the aim of finding a feasible plan to solve a given task, different approaches have dealt with various strategies to combine task and motion planning (TAMP), depending on the task planning algorithms used.

Planning Graph-based TAMP. Variants of the Graph Plan have been proposed in [21], [22] to retrieve alternative plans that are later evaluated using a physics-based motion planner, and applied to a manipulation problem of a mobile robot that is able to push and pull movable objects. These approaches can be computationally expensive in terms of the number of calls to motion planning as they need to evaluate many actions.

FF-based TAMP. The study in [23] proposed an interleaved search at the symbolic and geometric levels, where a PRMs motion planner calls the FF task planner to guide roadmap sampling. The approach computes the heuristic value based on the symbolic distance to goal, thus the heuristic function is not informed in terms of geometric information. On the contrary, the work in [24] proposed an approach, called *FFRob*, that when computing the heuristics analyses the action feasibility by using a *Conditional Reachability Graph* (CRG) based on a version of PRM planner, i.e. using geometric information. In an analogous way, the study in [25] also considers geometric information when computing the heuristics. In the NAMO problem of a mobile robot, during the RPG construction, the method calls a physics-based motion planner for those actions that change the topology of the configuration space. To reduce the number of calls to the motion planner, the current study follows this line but only applies relaxed geometric information in the heuristic computation.

Hierarchical-based TAMP. The work in [26] focuses on a combination based on the HTN planner. It facilitates backtracking at different levels, also including an interleaved backtracking procedure. The work in [27] has addressed an aggressively hierarchical approach that constrains the abstract plan steps so that they are serializable (i.e. so that the particular way that the first step is carried out does not make it impossible to carry out subsequent steps), and handles the integration by operating on detailed, continuous geometric representations.

III. Proposed framework overview

Let consider a bi-manual robot able to manipulate objects on its working table, i.e. a robot with the capacity to pick objects (from different grasping configurations) and place them (at different placement regions like shelves, trays or intermediate zones), and also with the capacity to push them (from anywhere or interacting with them through some of their parts). Many objects of different shapes and physical features may be present in the scene. Different tasks may be defined in this scenario, ranging from the grasping of an object in a cluttered scenario to the rearrangement of some of the objects. Several challenges must be faced in order to find a feasible sequence of actions to fulfill the task. These challenges include the detection of blocking objects, the selection of grasps that are kinematically feasible and collision-free for the combined pick and place actions, the finding of object placement taking into account reachability issues and the possible limited space, and the search for collision-free movements of the robot either alone or transferring an object, among others.

To cope with these type of problems and challenges, we propose a framework based on three main parts:

- *Ontologies for manipulation*: A formalization of the manipulation knowledge as an ontology framework that includes information about objects, robots, sensors and actions, the workspace and the context, as well as an inference mechanism for reasoning over this knowledge.
- *Motion planning for grasping*: A physics-based motion planner able to interact with movable obstacles to smartly move away those blocking the target object, and thus allowing simple and efficient grasp-in-the-clutter tasks.
- *Manipulation planning*: A heuristic task planner based on the Fast Forward method with an heuristic to guide the search based on geometric reasoning processes on symbolic literals to determine the actions feasibility and applicability.

IV. Ontologies for manipulation

The framework[1] presented in [12] to organize the knowledge needed for physics-based manipulation planning by automatically construct a semantic map to categorize the objects into different types according to the objects and task constraints. The framework has been proposed to deal with

[1] https://sir.upc.es/projects/ontologies/

autonomous and manipulation tasks by following, on the one hand, some ideas of [10] and on the other hand, some standardized common vocabulary that ORA WG proposes in [6], [7]. Both are breifly described below.

The former is used to describe the concepts through the use of a hierarchy of ontologies composed of three layers: metaontology, ontology schema and ontology instance, as shown in Fig. 1. Metaontology is used to represent generic information, such as the concept of physical object. Ontology schema is used for domain specific knowledge, for instance, ontology layer contains the knowledge of a particular domain, such as kitchen. Ontology instance is used to store the information of the particular objects, such as a given bottle and its properties. These layers are composed of six classes: *Feature, Object, Actor, Space, Context,* and *Action*. Each of them has three gradual levels (except feature class that has two [10]). *Feature class* represents the knowledge related to the properties of the manipulation world such as physical interaction parameters. *Object class* represents the knowledge related to the physical objects and their components such as a cup has handle and container. *Actor class* represents the knowledge related to the robots and their components. *Space class* represents the knowledge related the robot workspace. *Context class* represents the knowledge related to the situation based on space and time. *Action class* represents the knowledge related to the planning processes including motion and manipulation components.

The latter is used to cover the knowledge related to manipulation world, planning and data under the standardized concepts. The manipulation environment knowledge includes the description of objects and robots in the workspace. The manipulation planning knowledge represents the part that is responsible for reasoning about the situation of the objects and the robot, and for planning the tasks. The manipulation data knowledge represents the features of the environment entities, such as color, mass, robot constraints (e.g joint limits).

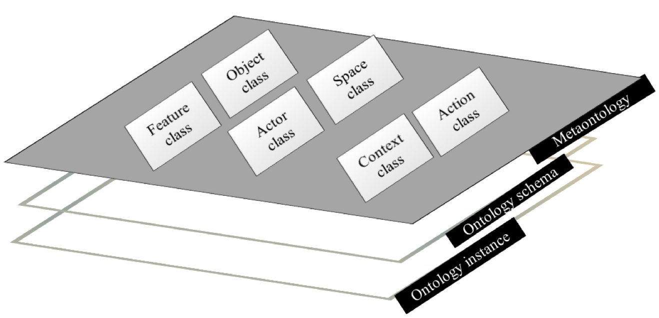

Fig. 1. Structure of the ontological layers metaontology, ontology schema and ontology instance.

Because an ontology is an object-oriented and frame-based language, the metaontology layer can provide a template for the ontology schema layer to build terminology, while the ontology instance layer can be defined as an individual frame. The information of ontological classes, properties, and instances is transferred with bidirectional reasoning in the same knowledge layer. Whereas, unidirectional reasoning relates several knowledge classes of different layers.

V. Motion planning for grasping

One of the challenging issue in Physics-based motion planning for grasping in unstructured and clutter environments is to handle interactions (between robot-object and object-object) in a robust way. These interactions are very difficult to model accurately due to the several parameters involved such as exact value of friction, interaction force direction and pressure distribution under the object surface. The imprecise modeling of such parameters give rise to the uncertainty in the objects' poses. This uncertainty is propagated from initial state to the future states. The computed plan should be robust enough to move the robot from a start to the goal state even in the presence of such uncertainty.

We proposed an open-loop physics-based motion planning approach [28], that computes the motion plan in such a way that it absorbs the potential deviation in the objects' poses (due to sensing and due to interactions). Based on the KPIECE [15] planner, a robust tree-growing strategy is proposed for the underlying sampling-based kinodynamic motion planner. This strategy works in two phases that are; *motion sampling* and *belief computing* phase. Once a state is selected from where the tree will grow, the *motion sampling phase* samples n random controls and time durations to generate n motions by applying the sampled controls for the sampled durations. No uncertainty is considered in the system during this phase. Whereas, *belief computing* phase considers uncertainty into the system and repeatedly apply the sampled controls for the sampled durations. The probability of valid resultant states describes the belief of the corresponding motion. These belief values are then associated with the corresponding motions and influence the tree-growing process. The motions that have high belief are preferred to grow the tree.

Three sources of uncertainty are introduced during the belief computation phase that are: uncertainty in the interaction parameters, uncertainty in the objects' poses and uncertainty in the robot controls. Since a physics-engine is used as state propagation, the interaction parameters of the physics-engine (such as in case of ODE, friction and constraint force mixing, error reduction parameter and bounce velocity) are approximated and Gaussian noise is introduced around the approximated values. The initial pose uncertainty is modeled using multivariate Gaussian distribution. Whereas, as a result of interactions (in the presence of uncertainty) the resultant poses of the objects may vary significantly, the uncertainty in the object poses is multi model and it is computed using a Gaussian mixture model. The uncertainty in the controls is modeled as zero mean multivariate Gaussian distribution.

The proposed method has been successfully tested in simulation and real experiments with both a Kuka LWR robot and a YuMi robot, in cluttered scenes with different daily objects like cans (Fig 2). The method is currently being extended to consider different goals determined by different ways to grasp the object.

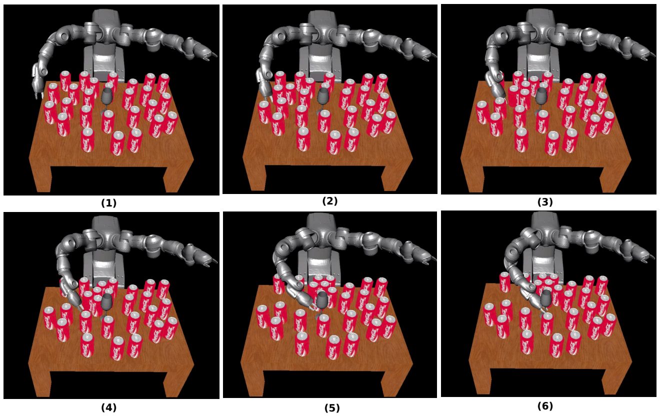

Fig. 2. A grasp-in-the-clutter task executed with the YuMi robot using the proposed physics-based motion planner.

Fig. 3. A table-top manipulation task executed with the YuMi robot using the proposed task and motion planner.

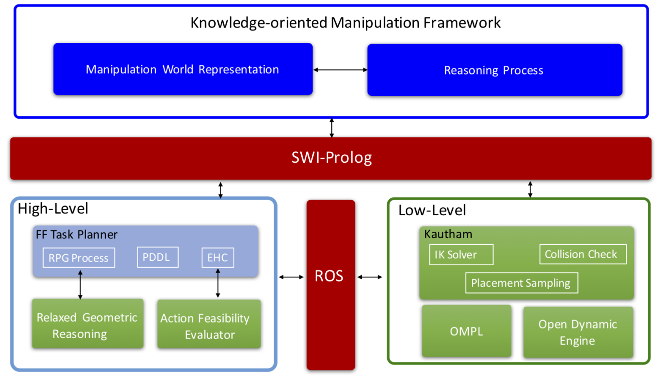

Fig. 4. Implementation framework: components and communications.

VI. Manipulation Planning

The proposed simultaneous task and motion planner, K-TMP, extends the basic FF planner in order to consider geometric and manipulation knowledge information while planning. It consists of three main parts: *Heuristic Computation, State Space Search*, and *Action Selection Process*.

The *Heuristic Computation* gets a state and returns the heuristic value and a set of helpful actions. The standard RPG is first constructed and the relaxed plan is extracted. The actions of this relaxed plan are then forwarded to the relaxed geometric reasoning. The reasoning process (without calling motion planner) tries to find valid configurations for pre-gasping poses and, if required, proper locations for object placements. Information regarding the grasping poses and regions to place objects is retrieved from the knowledge, as well as information to know, for instance, the feasibility of a push action taking into account the object's physical features and the robot capacity If the relaxed plan is feasible, then the heuristic value along the helpful actions are returned. In the case of failure, the reason is identified and fed back to the state (like a blocking obstacle), and an alternative relaxed plan is looked for. The reasoning process makes the heuristic more informed in terms of geometry information.

The *State Space Search* includes the search component that comprises two algorithms, the Enchanced Hill Climbing (EHC) and the Bes First Search (BFS), as the standard FF has. From each state, the search module selects the best action, resulting in a state which has a lower heuristic value in terms of geometry and symbolic information.

The *The Action Selection Process* tries to find a motion for a given action. The action chosen by the state space search is parametrized by setting a query and forwarded to a motion planner in order to compute a path. For the transit and transfer actions, a geometric motion planner is used to find a collision-free path; for push actions, a physics-based motion planner is applied. If motion planning fails, the current state is updated with the failure reason and the search restarts. Otherwise the action is stored in the plan.

The proposal has been successfully applied to a table-top problem executed by the YuMi bimanual robot (Fig 3). The task requires the removing of several blocking objects, the carefully selection of grasps, the arm to be used for each action, the IK solutions chosen, and the selection of the goal ordering.

VII. Framework Implementation

The proposed framework implementation consists of three main phases: task planning, motion planning, and knowledge processing. Task planning is implemented using a modified version of the FF planner developed based on the C++ language that incorporates geometric information, motion planning and knowledge processing. Motion planning is done using The Kautham Project[2] [29] which is a C++ based open-source tool for motion planning, that enables to plan under geometric and kinodynamic constraints. It uses the Open Motion Planning Library (OMPL) [30] as a core set of sampling-based planning algorithms. In this work, the RRT-Connect motion planner and KPIECE are used for geometric and physics-based motion planning respectively. For the computation of IK module, the approach developed by [31] has been used. Regarding the knowledge processing, the knowledge is coded by the Protege editor and the reasoning process is done using the Prolog language. The communication between all modules modules is done via Robotic Operating System (ROS) [32].

VIII. Results and Discussion

Can we pose a problem that highlights the whole framework???

This paper has highlighted the use of knowledge in both task and motion planning levels, and has demonstrated a smart combination between them, allowing an efficient performance in table-top manipulation tasks where there are blocking objects, the correct selection of grasps and placements is critical and the feasibility of actions needs

[2]https://sir.upc.edu/projects/kautham/

to be analyzed. The use of a physics-based motion planner has allowed the consideration of push actions as well as the possibility to permit interactions with obstacles. This allows to efficiently execute grasping motions in cluttered scenarios, by avoiding the need of explicitly computing the removal of obstructing objects and just relying in pushing them away. Knowledge has been used to describe the scene, to determine how objects have to be manipulated and define the actions the robot is able to execute. We claim that a formal description of the manipulation knowledge as an ontological framework gives versatility and flexibility to the robots, further enhanced by following the IEEE robot ontology standards.

Some of the future work directions are the following:

- To test the proposal with more challenging problems requiring larger task spaces.
- To include sensing actions and extending task planning by using the contingent Fast Forward method.
- To consider simultaneous grasp and motion planning strategies in order not to rely on a predefined set of grasping configurations.
- To consider uncertainty at task level by using the probabilistic Fast Forward method.

REFERENCES

[1] T. R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies*, vol. 43, no. 5, pp. 907 – 928, 1995.

[2] C. Schlenoff, E. Prestes, R. Madhavan, P. Goncalves, H. Li, S. Balakirsky, T. Kramer, and E. Miguelanez, "An IEEE standard ontology for robotics and automation," in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 1337–1342.

[3] E. Prestes, J. L. Carbonera, S. R. Fiorini, V. A. M. Jorge, M. Abel, R. Madhavan, A. Locoro, P. Goncalves, M. E. Barreto, M. Habib, A. Chibani, S. Grard, Y. Amirat, and C. Schlenoff, "Towards a core ontology for robotics and automation," *Robotics and Autonomous Systems*, vol. 61, no. 11, pp. 1193 – 1204, 2013, ubiquitous Robotics.

[4] I. Niles and A. Pease, "Towards a standard upper ontology," in *Proceedings of the International Conference on Formal Ontology in Information Systems - Volume 2001*, ser. FOIS '01, 2001.

[5] S. R. Fiorini, J. L. Carbonera, P. Gonalves, V. A. Jorge, V. F. Rey, T. Haidegger, M. Abel, S. A. Redfield, S. Balakirsky, V. Ragavan, H. Li, C. Schlenoff, and E. Prestes, "Extensions to the core ontology for robotics and automation," *Robotics and Computer-Integrated Manufacturing*, 2015.

[6] IEEE-SA, "IEEE Standard Ontologies for Robotics and Automation," pp. 1–60, 2015, Special Issue on AI and Robotics.

[7] J. I. Olszewska, M. Barreto, J. Bermejo-Alonso, J. Carbonera, A. Chibani, S. Fiorini, P. Goncalves, M. Habib, A. Khamis, A. Olivares, E. P. de Freitas, E. Prestes, S. V. Ragavan, S. Redfield, R. Sanz, B. Spencer, and H. Li, "Ontology for autonomous robotics," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2017.

[8] M. Tenorth and M. Beetz, "Representations for robot knowledge in the Knowrob framework," *Artificial Intelligence*, vol. 247, pp. 151 – 169, 2017, special Issue on AI and Robotics.

[9] J.-R. Ruiz-Sarmiento, C. Galindo, and J. Gonzalez-Jimenez, "Building multiversal semantic maps for mobile robot operation," *Knowledge-Based Systems*, vol. 119, pp. 257 – 272, 2017.

[10] G. H. Lim, I. H. Suh, and H. Suh, "Ontology-based unified robot knowledge for service robots in indoor environments," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 41, no. 3, pp. 492–509, May 2011.

[11] G. Gemignani, R. Capobianco, E. Bastianelli, D. D. Bloisi, L. Iocchi, and D. Nardi, "Living with robots: Interactive environmental knowledge acquisition," *Robotics and Autonomous Systems*, vol. 78, pp. 1 – 16, 2016.

[12] M. Diab, A. Akbari, J. Rosell, *et al.*, "An ontology framework for physics-based manipulation planning," in *Iberian Robotics conference*. Springer, 2017, pp. 452–464.

[13] Muhayyuddin, A. Akbari, and J. Rosell, "κ-PMP: Enhancing physics-based motion planners with knowledge-based reasoning," *Journal of Intelligent & Robotic Systems*, Sep 2017. [Online]. Available: https://doi.org/10.1007/s10846-017-0698-z

[14] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[15] I. Sucan and L. E. Kavraki, "A sampling-based tree planner for systems with complex dynamics," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 116–131, 2012.

[16] S. Zickler and M. Veloso, "Efficient physics-based planning: sampling search via non-deterministic tactics and skills," in *Proc. of The 8th Int. Conf. on Autonomous Agents and Multiagent Systems-Volume 1*, 2009, pp. 27–33.

[17] Muhayyuddin, A. Akbari, and J. Rosell, "Ontological physics-based motion planning for manipulation," in *IEEE International Conference on Emerging Technologies Factory Automation*, 2015, pp. 1–7.

[18] A. L. Blum and M. L. Furst, "Fast planning through planning graph analysis," *Artificial intelligence*, vol. 90, no. 1, pp. 281–300, 1997.

[19] J. Hoffmann and B. Nebel, "The FF planning system: Fast plan generation through heuristic search," *Journal of Artificial Intelligence Research*, pp. 253–302, 2001.

[20] M. Ghallab, D. Nau, and P. Traverso, *Automated planning: theory & practice*. Elsevier, 2004.

[21] A. Akbari, Muhayyudin, and J. Rosell, "Task and motion planning using physics-based reasoning," in *IEEE Int. Conf. on Emerging Technologies and Factory Automation*, 2015.

[22] A. Akbari, Muhayyuddin, and J. Rosell, "Reasoning-based evaluation of manipulation actions for efficient task planning," in *ROBOT2015: Second Iberian Robotics Conference*. Springer, 2015.

[23] S. Cambon, R. Alami, and F. Gravot, "A hybrid approach to intricate motion, manipulation and task planning," *The International Journal of Robotics Research*, vol. 28, no. 1, pp. 104–126, 2009.

[24] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "FFRob: An efficient heuristic for task and motion planning," in *Algorithmic Foundations of Robotics XI*. Springer, 2015, pp. 179–195.

[25] A. Akbari, Muhayyudin, and J. Rosell, "Task planning using physics-based heuristics on manipulation actions," in *IEEE Int. Conf. on Emerging Technologies and Factory Automation*, 2016.

[26] L. de Silva, A. K. Pandey, M. Gharbi, and R. Alami, "Towards combining HTN planning and geometric task planning," in *RSS Workshop on Combined Robot Motion Planning and AI Planning for Practical Applications*, 2013.

[27] L. P. Kaelbling and T. Lozano-Pérez, "Hierarchical task and motion planning in the now," in *IEEE Int. Conf. on Robotics and Automation Robotics and Automation*, 2011, pp. 1470–1477.

[28] Muhayyuddin, M. Moll, L. Kavraki, and J. Rosell, "Randomized physics-based motion planning for grasping in cluttered and uncertain environments," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 712–719, April 2018.

[29] J. Rosell, A. Pérez, A. Aliakbar, Muhayyuddin, L. Palomo, and N. García, "The kautham project: A teaching and research tool for robot motion planning," in *Proceedings of the IEEE Emerging Technology and Factory Automation (ETFA)*, 2014.

[30] I. Sucan, M. Moll, L. E. Kavraki, *et al.*, "The open motion planning library," *Robotics & Automation Magazine, IEEE*, vol. 19, no. 4, pp. 72–82, 2012.

[31] J. Zaplana, I.; Claret and L. Basanez, "Kinematic analysis of redundant robotic manipulators: application to kuka lwr 4+ and abb yumi," *Revista Iberoamericana de Automtica e Informtica Industrial. In press.*, 2017.

[32] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, vol. 3, no. 3.2, 2009, p. 5.