https://doi.org/10.31256/Nc9Mp9N

Shared autonomy for robotic inspection

Jonatan Scharff Willners, Shida Xu, Tomasz Łuczyński, Sean Katagiri, Joshua Roe,
Sen Wang and Yvan Petillot
Institute of Sensors, Signals and Systems
Heriot-Watt University
Edinburgh, UK
{j.scharff_willners, sx2000, t.luczynski, s.katagiri, joshua.roe, s.wang, y.r.petillot}@hw.ac.uk

***Abstract*—In this paper, we describe how an autonomy engine can work together with an operator to perform human-machine collaborative inspection of offshore structures using shared autonomy. The approach is driven by the robot's autonomy, which can make suggestions of what the autonomy engine considers to be desirable actions to fulfil a mission. The operator can at any time inject its own action, choose from the robot's generated suggestions or allow the robot to continue fully autonomously until the next user input. The approach removes the need for constant monitoring and can enable a single operator to control multiple vehicles at once, as the robot will ensure safe and collision-free operation while performing actions. The approach has been tested in various underwater environments with successful results, including offshore wind turbines.**

***Index Terms*—Marine robotics, autonomy, mapping, navigation, human-robot interaction**

I. Introduction

The maritime domain is a treacherous environment, the continuous increasing pressure while descending into the depths of the ocean makes it an undesirable and dangerous place for humans to operate it. Due to this and to the growing expansion of offshore renewable energy infrastructure such as wind turbines – Remote Operated Vehicles (ROVs) and autonomous underwater vehicles (AUVs) are seeing more usage every year. These vehicles are getting increased capabilities in terms of autonomy, duration, and perception *etc*. However, for many objectives such as inspection of assets including harbour walls, ship hulls and wind turbine foundations – most of the work is still carried out using human operators controlling the ROV with very limited, if any, autonomy involved. In this paper, we present a system that can relieve the operator of much work, by instead of controlling the robot – working together with it to accomplish a mission by using shared autonomy. This means, that the robot can operate completely independently without input from an operator while displaying actions that would be beneficial for completing the mission. The operator can at any point order the robot to choose a different action or to do a completely different, operator-defined action instead. This way the operator does not need to focus on just a single robot but can be placed onshore, controlling multiple platforms simultaneously as the low-level control and planning are handled internally by the robot.

The paper shows, to our knowledge, the first application of shared autonomy with fully autonomous capabilities used for marine robots exploring and inspecting offshore assets.

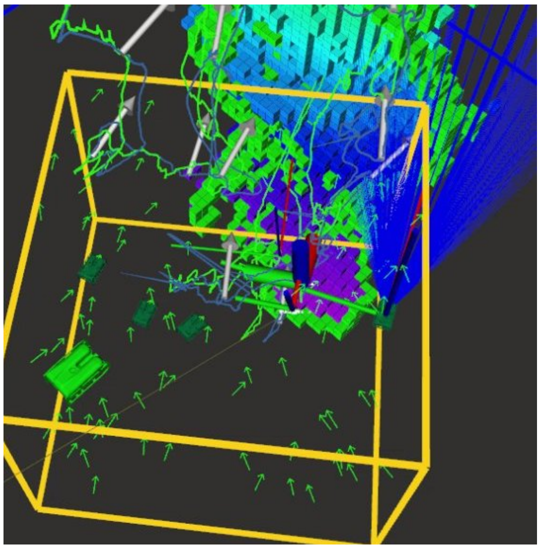

Fig. 1. An example of an instance showing the suggested actions as green models of the vehicle. The environment is shown as an octomap, with the bright green voxels representing the frontier. One suggested action shows the simulated sensor as blue rays. The yellow box is the local region for sampling, where the green arrows are evaluated - the green vehicle models are interactive actions that the operator can select to go to at any point. The scenario is from an offshore inspection of a wind turbine.

II. Shared Autonomy

The autonomy system presented in this paper has the objective to explore an unknown structure. The approach is based on using an next-best-view (NBV) [1] selection process with the current map of the environment represented as an octomap [2]. The octomap is a discrete representation using voxels. In our implementation of a voxel can be marked as occupied, empty, unknown or as a frontier. The frontier represent the boundary between the occupied and what is yet to be explored (unknown). The frontier defines where new information can be obtained about the environment. Using this map, we simulate the robot's sensing capabilities at multiple viewpoints (a viewpoint is a potential pose along with the simulated perception sensor's data) with 3 different approaches: 1) randomly sampled viewpoints in the region around the robot for a local position, 2) the boundary of the occupied map, enlarged by the maximum sensing distance as global viewpoints and 3) we cluster the frontier voxels and

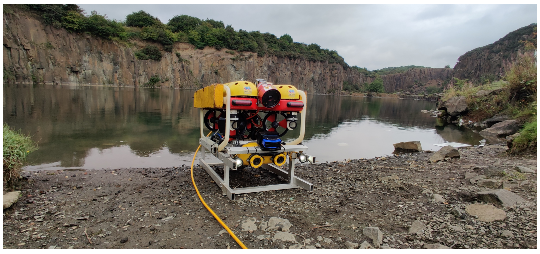

Fig. 2. An ROV equipped with AUIP mounted below.

estimate the normal from the clusters' centres. At a user-defined distance along this normal a viewpoint is generated with an orientation facing towards the cluster centre. These approaches of defining viewpoints give us 3 sets of viewpoints. Each set is evaluated to generate a utility score for each viewpoint. The utility score (seen in eq. (1)) is based on a weighted (w) sum of 5 individual scores (each score (u) is in the range $[0, 1]$); 1) the number of frontier voxel seen in the simulated sensor, 2) the number of occupied voxel seen, 3) the distance from the current position, 4) the distance from the closest previous visited position and 5) the difference between optimal sensing range and closest distance to an occupied voxel in the field of view. Each individual score is normalised to the range and is then multiplied by a weight. The weights are normalised, and hence the final utility is in the range $[0, 1]$, as in eq. (2).

$$\mathcal{U} = \sum_{i=1}^{5} (u_i w_i) \quad (1) \qquad \sum_{i=1}^{5} (w_i) = 1.0 \quad (2)$$

The n viewpoints for each set with the highest utility score are presented as potential actions for the robot to take. The approach is similar to NBV, but instead of selecting the highest scoring viewpoint as the next waypoint, it proposes a set of actions. If no action is selected by the operator within a time limit, the robot will switch to fully autonomous mode and move to the highest scoring viewpoint. While in the fully autonomous mode, it will continue to generate suggestions for the operator to select, hence at any given time the operator can select an option to override the current action of the robot and switch back to shared autonomy mode. The sequence: generate suggestion - select suggestion (either autonomously or from the operator) is repeated until the operator decides that the mission is finished or until the autonomy engine cannot provide any suggestions that are deemed to yield enough new information about the environment to continue the exploration. The decision making for the autonomy is based on a behaviour tree [3] and the implementation is using Robot Operating System (ROS) [4].

III. Experimental Evaluation

The system has been deployed in various scenarios, including a tank, a quarry and an offshore wind farm. The presented experimental evaluation in this paper will focus on the exploration of a wind turbine foundation using a

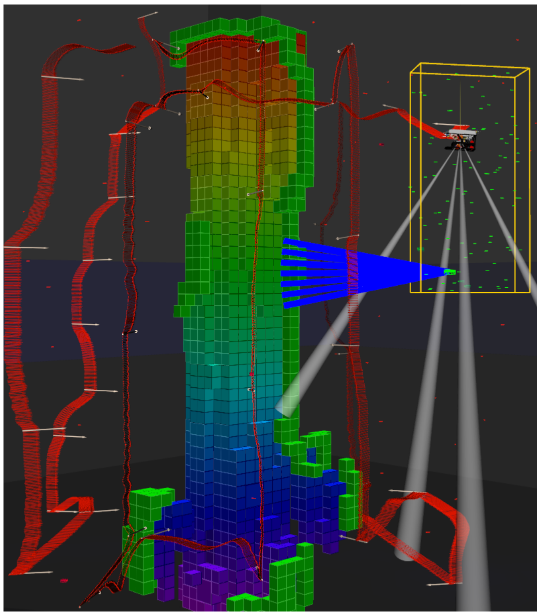

Fig. 3. The shared autonomy system successfully achieving complete visual coverage of a wind turbine foundation in simulation.

Saab Seaeye Falcon equipped with AUIP. AUIP is a custom payload containing stereo camera system, inertial sensors and additional computational power [5]. The ROV can be seen in Fig. 2. The windfarm is located in the North Sea, which has strong tidal currents, as such the operation was limited to the region behind the structure which had more shelter from the current. An example of how the system sees the environment and presents suggestions to the operator can be seen in Fig. 1. The inspected structure is gravity-based and sits at a depth of roughly 45 metres. The shared autonomy helped the operator to focus the exploration at regions which were feasible due to the current, as a fully autonomous frontier based exploration approach would go around the structure to complete the inspection, which would not have been possible with the conditions during the trials. To show the system in operation when not constrained by the tidal current, we simulate it using UUV Simulator [6]. In Fig. 3 it can be seen how the system successfully achieves full sensor coverage the structure.

References

[1] N. Palomeras, N. Hurtos, E. Vidal, and M. Carreras, "Autonomous exploration of complex underwater environments using a probabilistic next-best-view planner," IEEE Robotics and Automation Letters, 2019.

[2] A. Hornung, K. Wurm, M. Bennewitzm, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," Autonomous Robots, 2013.

[3] M. Colledanchise, and P. Ögren, "Behavior Trees in Robotics and AI: An Introduction," http://arxiv.org/abs/1709.00084.

[4] M. Quigley, et al., "ROS : an open-source Robot Operating System," ICRA Workshop on Open Source Software 2009.

[5] J. Scharff Willners, et al., "From market-ready ROVs to low-cost AUVs," IEEE Oceans 2021.

[6] M. M. M. Manhães, S. A. Scherer, M. Voss, L. R. Douat, and T. Rauschenbach, "UUV Simulator: A Gazebo-based package for underwater intervention and multi-robot simulation," IEEE Oceans 2016.