Integration of risk in hierarchical path planning of underwater vehicle

Nicolas Lefebvre * **Ingrid Schjølberg** * **Ingrid B. Utne** *

* *Department of Marine Technology, Norwegian University of Science and Technology (NTNU), NO 7491 Trondheim, Norway (e-mail: nicolas.lefebvre@ntnu.no, ingrid.schjolberg@ntnu.no, ingrid.b.utne@ntnu.no)*

Abstract: These instructions give you guidelines for preparing papers for IFAC technical meetings. Please use this document as a template to prepare your manuscript. For submission guidelines, follow instructions on paper submission system as well as the event website.

Keywords: Path planning, Probabilistic risk assessment, Multiobjective optimisations, Autonomous vehicles, Hierarchical decision making

1. INTRODUCTION

In underwater inspection, maintenance and repair (IMR) operations, the use of autonomous underwater vehicles (AUVs) as an alternative to remotely operated vehicles (ROVs) constitutes a significant potential for saving costs. The benefits are particularly related to the ability of autonomous systems to operate in complex, harsh and remote environments without the support of a surface vessel. The day rate for surface vessels is high and operations are highly dependent on the weather and climate. The use of autonomous systems will lead to enhanced cost-effectiveness, reduced downtime, and may have a positive impact on health, safety and the environment (HSE), see Winnefeld and Kendall (2011).

AUVs are exposed to random environmental conditions and operational hazards challenge the risk management of the IMR operations. In this context, a direct approach to risk control is path planning adaptation, specially related to AUV inspection task where collision is one of the major undesired events. This gives significant advantages compared to existing systems which mainly optimize the path planning on minimal cost or operational time. Note that the application developed in this paper is AUV operations but the approach can be applied for other autonomous operations.

Several related works can be mentioned. Bae et al. (2015) consider a problem of finding a risk-constrained shortest path for an unmanned combat vehicle. The problem is solved by the use of a dynamic programming approach. The problem of computational time is tackled by a straightforward limitation of the explored area. Pereira et al. (2013), as Greytak and Hover (2009), use the A-Star algorithm to find minimum risk path for an AUV or underactuated surface vessel. The dealing between path length and risk is also discussed but no solution is suggested to balance out the inefficiency of the considered heuristic in A-Star. De Filippis et al. (2011) discuss the use of A-Star and genetic algorithm to find minimal risk path.

The main contribution of this paper is to propose a trade-off between computational requirements and the optimality of the found path. This is obtain through the hierarchical path planning methodology. This method enables, in a certain way to compensate the inefficiency of the heuristic associated to the integration of the risk of collision in the path optimization. To a lesser extent, this paper does not integrate risks of collision as given quantities but decompose risk through the probabilities of collision and consequences cost. ?

The remainder of this paper is organized as follows: section 2 gives an overview of risk and general considerations on how the risk can be associated to the path planning task. Section 3 focuses on path planning algorithms, on the definition of a relevant heuristic in association to the risk. The proposed approach is developed in this section. Section 4 presents the case study. Section 5 provides computational experiments to illustrate the developed strategy. Finally, the article ends with concluding remarks and perspectives.

2. RISK IN PATH PLANNING

The aim of this section is, firstly, to provide a brief introduction to risk. Secondly, the interactions between risk and AUV control and especially the path planning are discussed. Then, risk assessment along a path is presented.

2.1 Risk

According to Kaplan and Garrick (1981) and Rausand (2011), risk analysis answers three main questions: 1. What can go wrong? 2. What is the likelihood of that happening? and 3. What are the consequences? Let risk be defined as a set of n_R triplets where E_i denotes the hazardous event i, $\Pr\{E_i\}$ the probability of occurrence of this event and C_i the cost associated to its consequences.

$$R = \{(E_i, \Pr\{E_i\}, C_i)\}_{i=1}^{n_R} \tag{1}$$

The risk management framework, proposed in ISO (2009), identifies the steps for risk identification (i.e., identification of the set R). If all hazardous events have been considered and included, the set of triplets is considered complete and represents risk. Let r_T be the random variable which characterises the cost associated to the occurrence of the hazardous events:

$$r_T = \sum_{i=1}^{n_R} C_i \mathbb{I}_{\{E_i\}} \tag{2}$$

The total risk R_T is defined as the mean value of r_T:

$$\begin{aligned} R_T &= \mathrm{E}\left[r_T\right] = \mathrm{E}\left[\sum_{i=1}^{n_R} C_i \mathbb{I}_{\{E_i\}}\right] \\ &= \sum_{i=1}^{n_R} C_i \mathrm{E}\left[\mathbb{I}_{\{E_i\}}\right] = \sum_{i=1}^{n_R} C_i \mathrm{Pr}\left\{E_i\right\} \end{aligned} \tag{3}$$

One of the major steps in the process of risk identification is the determination of the risk influencing factors (RIFs). RIFs can be defined as an aspect (event/condition) of a system or an activity that affects the risk level of this system or activity, see Oien (2001). In the context of AUV operation they are related to the situation of the vehicle and represented by parameters, e.g., energy capacity, vehicle speed and/or location, environmental conditions like wind, waves, current. The most part of these parameters are connected to the vehicle path.

TODO: Try to insert here some elements about the fact that we focus in the following only on the risk of collision.

2.2 Integration of risk in path planning

Referring to National Research Council (US) (2005), an AUV is a vehicle that possesses self-governing capabilities allowing it to carry out tasks without human intervention. Basically, the following units constitute the control architecture of an AUV (See Fig. 1).

- the definition of a path is achieved by the path planning unit. This one integrates vehicle environment (e.g., obstacles) and vehicle characteristics (e.g., dimensions).
- the estimation of the vehicle state (e.g., position) is achieved by the navigation unit which may rely on satellite navigation system and on inertial measurement unit.
- the determination of the trajectory that minimizes the distance apart the theoretical position of the vehicle and its actual one (respectively provided by path planning and navigation units) is achieved by the guidance unit.
- At last, the command of actuators to pursue the trajectory defined by the guidance unit is achieved by the control unit.

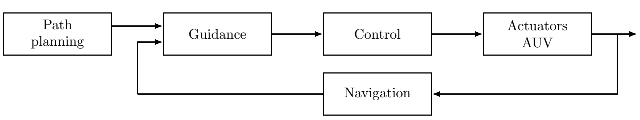

Fig. 1. AUV flying control architecture

According to Campbell et al. (2012) -see also Khorrami and Krishnamurthy (2009)- a simple way to integrate risk is at the supervision level through the adaptation of the vehicle path to the safety requirements. Basically, this task is achieved offline on the basis of a mapped environment with known static obstacles. The increase of sensing capabilities enables to consider the implementation of online and embedded path (re)planning to take into account changing environment. One of the challenge is then to carry out this new capability in combination with standard reflexive avoidance strategies. Another one is to provide path (re)planning solutions in accordance with the computational capabilities and time constraints intrinsic to embedded system in AUV applications.

The approach proposed is this paper stands at an intermediate level. The aim is to provide a solution for path planning, with risk integration, compatible with an embedded use, i.e., requiring a limited amount of time. A mapped environment with known static obstacles is assumed. Start and goal position, as well as weights put on the different criteria considered for path planning might be provided by the AUV itself, increasing by this way its autonomy. ?

In the next section path planning problem and the integration of risk is formalized.

2.3 Problem formalization

Path (or motion) planning is the subject of an extensive literature. For an overview see for instance LaValle (2006) or Paull et al. (2013). The general problem statement is to find a *path* between given start and goal *states* (i.e., positions). Generally, this problem is associated with the optimisation of a performance criterion (e.g., path length, travel time) and is potentially subject to constraints (e.g., path smoothness). So, risk integration in path planning can be seen as adaptation of the performance criterion or of the constraints.

There are several approaches to achieve the path planning: e.g., potential fields, sampling-based (randomized) approaches, combinatorial (deterministic) approaches. To avoid the inherent problem of potential fields to get trapped in local minima and the intrinsic weaker of random approaches (near optimal solution if a solution is found), the combinatorial approaches and especially heuristic based algorithm are considered in the following. Therefore, it is assumed that the continuous vehicle environment (i.e., the search space) can be modelled as a graph $G = (S, E)$ where $S = \{s_1, s_2, \ldots, s_n\}$ is the set of possible state (e.g. locations of the vehicle) and $E = \{(s_i, s_j) : s_i, s_j \in S, s_i \neq s_j\}$ the set of possible transitions between states S. All transitions in E can be weighted with several non negative values (e.g. length, travel time, risk).

Path planning corresponds to the search in G of a path from a starting state s_s to a goal state s_g ($s_s, s_g \in S$). A path $p_{s_s,s_k} = [s_s, s_i, \ldots, s_j, s_k]$ defines the successive states that have to been reached to go from the starting state s_s to the state s_k. In the following, path p_{s_s,s_k} is noted p_{s_k} if the starting state is s_s; $p_{s_k}(i)$ denotes the i^{th} states reached and $|p_{s_k}|$ is the number of states in the path. Path planning can be formulated as an optimisation problem

$$p^*_{s_g} = \underset{p_{s_g} \in \mathcal{P}_{s_g}}{\arg\min}\, g\left(p_{s_g}\right) \tag{4}$$

where $p^*_{s_g}$ denotes the path, amongst the set $\mathcal{P}_{s_g}$ of possible paths from s_s to s_g, that minimizes the cost function g. Constraints may be added to equation 4 (e.g., $f^{(i)}(p_{s_g}) - f^{(i)}_{th} < 0$ where $f^{(i)}, \forall i = 1, \dots, n_c$ is a set of cost functions and $f^{(i)}_{th}$ a set of specified maximal threshold). Through this formulation path planning problem may be formulated as the search of *i)* the minimal length path, *ii)* the minimal risk path, *iii)* the minimal length (respectively risk) path subject to constraint of a maximal level of risk (resp. length).

2.4 Cost assessment along a path

We define here several cost functions g depending on the criteria considered for path planning (length and risk).

Path length is defined by the equation (5) where the distance associated to the transition from state $p_{s_g}(i-1)$ to state $p_{s_g}(i)$ is denoted by $d\left(p_{s_g}(i-1), p_{s_g}(i)\right)$.

$$g_{length}(p_{s_g}) = \sum_{i=1}^{|p_{s_g}|-1} d\left(p_{s_g}(i), p_{s_g}(i+1)\right) \qquad (5)$$

Regarding risk, as mentioned before, we focus only on the risk of collision. Moreover it is assumed that whatever is the collision consequence and whenever it occurs, it leads to:

- the aborting of the mission. Therefore only the first collision is of interest.
- to a constant cost C_{C_o} and this without loss of generality ?.

We consider the function $C_o(s_i, s_j)$ which equals 1 if there is collision on the transition from the states s_i to s_j with $(s_i, s_j) \in E$ and equals 0 if not. Let $\Pr\{C_o(s_i, s_j) = 1\}$ be the probability that a collision occurs during this transition. Such quantity is associated to all transitions in the set E of G. Due to the sequencing of displacements, the risk associated to the path p_{s_g} corresponds to the product of the cost C_{C_o} with the probability that:

- the collision occurs during the transition from the node $p_{s_g}(1) = s_s$ to the second state of the path $p_{s_g}(2)$,
- *or*, that the collision occurs during the transition from state $p_{s_g}(i)$ to $p_{s_g}(i+1)$ with $2 \leq i < |p_{s_g}| - 1$ *and* has not occurred during the previous transitions.

Hence:

$$\begin{aligned} g_{risk}(p_{s_g}) = C_{C_o} \Bigg[& \Pr\left\{C_o(p_{s_g}(1), p_{s_g}(2)) = 1\right\} \\ & + \sum_{i=2}^{|p_{s_g}|-1} \Pr\left\{C_o(p_{s_g}(i), p_{s_g}(i+1)) = 1\right\} \\ & \prod_{j=1}^{i-1} \left(1 - \Pr\left\{C_o(p_{s_g}(j), p_{s_g}(j+1)) = 1\right\}\right) \Bigg] \end{aligned} \qquad (6)$$

Note that a conservative approach would be to consider the formulation of equation (7) for risk assessment along the path. This more simple formulation leads to equivalent solutions when the minimal risk path is targeted but is not fully relevant, although conservative, when risk is formulated as a constraint of equation (4) since risk is overestimated.

$$g'_{risk}(p_{s_g}) = C_{C_o} \sum_{i=1}^{|p_{s_g}|-1} \Pr\left\{C_o(p_{s_g}(i), p_{s_g}(i+1)) = 1\right\} \qquad (7)$$

3. PATH PLANNING ALGORITHNS

In this section we briefly present A-Star and Disjkstra's algorithms and propose heuristics in accordance with the different cost functions developed in 2.4. Then we present the hierarchical approach for path planning that can exploit one or other of the two mentioned algorithms. And finally, we give some elements about constrained path planning.

3.1 A-Star, its heuristics, and Dijkstra's algorithm

A-Star and Disjkstra's algorithms are very common solutions to solve minimal cost path planning. The reader is refereed, for instance, to Ferguson et al. (2005) for further description of these algorithms. These two algorithms operates in a similar way but A-Star relies on an estimate, a heuristic h, to drive the graph exploration to the most favourable *areas*. By this way, A-Star may be much less computationally expensive than the Dijkstra's algorithm and leads, as the Dijkstra's algorithm, to an optimal solution provided that the considered heuristic is admissible.

Let $h(s_k, s_g)$, with $s_k, s_g \in S$, an estimate of the cost to reach the state s_g from the state s_k, without presuming a specific path. This heuristic is admissible if it does not overestimate the effective minimal cost h^* to reach the goal state (i.e., $h(s_k, s_g) \leq h^*(s_k, s_g)\ \forall s_k \in S$). Dijkstra's algorithm can be seen as a special case of A-Star when $h(s_k, s_g) = 0\ \forall s_k \in S$ which constitutes an admissible heuristic but not an efficient one.

When length cost is considered, equation (5), an admissible heuristic is

$$h_{length}(s_k, s_g) = \mathrm{d}_e\left(s_k, s_g\right) \qquad (8)$$

where d_e denotes the euclidean distance between the two states.

When risk cost is considered, equation (6), an admissible heuristic is

$$\begin{aligned} h_{risk}(s_k, s_g | p_{s_k}) = C_{C_o} & \Pr_{min}\left\{C_o = 1\right\} \\ \alpha(p_{s_k}) & \frac{1 - \left(1 - \Pr_{min}\left\{C_o = 1\right\}\right)^{N+1}}{1 - \left(1 - \Pr_{min}\left\{C_o = 1\right\}\right)} \end{aligned} \qquad (9)$$

where:

- N is the minimum number of transitions between the state s_k and the goal one:
$$N = \lfloor \frac{d_e\left(s_k, s_g\right)}{d_{max}} \rfloor \qquad (10)$$
with d_{max} the maximum euclidean distance between adjacent states,
- $\Pr_{min}\left\{C_o = 1\right\}$ is the minimum collision probability affected to a transition in the graph G.
$$\Pr_{min}\left\{C_o = 1\right\} = \min_{(s_i, s_j) \in E} \Pr\left\{C_o(s_i, s_j) = 1\right\} \qquad (11)$$

- $\alpha(p_{s_k})$ is a coefficient enabling to take into account the path followed from s_s to s_k

$$\alpha(p_{s_k}) = \begin{cases} 1, \text{ if } |p_{s_k}| = 1, \\ \text{else}, \\ \displaystyle\prod_{i=1}^{|p_{s_k}|-1} 1 - \Pr\left\{C_o\left(p_{s_k}(i), p_{s_k}(i+1)\right) = 1\right\} \end{cases}$$

The third term of equation (9) corresponds to the terms of the geometrical suite generated by the N transitions with a probability of collision equals to $\Pr_{min}\{C_o = 1\}$.

If the considered cost function for risk is expressed by equation (7), an admissible, and more simple, heuristic is

$$h_{risk}(s_k, s_g|p_{s_k}) = N\, C_{C_o} \Pr_{min}\{C_o = 1\} \tag{12}$$

where N and $\Pr_{min}\{C_o = 1\}$ are respectively expressed by equations (10) and (11).

3.2 Hierarchical approach

BlaBla

3.3 Constrained optimization

BlaBla

4. CASE STUDY

The considered use-case is in two dimensions but can be expanded to a higher dimensions (e.g., vehicle depth, vehicle speed) without loss of generality. As mentioned before, only the risk of collision is considered. In addition, the collision cost is assumed to be constant all along the travel. Therefore, for simplicity, risk is reduced to the collision probability in the following (i.e., $C_{C_o} = 1$).

4.1 Environment

Since 2012, the Applied Underwater Robotics Laboratory (AUR-Lab) at NTNU uses an area of Trondheim harbor, not far from Munkholmen, as a testing ground for applications of underwater robots and sensors to marine archaeology. This area is called *The Reference Wreck* due to the presence of a shipwreck dated to late 17th century at 60 meters depth. In the following we use the map of this area to illustrate the proposed approach.

Obstacles inside this area are defined according to 3 different ways:

(1) a first class of obstacles has been defined based on a simple threshold on the depth of the area;
(2) a second class of obstacles has been defined in such a way that it is forbidden to the vehicle to go outside the mapped area;
(3) at last, to make the path planning task more challenging, a third class of additional obstacles has been defined. Dimensions and positions of these obstacles have been randomly generated.

Whatever the way in which obstacle have been defined they have the same "behavior".

A basic cellular decomposition technique into non overlapping cells of simple predefined shapes is used to discretize the search space. The entire cell is considered as an obstacle, and is considered as an invalid state, if it encompasses a part of an obstacle. The cells dimension and connectivity directly yields the size to the graph figuring the search space. In the following, cells are a 0.125m square side, 8-connected neighbourhood, leading to a graph composed by 18495 nodes and 138908 edges.

4.2 Map of collision probability

According to section 2.4 it is assumed that all transitions $(s_i, s_j) \in E$, in addition to the distance between state s_i and s_j ($d_e(s_i, s_j)$), are valuated with the probability $\Pr\{C_o(s_i, s_j) = 1\}$. To determine this probability, a "map" is used where each state (i.e., each cell) $s_i \in S$ is associated to a collision probability denoted $\Pr\{C_o(s_i) = 1\}$ where $C_o(s_i)$ is a function which equals 1 if there is collision in s_i and 0 if not. The probability of collision during a transition between s_i and s_j, $(s_i, s_j) \in E$, is defined as the average of probabilities of collision cells associated to states s_i and s_j

$$\Pr\{C_o(s_i, s_j) = 1\} = \frac{1}{2}\left(\Pr\{C_o(s_i) = 1\} + \Pr\{C_o(s_j) = 1\}\right) \tag{13}$$

The probability $\Pr\{C_o(s_i) = 1\}$, $\forall s_i \in S$ is the result of the aggregation, trough an "or" rule, of collision probabilities to each obstacle j for $j = 1$ to n_O present in the environment

$$\Pr\{C_o(s_i) = 1\} = 1 - \prod_{j=1}^{n_O}\left(1 - \Pr\left\{C_o^{(j)}(s_i) = 1\right\}\right) \tag{14}$$

where $\Pr\left\{C_o^{(j)}(s_i) = 1\right\}$ is the collision probability between the vehicle and the obstacle j when the reference state of the vehicle is s_i.

The function used to determine $\Pr\left\{C_o^{(j)}(s_i) = 1\right\}$ for $\forall s_i \in S$ and all obstacles j is not detailed here. It depends on several parameters, typically RIFs, like: e.g., the shortest distance between the state s_i and the obstacle j, the current intensity and orientation, the visibility and the way the states s_i is crossed (a pass with a diagonal way is longer than a pass with a longitudinal or a lateral way). In practice this function might be tuned on the basis of experts judgement and/or collected data.

In the present application, there is no current, visibility is assumed to be constant and the collision probabilities is homogeneously distributed around single obstacle. The resulting map of collision probabilities is illustrated on figure 2. Obstacles are depicted in black. Start and goal positions have been arbitrarily set around and are depicted white white and red dots respectively.

5. SIMULATION RESULTS

Speak here about the pre processing map and the time required for this step.

5.1 Minimal length path

In this section, the problem of determination of the minimal length path is considered.

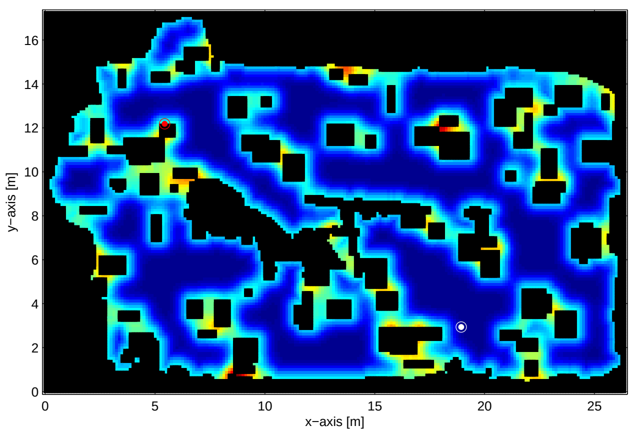

Fig. 2. Map of collision probabilities

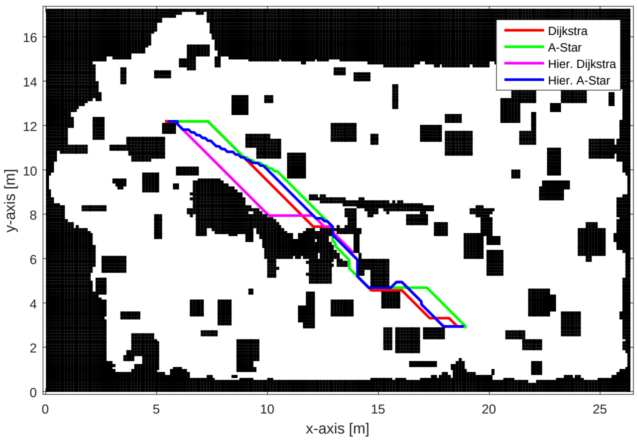

Fig. 3. Minimal length path.

Comments:

- Dijkstra and A-Star algorithms do not lead to the same paths (see figures above) but distances are equal and are the minimal one that can be expected. The fact that these two paths are different explains that the associated risk are slightly different.
- Hierarchical algorithms do not give neither the same paths. The distance of obtained paths are around 1% higher than the optimal one. But, in return, the computational time is reduced by a factor 12 in the case of the Dijkstras algorithm; and by a factor close to 3 in the case of the A-Star algorithm.
- Risk are different for each path. It is consistent with the fact that all paths are different and that the risk was not considered in the path determination process.

Table 1. Minimal length path.

Algorithm	Distance	Risk	Duration
Dijkstra	17.99	1.16E-02	100.00%
A-Star	17.99	1.22E-02	14.08%
Hierarchical Dijkstra	18.16	1.36E-02	7.62%
Hierarchical A-Star	18.16	1.24E-02	2.80%

5.2 Minimal risk path

The problem of determination of the minimal collision risk path is considered in this subsection.

Comments:

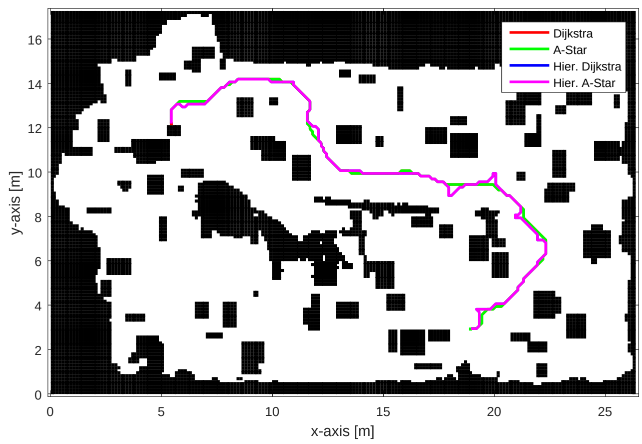

Fig. 4. Minimal risk path.

- Due to the "inefficiency" of the used heuristic, the A Star algorithm does not enable to reduce the computational time as seen in the previous section. But the heuristic is admissible and we can note that the same optimal risk (and actually, the same paths: see figures above) are obtained with Disjkstra and A-Star algorithms.
- Hierarchical approaches are here still relevant and lead to a reduction of the computational time by a factor of around 9 in return of a non-optimality of the obtained paths (+8%).

Table 2. Minimal risk path.

Algorithm	Distance	Risk	Duration
Dijkstra	28.36	2.28E-03	107.89%
A-Star	28.36	2.28E-03	108.88%
Hierarchical Dijkstra	30.89	2.46E-03	12.30%
Hierarchical A-Star	30.89	2.46E-03	13.74%

5.3 Pareto curve

In this section, the problem of determination of the minimal length, respectively risk, path under the constraint of maximal risk, respectively length is considered.

The figure below shows the Pareto curve: minimum risk (i.e., collision probability along the path) depending on the minimum path length. 4 curves are depicted depending of the algorithm used but Dijkstra and A-star curves are overlapped because of the admissibility of the heuristic used with the A-Star algorithm.

The non-optimality of solutions provided by hierarchical algorithms is obvious (see Figure 5) but it is in return of a drastic reduction of the computational time as it can be seen on the Figure 6 where the gain is around a factor 10.

In addition, we clearly see on figure 6 the impact on the computational time of the "inefficiency" of the heuristic associated to the risk: greater is the λ value, greater is the weight put on this criteria and more important is the computational time. This is not only true for the A-Star algorithm but also to the Hierarchical A-Star even if it is less visible on this graph.

At last we can notice an unexpected effect with the decreasing of the computational time for the different curves even if it is more pronounced on the Dijkstra ones.

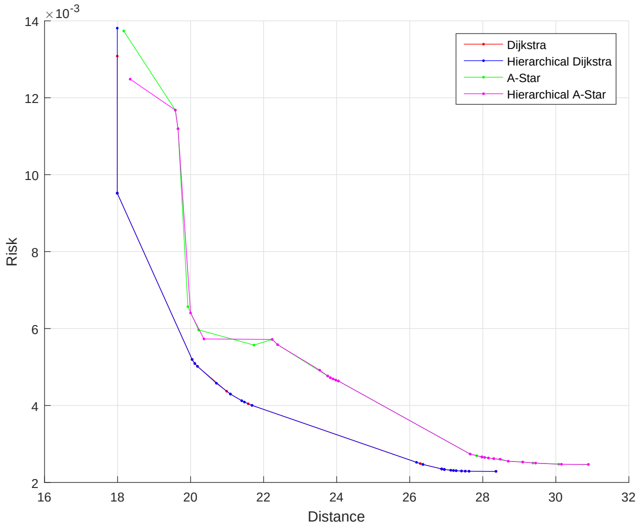

Fig. 5. Pareto curve Risk/Length.

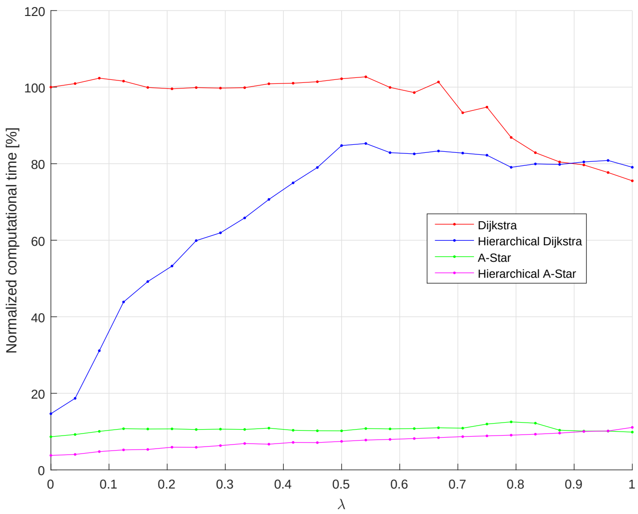

Fig. 6. Computational time for the Pareto curve.

After investigations it appears that the number of visited nodes decreased when the weigh on the risk cost increased. This is due to a "gate" effect.

6. CONCLUSIONS AND FUTURE WORKS

See **?**, **?**, **?** and **?**

REFERENCES

Bae, K.Y., Kim, Y.D., and Han, J.H. (2015). Finding a risk-constrained shortest path for an unmanned combat vehicle. *Comput. Ind. Eng.*, 80(C), 245–253.

Campbell, S., Naeem, W., and Irwin, G. (2012). A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36(2), 267 – 283.

De Filippis, L., Guglieri, G., and Quagliotti, F. (2011). A minimum risk approach for path planning of uavs. *J. Intell. Robotics Syst.*, 61(1-4), 203–219.

Ferguson, D., Likhachev, M., and Stentz, A. (2005). A guide to heuristic-based path planning. In *Proceedings of the International Workshop on Planning under Uncertainty for Autonomous Systems, International Conference on Automated Planning and Scheduling (ICAPS)*.

Greytak, M. and Hover, F. (2009). Motion planning with an analytic risk cost for holonomic vehicles. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, 5655–5660.

ISO (2009). Risk management - principles and guidelines. ISO 31000-2009, International Organization for Standardization, Geneva, Switzerland.

Kaplan, S. and Garrick, J. (1981). On the quantitative definition of risk. *Risk Analysis*, 1(1), 11–27.

Khorrami, F. and Krishnamurthy, P. (2009). A hierarchical path planning and obstacle avoidance system for an autonomous underwater vehicle. In *American Control Conference, 2009. ACC '09.*, 3579–3584. doi: 10.1109/ACC.2009.5160300.

LaValle, S.M. (2006). *Planning Algorithms*. Cambridge University Press, New York, NY, USA.

National Research Council (US) (2005). *Autonomous Vehicles in Support of Naval Operations*. The National Academies Press, Washington, DC.

Oien, K. (2001). Risk indicators as a tool for risk control. *Reliability Engineering and System Safety*, 74, 129–145.

Paull, L., Saeedi, S., and Li, H. (2013). Path planning for autonomous underwater vehicles. In M.L. Seto (ed.), *Marine Robot Autonomy*, 177–224. Springer, Oxford.

Pereira, A.A., Binney, J., Hollinger, G.A., and Sukhatme, G.S. (2013). Risk-aware path planning for autonomous underwater vehicles using predictive ocean models. *Journal of Field Robotics*, 30(5), 741–762.

Rausand, M. (2011). *Risk Assessment: Theory, Methods, and Applications*. Statistics in Practice. Wiley.

Winnefeld, J.A. and Kendall, F. (2011). Unmanned systems integrated roadmap fy 2011-2036. *Office of the Secretary of Defense. US.*