Official URL:
https://doi.org/10.1109/ECMSM51310.2021.9468850

To cite this version:

Léoty, Florent and Fillatreau, Philippe and Archimède, Bernard *Path planning control using high abstraction level environment model and industrial taskoriented knowledge.* (2021) In: 2021 IEEE International Workshop of Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM), 21 June 2021 - 22 June 2021 (Liberec, Czech Republic).

Path planning control using high abstraction level environment model and industrial task-oriented knowledge

Florent LEOTY
Laboratoire Génie de Production
Université Fédérale de Toulouse, INPT/ENIT
Tarbes, France
florent.leoty@enit.fr

Philippe FILLATREAU
Laboratoire Génie de Production
Université Fédérale de Toulouse, INPT/ENIT
Tarbes, France
philippe.fillatreau@enit.fr

Bernard ARCHIMEDE
Laboratoire Génie de Production
Université Fédérale de Toulouse, INPT/ENIT
Tarbes, France
bernard.archimede@enit.fr

***Abstract*— In order to face an increasing economic competition, industrial manufacturers wish to reduce the time and cost of product development. Furthermore, up-to-date products are more and more integrated, and must be assembled, disassembled or maintained under potentially very strong geometric constraints. In the context of Industry 4.0, manufacturers are therefore expressing the desire to validate all the tasks related to their products lifecycles, from design stage on, by simulation using a digital mock-up, and before building the physical prototypes. A key issue is then to find a trajectory, a movement, to show the feasibility of the simulated scenarios. Automatic path planning algorithms, developed by the robotics community from the 1980s on, have been widely used for this purpose. In this paper, we intend to improve the relevance of the trajectories proposed by such algorithms and the associated computation times. To do so, we consider: a) the use of path planning algorithms or of combinations of these; b) the involvement for the environment modelling of data with a higher abstraction level than the purely geometric data traditionally used; and c) the representation of the knowledge related to the task to be performed by using ontologies. The approaches developed and associated improvements of the state of the art are validated experimentally through the simulation of highly geometrically constrained manipulation tasks.**

Keywords*— *path planning, environment modelling, geometry, topology, semantics, ontologies, Industry 4.0.

I. Introduction

Industrial competition is increasingly strong, particularly at the economic level. To face it, manufacturers wish to reduce costs and development time. Moreover, up-to-date products are more and more integrated and must therefore be assembled, disassembled and maintained potentially under very strong geometric constraints. In the context of Industry 4.0, industrial companies want to validate all the tasks of the lifecycle product, from design stage on, using virtual mock-ups. To do so, an important key issue is then to find an achievable trajectory to prove the feasibility of the simulated scenarios.

In this work, we are particularly interested in automatic path planning techniques. These methods, studied by the robotics community from the 1980s on, are widely used. However, these methods mainly use purely geometric data, which can take a very long time to calculate a path, find irrelevant trajectory or may fail.

However, a human being can solve complex assembly problems in a short time thanks to his cognitive abilities. Indeed, his understanding of the environment and the manipulated object is not limited to the geometrical information, he uses higher abstraction level information. Moreover, he also has task-related information to perform manipulation tasks.

A tool is then useful for knowledge modelling. Moreover, the knowledge modelling allows to reason on the task to be performed. For that, ontologies are widely used. Ontology is an "explicit specification of conceptualizations" for a certain domain of interest [1].

Our goal in this work is to improve the performance of automatic path planners for the manipulation tasks. We want to improve the relevance of the trajectories proposed, reduce the associated computation times and reduce the failure rate. In this paper, we propose a new path planning strategy that uses a combination of several algorithms for a global path planning query based a multi-level architecture and using ontologies.

This paper is organised as follows: the second section presents an overview of the existing work about path planning techniques, the task-related information, the existing ontology models and previous works in our laboratory. The third section explains our approach. The fourth section describes the obtained results. Finally, the conclusion and future works are presented in the fifth section.

II. State of the Art

A. Path planning techniques

1. Geometric motion planning

To validate the tasks of the lifecycle product, a key issue is path planning. This concept allows to find a collision-free path, from the initial to the final configurations and was widely developed by the robotic community forty years ago in particular thanks to the work of [2]. Path planning techniques can be categorised according to the environment model used and how it is explored. There are global approaches based on a comprehensive environment model and local approaches that use progression around the neighbourhood. Some techniques use deterministic exploration of the environment when other techniques are probabilistic.

Deterministic techniques with global approaches use a complete model or cartography of the free space of the environment [3]. Then a deterministic exploration is used to find a trajectory. The model can be computed off-line in the case of a static environment and so be reused for many path planning queries. These are complete techniques (or at least complete in resolution) which is a big advantage. However, the computation time to build the environment can be very long.

Deterministic techniques with local approaches use potential fields (attractive field for the goal configuration, repulsive field for obstacles) [4]. These techniques are faster but have the disadvantage of not being complete

Probabilistic techniques with global approaches generally construct roadmaps that are built by random draw in free space. The resulting graph is then explored to find a trajectory. They are complete methods.

Finally, probabilistic techniques with local approaches randomly explore the environment from a given configuration (or from several configurations). A tree is constructed and is modified when a new draw is done. An example is the Rapidly-exploring Random Tree algorithm [5]. These techniques seem to be the most relevant for manipulation scenarios under very strong geometric constraints because they ensure to find a trajectory, if one exists, without building the environment.

All the presented strategies are traditionally using geometrical models of the environment and can have a very high computation time or fail, while generated trajectory could be not very relevant. The path planning query can also fail. So, move away from the "all geometric" framework could be interesting. Indeed, humans can easily complete a manipulation task.

2. Motion planning using higher abstraction level information

Inspired by the human being's cognitive capacities, researchers have started to take an interest in higher abstraction level information for path planning. Indeed, when a human being analyses an environment, he doesn't only consider it as a sum of geometric objects but also relies on his knowledge and a higher level of abstraction to best interpret it.

Some works propose a semantic 3D object map to modelling the environment [6]. Moreover, information, as topological or semantic information, can also be used for navigation [7]. However, these works consider only the environment information. But the task-related information can also be important during a simulated manipulation task.

B. Modelling task-related information using ontology

The information at the task level can be categorized into two parts [8]: the 3D environment information which is the place where the manipulation is performed and the task-related geometric constraints which restrict the final pose of the manipulated object. They are dependant on the manipulation tasks, so, when an environment changes, all information about the environment and task should be reconsidered. Using a modelling tool as ontology is then relevant. An ontology allows the reasoning on the taxonomic knowledge and gives a precise and easy way to model concepts of a specific point of view and the relation between these concepts. Besides, it is easy to reuse an existing ontology. Several works have conceptualized ontologies that exclusively relate to environment modelling. The environment is a closed part of 3D Cartesian space with free space and obstacles. The main information, in relation to our works, are:

- The geometric information consider the representation of a CAD model. OntoBREP [14]. This introduces an approach that exploits the geometric descriptions of CAD models at the knowledge level.
- The topological information: KnowRob ontology [9] defines both the concept of places representing the relevant places in the environment and the concept of topological maps.
- The semantic information attach attributes for each obstacle (e.g. shape, functionality) and free space. To attach the attribute. KnowRob ontology relies on encyclopaedic knowledge to describe the types and properties of objects, as well as the development of common sense to describe the usefulness of an object

The task-related geometric constraints are usually defined with mathematical functions [10], generally equality or inequality. There are ontologies concerning the definition of geometrical constraints to better control the trajectory. [11], after building a model based on the geometry of space, defines a set of geometric constraints to guide trajectory planning.

C. Multi-level GTS and GTO path planning architecture

In this work, we use the multi-level architecture for environment modelling and path planning proposed in [12]. It allows using higher abstraction level information as topological and semantic information to improve the path planning with more relevant trajectories, lower computation time and fewer failures. This approach was also completed by [8] by ameliorating the semantic information using ontologies.

1. Multi-level environment modelling

This architecture is presented in a 2D environment example, composed of 4 rigid bodies but works similarly in 3D cases. The environment model is made of two parts: the free space model and the rigid bodies model (which may be static or fixed). In this case, the rigid bodies are represented by a geometric layer (Figure 1a) and a semantic layer (given information as shape, mobility …). Free space, on the other hand, is composed of three layer: a geometric layer (cell decomposition based on an unbalanced tree (Figure 1b)), a topological layer (based on a graph made of borders and places (Figure 1c) which are associated with a set of geometric cells) and a semantic layer (attributes attached to the free space of the topological graph (Figure 1d)).

2. GTS motion planning strategy

A multi-level path planning is carried out with the help of the multi-layer environment information and is also divided into two steps:

- Coarse planning, finding a topological path. This topological path is broken down into topological steps (consisting of a place to be crossed and a border to be reached). The topological path corresponds to the least expensive path according to the topological graph. The weights of the borders are defined according to the local semantics.
- Fine planning which, for each topological step, applies a bi-RRT algorithm.

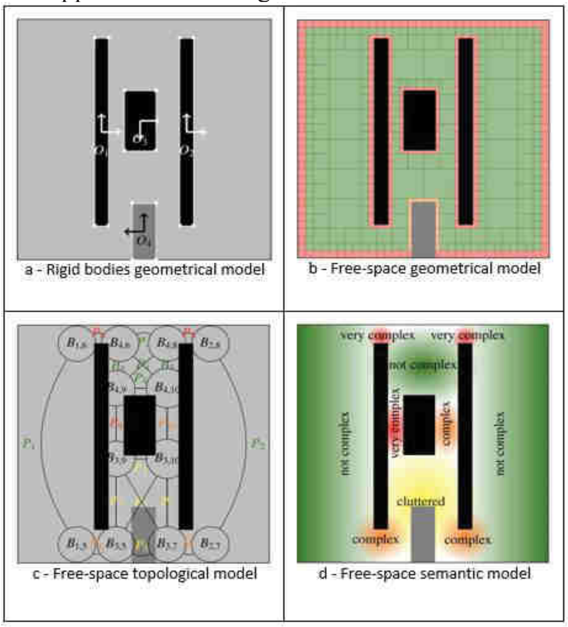

Figure 1 Multi-level environment

Three planning strategies have been defined. The G strategy, which is a reference planning strategy in the literature that exploits only the geometric level, the GT strategy exploits the geometric and the topological information and the GTS strategy executes the two planning strategies described above. As the goal of the work is to validate manipulation tasks under high geometric constraints, the geometric planner used is the bi-RRT algorithm [13]. This algorithm deploys a tree from the start configuration and a second tree from the goal configuration until they can be linked. This is a complete method but does not guarantee to find a result in a finite time. These strategies permit to improve the path planning control. However, the semantic information can be enriched and the task-related information can be used.

3. GTO motion planning strategy

Using ontologies in this approach allows to modelling both the environment and the task-related information. Ontologies permit also reason on the task to be performed with the modelling information. The EnvOn ontology [8] has been developed to model the multi-level environment presented in II.C.1 . Then, an action-specific knowledge ontology, using this information, generates path planning queries while generating geometric constraints for each milestone (e.g. goal configuration or borders) to better control trajectory planning.
Two strategies were then implemented. The GTO strategy, which is an ontology-driven two-step path planning with geometric and topological information, and the GO strategy, which is an ontology-driven geometric path planning. These different strategies allow improving the relevance of the trajectory found while reducing computation times. However, the trajectories obtained are not natural regarding what a human would do.

III. Proposed Motion Planning Approach

In our approaches, we will improve the control over the path planning process to improve the found trajectory, the associated computation time and the failures rate. Specifically, we are interested in a) reducing the search space of a probabilistic algorithm and b) using different planners or a combination of planners to solve a global path planning query, depending on local semantics context.

A. Reducing the search space of probabilistic algorithms

For the resolution of simulated manipulation tasks under strong geometric constraints, we chose bi-RRT algorithm because it allows being sure to find a trajectory if one exists. Such an algorithm is "blind", which means it randomly draws configuration in the whole environment, without considering the relevance of the draws made. Only the collision checker is important. So, the number of random samples can be very high and the associated computed times can be long. Consequently, it is necessary to improve control over this algorithm. To do that, we propose to reduce the search space of the algorithm to avoid irrelevant areas. This should have the effect of reducing the number of random draws needed to resolve a path planning request while making the resulting trajectory more relevant. This is achieved by using the A* algorithm [14] on the octree to find a 3D tunnel of continuous free space between the start and goal configurations. This 3D tunnel can then be considered as a new sub-space for a probabilistic algorithm. On an example of insertion in a box, the search space is the whole workspace (Figure 2a). After cell decomposition (Figure 2b) and the use of the A* algorithm, the search space is reduced to the yellow area (Figure 2c).

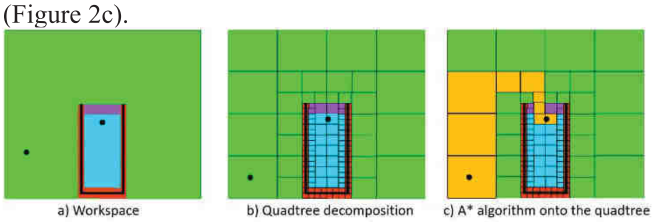

Figure 2: Algorithm A* into the octree

B. A multi-planners strategy

In the literature, solving motion planning query is usually done by using a single planner. However, in some cases, the single algorithm uses isn't relevant for the whole path planning. It could be better to use, locally, a simple planner (e.g. A*) if the environment is low constraint and a complex planner (e.g. bi-RRT) if the environment is highly constrained. Consequently, our second contribution is to show the utility of using different algorithms in the same path planning query, depending on the local semantic context.
To do that, two actions are required:

- Obtained a topologic path and on each topological step, use different path planner strategies [12].
- Use the local semantics to choose the more relevant motion planning strategy. The semantic is defined in the EnvOn ontology.

In this work, we are using only two planners: a straight line if there is no obstacle between the start and the goal configurations, and bi-RRT otherwise. The latter is a complex algorithm, we use it when it is relevant. Else, a simpler algorithm is favoured if it is possible.

C. *The proposed strategies*

Different strategies have been developed to compare our two different approaches:

- G*T strategy that is based on the GT strategy. It uses topological and geometric level information and deploys the bi-RRT into the tunnel-shaped subspace obtained with the A*
- G*TO strategy which is an ontology-driven two-step path planning with geometric and topological information, using bi-RRT in the subspace found by the A* on the octree.
- MGTO strategy is an ontology-driven multi-planners path planning with topological and geometric data. It uses the semantic information of the EnvOn ontology to choose the planner.
- MG*TO which combined MGTO and G*TO strategies.

The path planning method of the MG*TO is presented in Figure 3. The *Planning Manager* is responsible for the path planner, which is made of two parts: *Coarse motion planning*, which returns a topological path, and *Fine motion planning*, which returns the 6D geometric path, and ontology (in blue), where the semantics information are described and which automatically generates geometric constraints and path planning requests [8].
This paper focuses on the modification of *Fine motion planning* (Figure 4).

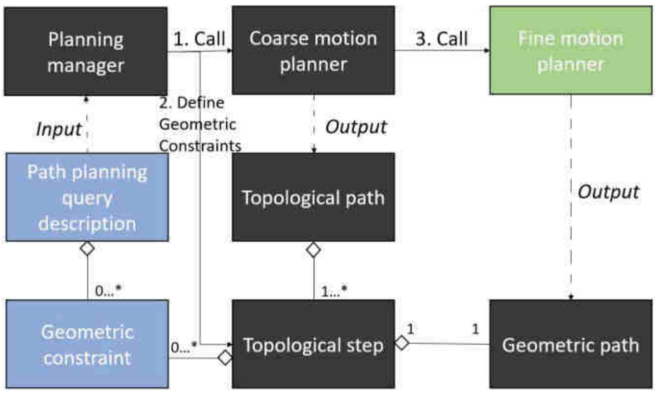

Figure 3: Class diagram of the path planning method

The sequence diagram shows how *Fine motion planning* works. If the free space is sufficiently unrestrained between the start (S) and goal (G) configuration, the straight-line planner is called. Otherwise, the A* algorithm is used on the octree to obtain a tunnel-shaped subspace. Then the bi-RRT algorithm is deployed in the resulting tunnel to find the 6D geometrical trajectory.

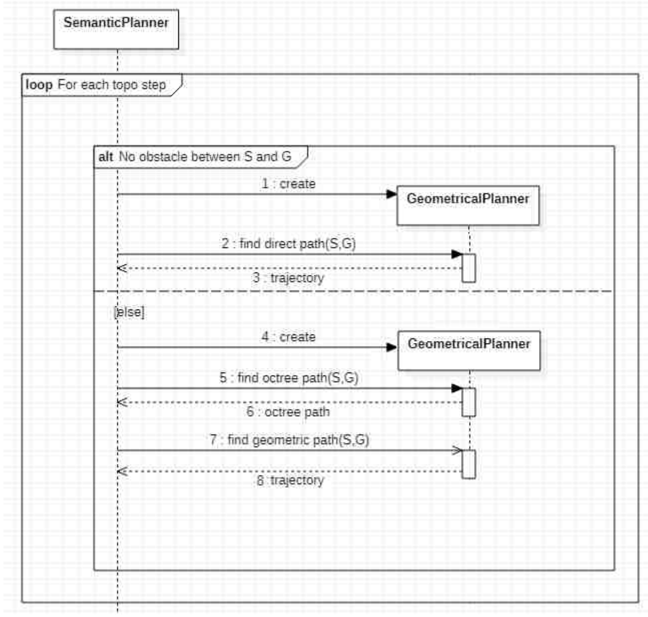

Figure 4: Sequence diagram of the fine motion planning of MG*TO strategy

IV. RESULTS AND VALIDATION

These approaches were validated on two different scenarios: insertion of a pen into a pen box, and a "shaped game" for babies which consists in inserting a shape into the corresponding hole. Even if the scenarios are basics, there are relevant to compare path planning strategies. Indeed, these applications are in a highly geometrically constrained environment. Moreover, the manipulated objects have standard shapes, commonly used in the industry. Then, they are manipulated through constrained passages or inserting into complex holes, which can be considered as generic tasks during the lifecycle product. Finally, we don't consider specific industrial tasks, so we can objectively validate our approach. For both applications, we compare our strategy to the G, GT and GTO strategies presented in sections II.C.2 and II.C.3. We chose these strategies as references because G strategy is a traditional strategy uses on a purely geometrical model, GT strategy because it uses, also, topological information and GTO strategy because it uses also semantic and task-related information.

A. *Study of the use case n°1: insertion of a pen in a pen box*

The use case is the insertion of a pen into a pen box. The results obtained are presented in Figure 5 and Figure 6. We can conclude:

- According to the state of the art, GTO strategy is better than GT strategy, which is better than G strategy. Using high abstraction level information allows better motion planning.
- MGTO strategy permits to reduce the number of random samples by 40% and the computation time by an order of magnitude of fifty percent in comparison to GTO strategy.
- The use of the A* algorithm before deploying the bi-RRT algorithm reduces the number of samples by 50% and allows to gain around 10% of computation time (G*T, G*TO, MG*TO respectively compared

to GT, GTO and MGTO). This is because the search space is greatly reduced Table 1 and Table 2.

Our MG*TO approach greatly improve the path planner's performance compared to the geometrical planner (e.g. G strategy). Indeed, the number of samples when using MG*TO strategy is divided by 75 in comparison to the G strategy. It is divided by 120 for the computational time.

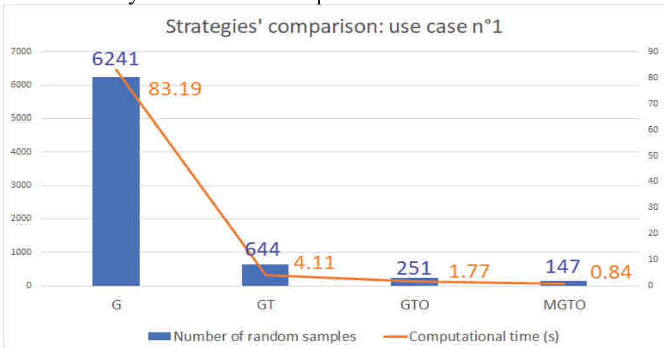

Figure 5: Path planning's results for use case n°1

Strategy	G	GT	GTO	MGTO
	6241	644	251	147
Reduced search space		G*T	G*TO	MG*TO
	X	287	137	83

Table 1: Strategies comparison without and with reducing search space: number of random samples for the use case n°1

Strategy	G	GT	GTO	MGTO
	83.19	4.11	1.77	0.83
Reduced search space		G*T	G*TO	MG*TO
	X	3.79	1.44	0.7

Table 2: Strategies comparison without and with reducing search space: computational time for the use case n°1

Moreover, these approaches allow having a more relevant path. Indeed, using MGTO strategy allows to reducing the path size by around 50% in comparison to the G strategy. In this case, the first topological step doesn't use bi-RRT but a straight line, which is shorter (Figure 6c). In this use case, the insertion task, corresponding to the second topological step, can't be a straight line because the margin between the pen and the hole isn't enough. Besides, reducing the search space of bi-RRT thanks to the A* algorithm (Figure 6b), allows for reducing the path size by 10% (Table 3) in comparison to the strategies without reducing search space.

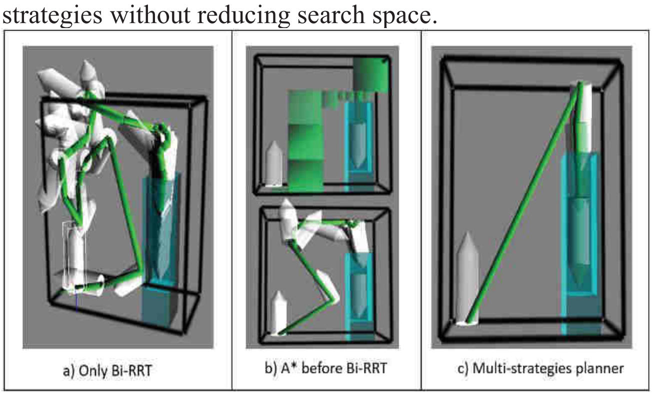

Figure 6: Obtained path using the different planner (case n°1)

Strategy	G	GT	GTO	MGTO
	31.39	26.28	24.22	15.20
Reduced search space		G*T	G*TO	MG*TO
	X	24.28	22.85	14.27

Table 3: Path size for the use case n°1

B. *Study of the use case n°2: "shape game for babies"*

The second use case corresponds to a shape game for babies. The aim is to insert an object in the hole with the corresponding shape. In this application, there are five rigid bodies (one fixe and four mobile) presented in Figure 7. Our approaches were tested on the triangle, the square and the cylinder forms.

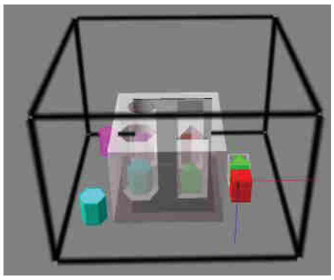

Figure 7: Use case n°2 environment

The results obtained are presented in Figure 8 (number of random samples) and Figure 9 (computational time). The size path is compared in Table 6 to discuss the path relevance. We can conclude:

- As for the first use case, we can conclude that GTO strategy is better than GT strategy and the latter is better than G strategy.
- The G*T, G*TO and MG*TO strategies are respectively better than GT, GTO and MGTO because the search space of the bi-RRT is highly reduced. Then, the number of samples and the computational time are reduced (Table 4 and Table 5).
- The results obtained with the multi-planners strategy are more interesting than the ones obtained with a unique planner. Indeed, for one topologic step, bi-RRT isn't used. Then, the computed time and the number of samples decrease.

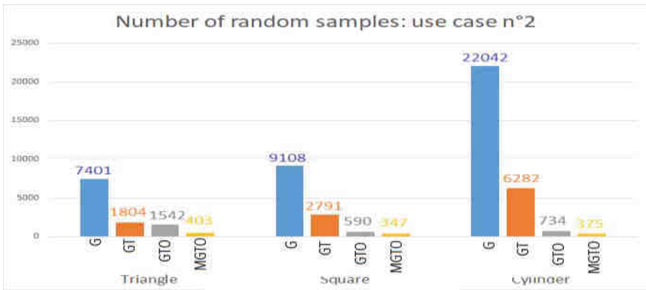

Figure 8: Path planning's results for use case n°2: number of random samples

Strategy	G	GT	GTO	MGTO
Triangle	7401	1804	1542	403
Triangle with A*	X	555	496	286
Square	9108	2791	590	347
Square with A*	X	601	501	307
Cylinder	22042	6282	734	375
Cylinder with A*	X	1952	186	107

Table 4: Strategies comparison without and with reducing search space: number of random samples for the use case n°2

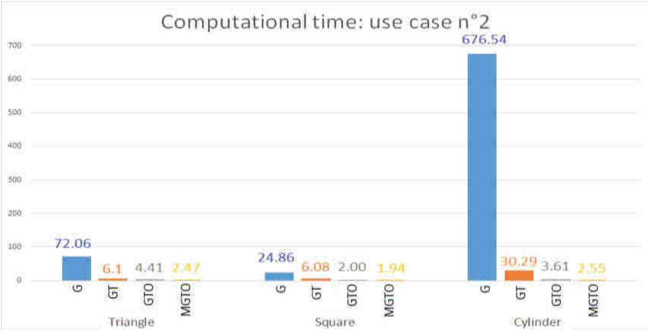

Figure 9: Path planning's results for use case n°2: computational time (s)

Strategy	G	GT	GTO	MGTO
Triangle	72.06	6.1	4.41	2.47
Triangle with A*	X	4.6	3.58	2.07
Square	24.86	6.08	2.00	1.94
Square with A*	X	4.46	1.92	1.66
Cylinder	676.54	30.29	3.61	2.55
Cylinder with A*	X	12.63	1.72	1.25

Table 5: Strategies comparison without and with reducing search space: computational time for the use case n°1

Table *6* presents the path size for the different strategies used. In all cases, the G strategy has the longest path size because bi-RRT isn't controlled. G*T, G*TO and MG*TO are respectively smaller than GT, GTO and MGTO because the A* algorithm allows reducing the search space (Figure 10b). This reduction of search space allows reducing the path size of 30%, 20% and 40% for the path planning of the triangle, the square and the cylinder. In this use case, the multi-planners strategy allows for reducing the path size by 5%. Indeed, a straight line can be used for the topological step corresponding to the insertion (Figure *10*c). In this use case, the margins between the holes and the manipulated object are enough to do that.

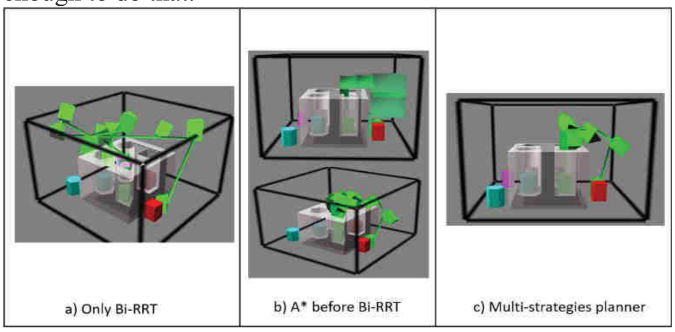

Figure 10: Obtained path using the different planner (case n°2)

	G	GT	GTO	G*T	G*TO	MGTO	MG*TO
Triangle	6.86	5.2	5.31	3.65	3.77	5.08	3.71
Square	6.8	5.3	5.19	4.18	4.11	4.91	3.98
cylinder	6.73	6.49	6.48	3.76	3.72	6.1	3.31

Table 6: Path size for the use case n°2

V. Conclusion and Futures Works

In this paper, we improved the control over the path planning and developed a new approach to improve the relevance of the trajectory and to reduce the associated computation times have been presented. This approach can be summarised in two points:

- Reducing the search space when using a probabilistic algorithm;
- Apply different path planning strategies according to the local semantics context.

However, other improvements can be made. It would be possible to use ontologies to automatically choose the trajectory planner according to the local semantics to have a relevant choice of the local planner. Besides, new planners could be added [15] or new combinations used.

Furthermore, jointly use path and task planner would allow better control of the trajectory. In addition, by adding an advanced task planner, it could be interesting to be able to interact directly with the found task plan to act on the manipulation task and thus act on the trajectory.

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, Nov. 1995.

[2] Lozano-Perez, "Spatial Planning: A Configuration Space Approach," *IEEE Transactions on Computers*, vol. C–32, no. 2, pp. 108–120, Feb. 1983.

[3] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 2, pp. 129–147, Jun. 1982.

[4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *1985 IEEE International Conference on Robotics and Automation Proceedings*, 1985, vol. 2, pp. 500–505.

[5] S. M. Lavalle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," 1998.

[6] R. B. Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments," *Künstl Intell*, vol. 24, no. 4, pp. 345–348, Nov. 2010.

[7] M. Devy, R. Chatila, P. Fillatreau, S. Lacroix, and F. Nashashibi, "On autonomous navigation in a natural environment," *Robotics and Autonomous Systems*, vol. 16, no. 1, pp. 5–16, Nov. 1995.

[8] Y. Zhao, P. Fillatreau, M. H. Karray, and B. Archimède, "An ontology-based approach towards coupling task and path planning for the simulation of manipulation tasks," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, 2018, pp. 1–8.

[9] M. Tenorth and M. Beetz, "KNOWROB — knowledge processing for autonomous personal robots," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 4261–4266.

[10] J. Guitton, "Taking Into Account Geometric Constraints for Task-oriented Motion Planning," p. 9, 2009.

[11] A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "An ontology for CAD data and geometric constraints as a link between product models and semantic robot task descriptions," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4197–4203.

[12] S. Cailhol, P. Fillatreau, Y. Zhao, and J.-Y. Fourquet, "Multi-layer path planning control for the simulation of manipulation tasks: Involving semantics and topology," *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 17–28, Jun. 2019.

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

[14] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[15] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime Motion Planning using the RRT*," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 1478–1483.