Navigation and collision avoidance of underwater vehicles using sonar data

Ørjan Grefstad
Dept of Marine Technology
NTNU
Trondheim, Norway
orjan1992@hotmail.com

Ingrid Schjølberg
Dept of Marine Technology
NTNU
Trondheim, Norway
Ingrid.Schjolberg@ntnu.no

***Abstract*—Collision avoidance is one of the main challenges in the field of autonomous underwater vehicles (AUV). In this paper a method for detecting obstacles is proposed, using a single-beam mechanically scanning sonar, including planning of an optimal path around the obstacles. Obstacle detection is achieved with an inverse-sonar model updating a vehicle-fixed occupancy grid. A new and obstacle-free path is planned using Voronoi diagrams and Dijkstras algorithm. The path is smoothed using Fermats spiral and a line of sight-guidance system with a time-varying lookahead-distance as guidance. The method is implemented and a full-scale test is performed from IKMs onshore control room on a remotely operated vehicle (ROV) operating at Statoils Snorre B oil field on the Norwegian Continental Shelf. The technology is applicable to ROVs and AUVs in underwater operations.**

***Index Terms*—obstacle detection, collision avoidance, path planning, single-beam sonar**

I. Introduction

Underwater vehicles and specifically remotely operated vehicles (ROVs) are commonly used for Inspection, Maintenance and Repair (IMR) missions in the oil and gas industry. This is a cost-driven industry and advances in automation is key-factor to reduce mission expenses. One of the main difficulties during automated missions is the risk of collision. The collision avoidance challenge is often solved using multi-beam sonars, in a Simultaneous Localization And Mapping approach [6] or with the image recognition based techniques [1]. By using a single-beam sonar the costs can be significantly reduced. The object detection challenge is thougher with single-beam sonars, but can be solved with occupancy grids [4] or with a potential field method [7]. In this paper, occupancy grids are populated using the dynamic inverse-sonar model presented in [9]. The detected obstacles are then used as input to an online re-planning algorithm. This algorithm is motivated by the work presented in [5].

II. System Description

The system developed in this paper is tested on IKMs Merlin UCV, Fig. 1, which is a work-class ROV permanently situated at Statoils Snorre B oil field. However, the method is applicable also to autonomous underwater vehicles (AUVs). The ROV has a Doppler-velocity log aided INS system, which together with a hydro-acoustic positioning system, situated at the rig provides accurate attitude and position information. The hydro-acoustic positioning system is a useful tool in the verification of the system, even though it is not needed for the algorithm.

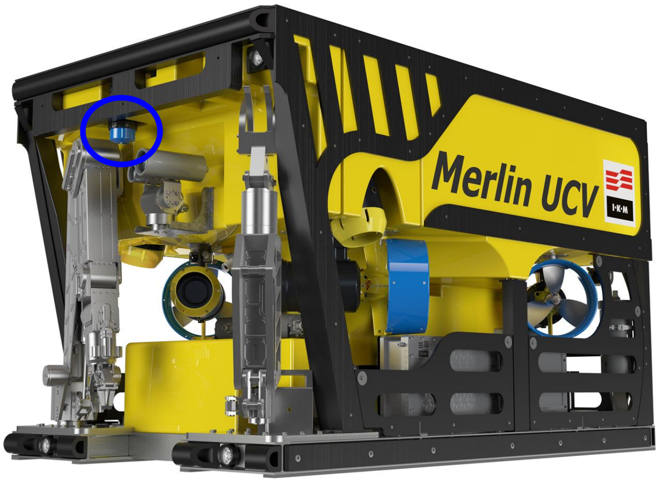

Fig. 1. The Merlin UCV. The sonar position is highlighted in blue.

The ROV is equipped with a Tritech Super SeaKing sonar. The mounting position of the sonar is highlighted in Fig. 1. The sonar is a single beam, mechanically scanning sonar, which utilizes CHIRP technology with frequencies centred at either 325 kHz or 675 kHz. During the the test the sonar was running at a centre frequency of 675 kHz, which gives a horizontal beam width of 1.5° and a vertical beam width of 40°. The step between horizontal scans was 1.8°, which causes a small gap between subsequent scans. An illustration of the horizontal beam width, with three subsequent scans can be seen in Fig. 2 and the vertical beam width, along with the blind-spot can be seen in Fig. 3. The scanning speed of the sonar was approximately 22.5°/s, which means that a full 180° scan takes 8 seconds. During the tests a cruising velocity of 0.5 m/s was selected.

The ROV is operated from IKMs onshore control-room at Bryne, Norway, which makes it accessible for testing of new algorithms. The communication with the ROVs control system and the sonar is performed with UDP-messages following a binary protocol. The position updates are received as NMEA-

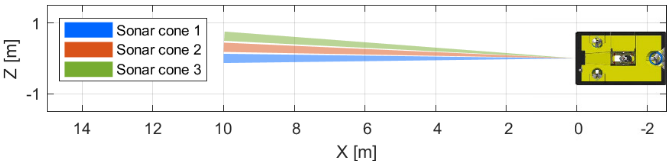

Fig. 2. Top view of the ROV. Beam width of sonar for three subsequent scans.

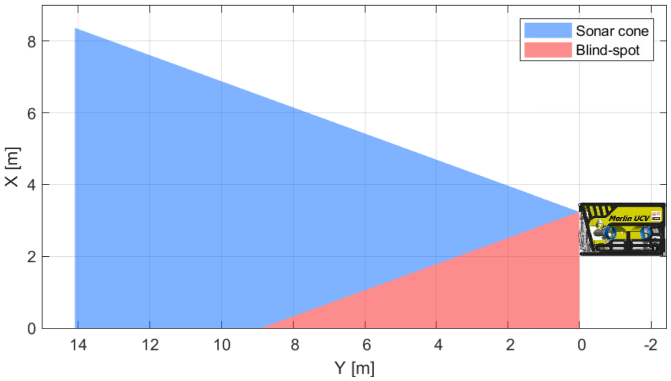

Fig. 3. Vertical beam width of sonar with blindspot underneath.

messages.

III. Methods and Implementation

The collision avoidance system is divided into three modules. First, the obstacles are detected in the object detection module. The detected obstacles are then passed on to the path planning module, which checks for potential collision threats, and calculates a new path if needed. The last module is the guidance system. The three modules and the information flow between them is shown in Fig. 4.

A. Obstacle Detection

A local vehicle-fixed and vehicle centred occupancy grid, $M_(x, y)$ is used. Each cell in $M_(x, y)$ is assigned an initial probability P_{initial} and updated using a log-odds formulation to reduce the computational impact.

Once new measurements are available from the sonar the grid is updated using one of two possible methods. If there is an obstacle present, the grid cells intersecting the sonar cone is updated according to the dynamic inverse-sonar model presented by [9]. When no obstacles are present, the grid is updated with an inverse-sonar model described thoroughly in [8]. The grid cells used in these updates has a lower resolution $M_(x, y)$, to reduce the computational load.

Both methods are dependent on translation from the sonar output to a possible obstacle. This is done with a dynamic threshold, where the threshold value applied to the raw data is tied to the dynamic range selected by the operator on the sonar display, rather than a fixed value.

The grid is stored as a matrix of floating point values, which is similar to the way digital images are represented. This similarity makes it easy to use an image processing technique called affine transformation to transform and rotate the grid in an effective manner. The translation is performed each time a position/relative position measurement is obtained, but very small changes are saved for the next iteration.

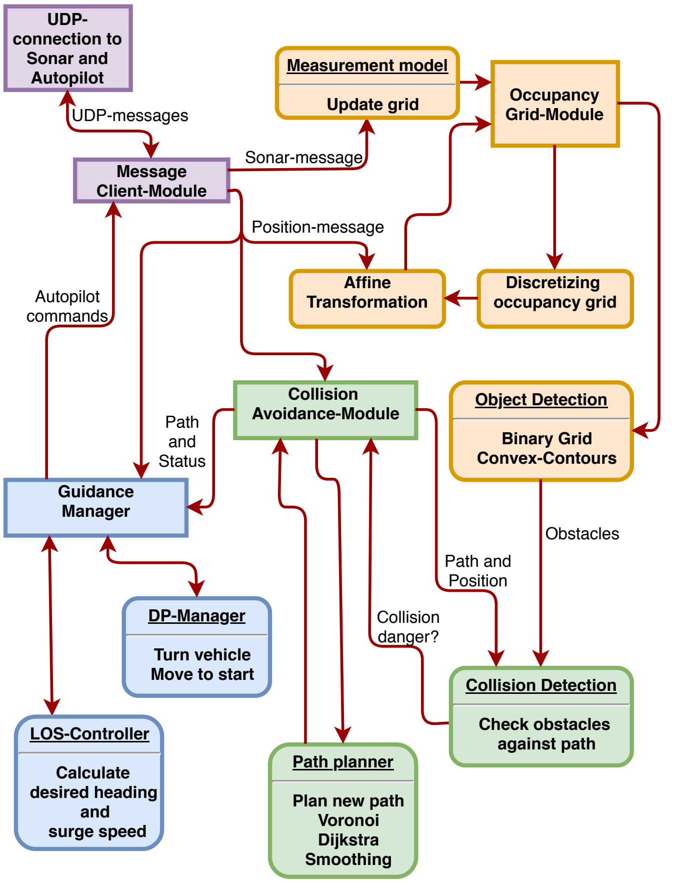

Fig. 4. Flowchart of the object detection and collision avoidance process. The sharp-cornered boxes indicates a module, while the ones with rounded corners represent functions/methods for the corresponding module.

The object detection procedure starts by applying a threshold to $M_(x, y)$, resulting in a binary grid $M_(x, y)^B$, which is useful for finding the contours of the obstacles. In this process, very small clusters of occupied cells are regarded as false positives and thus disregarded. The rest of the contours are regarded as possible objects. Another image processing technique, called dilation is then applied to the contours, such that a safety margin of a fixed distance is introduced. The contours of the obstacles are then approximated as convex shapes to reduce the number of input points to the path planning module. A new binary grid $C_(x, y)^B$ is then constructed with the convex shapes. $C_(x, y)^B$ is the basis for the incident-angle calculation used in the dynamic inverse-sonar model and the path planning module. The incident angle calculation finds the part of the obstacle line that intersects the middle of the sonar cone, and the angle is calculated as the angle between these two lines.

B. Path Planning

The path planning module is split into two parts, where the first part checks the current path for collisions and the second part calculates a new path if necessary.

The collision check is archived by drawing the path, with a safety-margin onto a binary grid, with the same resolution as $C_(x, y)^B$. A binary *AND*-check is then performed to detect any overlaps by the detected obstacles and the path. This check is relatively cheap computationally-wise and is run four times a second if no collision is detected. If a collision is detected the module will move on to the second step.

The re-planning uses the convex contour-approximations from $C_(x, y)^B$ as generator points for a Voronoi diagram (VD), as done by [5]. A VD is a tool for dividing space into a set of regions, such that the borders of each region are as far away from each generator point as possible. The borders make the basis for the new path, but the unfeasible paths have to be removed. This is done in the same manner as the collision check. The borders do not contain the position of the vehicle, or the destination point, which are inserted into the appropriate regions and connected to the surrounding vertices. The end point is selected as the first waypoint on the preplanned path that is outside the region of the occupancy grid. In addition another point is inserted in front of the vehicle, such that the transition between paths can be as smooth as possible.

From the set of feasible paths, the shortest one is found using Dijkstras algorithm. Dijkstras algorithm provides the optimal path between two given nodes in the VD, but due to the VD having a high density of nodes, the optimal path may contain unnecessary nodes. These nodes are removed in a two-step process: First all nodes that are close to colinear are removed. The next step is to remove nodes giving a larger clearance to the obstacle than necessary.

The new path still consists of straight line-segments, which is a problem for underactuated vehicles, and unsuitable for fully-actuated vehicles as a complete stop is needed. A suitable method for ensuring a continuous curvature path is presented by [2], where Fermats spiral is used for making a smooth path. The path is then discretized to make it suitable for the guidance system.

In some cases, with a very high obstacle density, the system is not able to find a feasible path. In these cases, the last resort is to halt the vehicle to a full stop, and afterwards slowly rotate the vehicle to find a feasible path. The direction chosen for rotation is based on the mean value of the starboard and port parts of the occupancy grid.

C. Guidance

The Merlin UCV has a control system with several possible configurations, where commands are sent as binary commands over the network.

In this work the dynamic positioning (DP) system and a heading/surge-controller are used. A LOS-controller is implemented, following the procedure described in [3, Ch. 10], with some modifications. The lookahead-distance is implemented as a lookahead-time, such that lookahead-distance is a time-varying parameter dependent on the vehicle velocity. The region of acceptance is reduced if the curvature is large, resulting in close waypoints.

The setpoint for the surge-controller is also time-varying. When a new path is calculated, the guidance-system will calculate the curvature for each path-segment-corner and subsequently choose the correct velocity for the turn. As the vehicle will need some time to reduce the velocity to the turn-velocity a break-distance is implemented in the same manner as the region of acceptance.

The guidance system tries to make the transition between paths as smooth as possible, which means that as long as the vehicle is on the new path, or sufficiently close to it, the new path is started immediately. If these conditions are not satisfied, the vehicle will stop and enter DP-mode to reach the start of the new path.

IV. Results

The system is first tested for object detection capabilities, then the complete system is tested with pre-planned paths, going straight through obstacles.

A. Obstacle Detection

In the first test the ROV was flown by a pilot, from the garage, around a subsea-module and back again. The results from this test can be observed in Fig. 5. A snapshot of the detected obstacles is taken every 10 seconds and then plotted in the same figure. The altitude of the ROV is controlled by the pilot and thus it appears to be flying straight through some obstacles, such as the toolstand and Module A. In these cases,

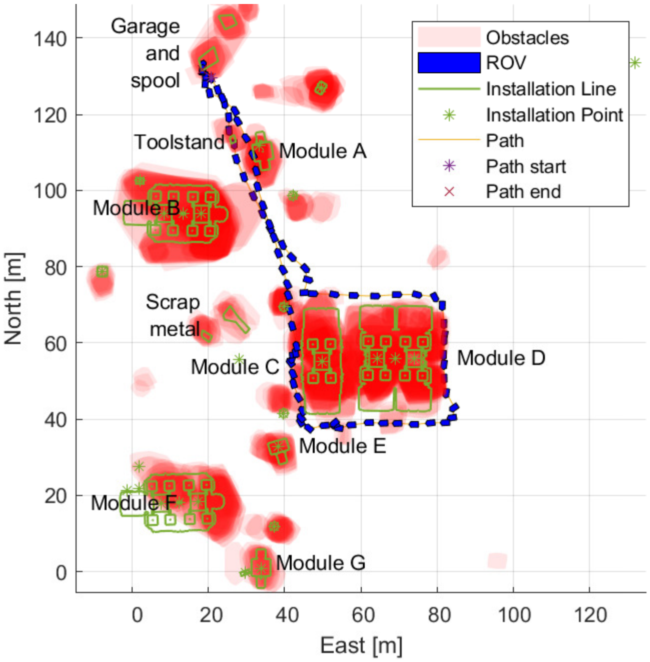

Fig. 5. Obstacle detection test. A map of the subsea structures are drawn in as green lines and points.

the ROV was flying above them. The ROV is flying at a mean altitude of approximately 2 meters until it reaches Module C, then the mean altitude is increased to 3.5 meters for the remainder of the flight. It can be observed that all obstacles are clearly detected. It should be noted that there is some drift in the position, which can be observed by looking at the tool stand, which has changed location between the beginning and the end. The detected obstacles appear larger than the obstacles on the map, and this is due to a safety-margin of two meters. It should also be noted that the algorithm has some problems accurately detecting the south-side of Module D. This is likely due to the high altitude and proximity to the module, which causes the sonar to miss it completely. The blindspot underneath the vehicle can be seen in Fig. 3.

B. Collision Avoidance

Several tests on collision avoidance are performed. All of the tests start close to the garage, with a preplanned path. The ROV was never able to reach the destination point due to constraints with the tether and ongoing operations.

The results of the first test can be observed in Fig. 6-8. In Fig. 6 the inital path can be observed. Soon after the vehicle started on the inital path, Module B is detected and a new path is calculated from the vehicles position to the end of optimization point. It should be noted that this path takes the vehicles current velocity into account, and thus the first turn is quite sharp. In Fig. 7 a new obstacle is detected and a new path is calculated according to the known obstacles. The toolstand and Module A is not detected due to the close proximity and the altitude of the vehicle. The sharp turn between Module C and D is due to an earlier recalculation of the path. It should be noted that most of the obstacles in the sonars field of view are detected. The only undetected obstacle is Module A (see Fig. 5), which is due to proximity and altitude. The system calculates a short deviation from the planned route, and is able to avoid the obstacle before the new path rejoins the old one.

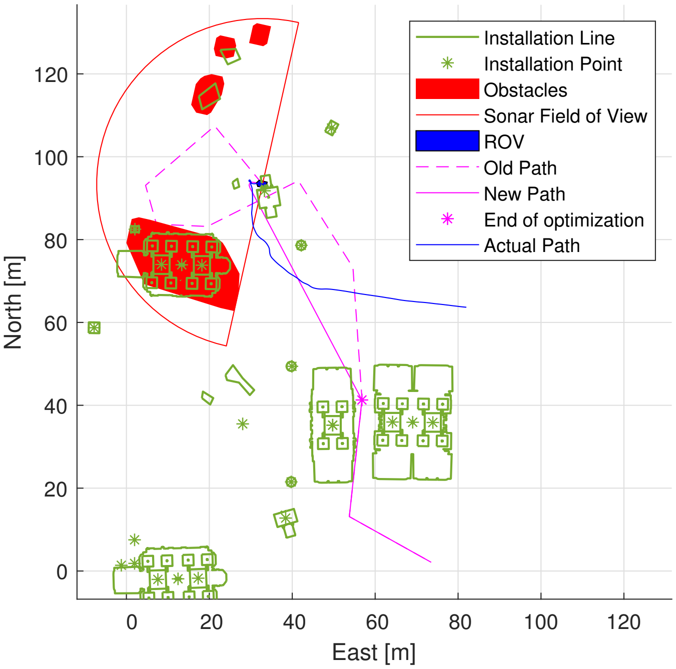

Fig. 6. Collision avoidance test. A map of the subsea structures are drawn in as green lines and points. The initial path is marked as old path

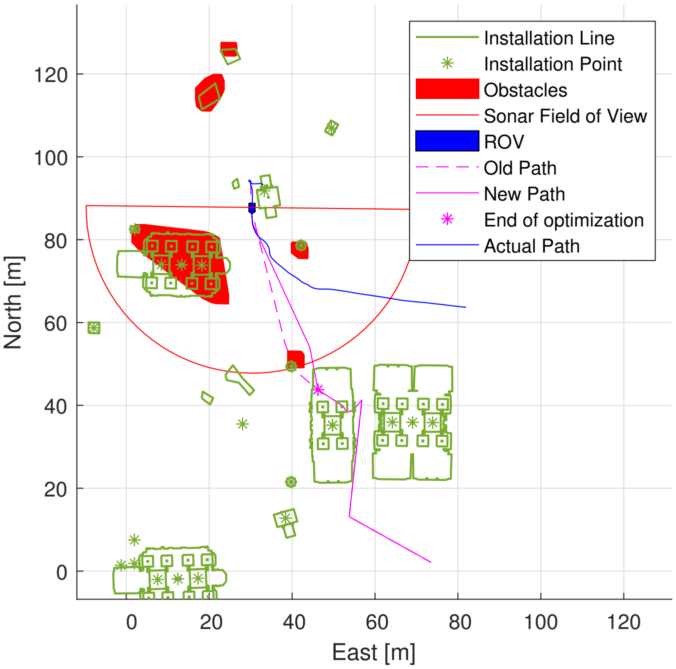

Fig. 7. Collision avoidance test. A map of the subsea structures are drawn in as green lines and points.

Right after the point where the optimization stops, between Module 2 and 3, there is a sharp bend in the path. This is the result of an earlier path recalculation. Since it is outside the optimized region, it is not smoothed away before it is closer to the optimized region.

In Fig. 8 the path is once again recalculated around the obstacles. It should be noted that the path is further away from the obstacle than necessary. There is also some smoothing problems.

After the tests were done the algorithm was improved to allow for better smoothing between the old and the new path, as well making the new path closer to the obstacles. When this improved algorithm was run on the test data shorter and smoother paths were selected. This can be seen in Fig. 9 where the test data from Fig. 8 is run through the improved algorithm.

The second test is made a few days later, with a different configuration of the sonar. The sonar configuration is decided by the pilot, and has implications for the effectiveness of the algorithm. When looking at Fig. 10 it can be observed that the obstacles appear smaller. This is due to the change in the sonar receiver gains causing less distinguishable return echoes. The initial and time variable receiver gains can be hard to adjust. The system still manages to calculate a route around the obstacles.

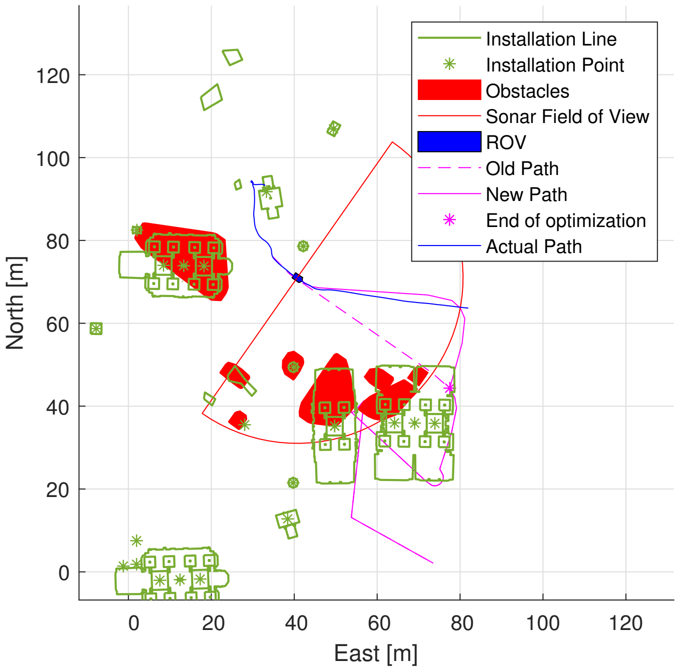

Fig. 8. Collision avoidance test. A map of the subsea structures are drawn in as green lines and points.

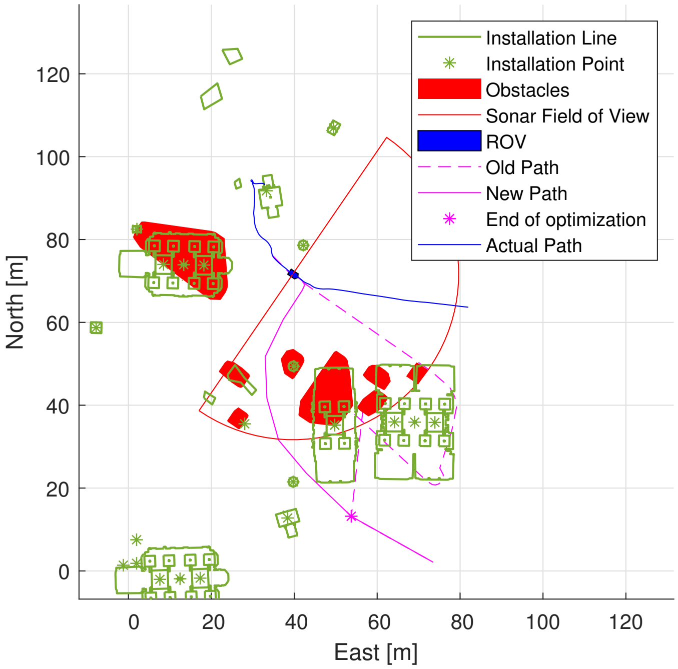

Fig. 9. Collision avoidance test with improved path planning algorithm.

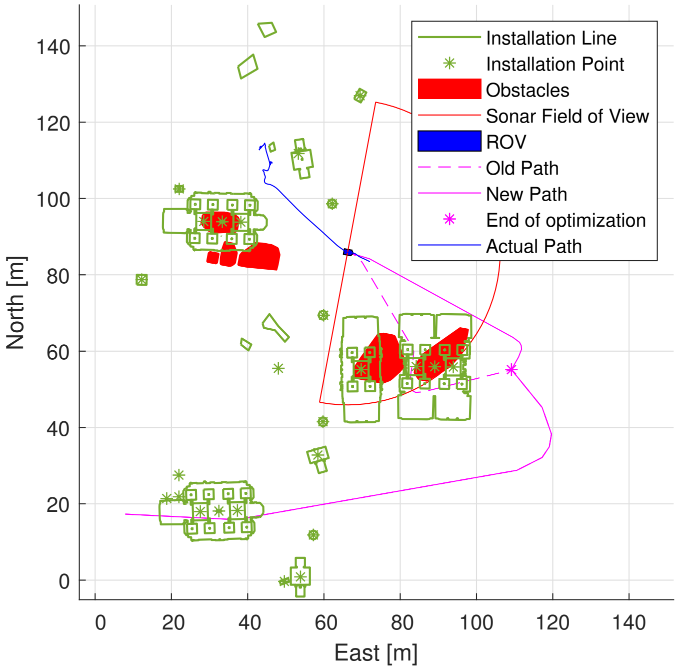

Fig. 10. Second collision avoidance test. A map of the subsea structures are drawn in as green lines and points.

V. Discussion and Conclusions

This paper has presented an effective method for detecting obstacles using a single-beam sonar, as well as an effective way of calculating a new obstacle-free path. A vehicle-fixed local occupancy grid has several advantages over a global map. The complexity is reduced, and thus calculation time is significantly less. The major advantage is that the grid can be completely decoupled from global positions. The changes in position can then come from only a doppler-velocity log and a compass.

The full-scale test showed that the system is capable of detecting the obstacles in its path in an effective manner. A new and obstacle-free path is calculated and executed. The full-scale test showed that a mechanically-scanning singlebeam sonar is adequate for detecting and avoiding obstacles.

The dependence on the sonar configuration is a problem that should be addressed, either through making the detection algorithm independent of the sonar configuration, or by making an algorithm that can automatically tune the sonar.

This system only considers paths in the two-dimensional space. Voronoi diagrams can easily be extended to work in a 3D-space, but this will most likely not be possible without the use of extra sensors, such as a camera, an extra sonar or a 3D-sonar to extract information about the height of obstacles. Adding data from a camera will also improve the detection capabilities at a close range. The methods are applicable to any underwater vehicle equipped with a single beam sonar, as the field of view can be reduced to increase the scanning velocity.

VI. Acknowledgements

The authors would like to thank IKM Technology for valuable help during the development and testing phase. We would also thank Equinor for the possibility of testing at their subsea production areas.

References

[1] B Braginsky and H Guterman. Obstacle Avoidance Approaches for Autonomous Underwater Vehicle: Simulation and Experimental Results. *IEEE Journal of Oceanic Engineering*, 41(4):882–892, 2016.

[2] Mauro Candeloro, Anastasios M Lekkas, Asgeir J Sørensen, and Thor I Fossen. Continuous Curvature Path Planning using Voronoi diagrams and Fermat's spirals. *IFAC Proceedings Volumes*, 46(33):132–137, 2013.

[3] Thor I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. Wiley, 2011.

[4] Varadarajan Ganesan, Mandar Chitre, and Edmund Brekke. Robust underwater obstacle detection and collision avoidance. *Autonomous Robots*, 40(7):1165–1185, 2016.

[5] Anastasios M Lekkas. *Guidance and path-planning systems for autonomous vehicles*. PhD thesis, Trondheim, 2014.

[6] Albert Palomer, Pere Ridao, and David Ribas. Multibeam 3D underwater SLAM with probabilistic registration. *Sensors (Switzerland)*, 16(4), 2016.

[7] Franco J. Solari, Alejandro F. Rozenfeld, Villar A. Sebastian, and Gerardo G. Acosta. Artificial potential fields for the obstacles avoidance system of an AUV using a mechanical scanning sonar. In *2016 3rd IEEE/OES South American International Symposium on Oceanic Engineering, SAISOE 2016*, Grupo INTELYMEC, Centro de Investigaciones en Física e Ingeniería Del Centro (CIFICEN), Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Facultad de Ingeniería, Universidad Nacional Del Centro de la Provincia de Buenos Aires (UNCPBA),, 2017. Institute of Electrical and Electronics Engineers Inc.

[8] Sebastian Thrun. Learning Occupancy Grid Maps with Forward Sensor Models. *Autonomous Robots*, 15(2):111–127, 2003.

[9] M Zhou, R Bachmayer, and B Deyoung. Mapping for control in an underwater environment using a dynamic inverse-sonar model. In *2016 OCEANS MTS/IEEE Monterey, OCE 2016*, pages 1–8, Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. John's, NL, Canada, 2016. Institute of Electrical and Electronics Engineers Inc.