OTC-30694-MS

# Legged Robots for Autonomous Inspection and Monitoring of Offshore Assets

Milad Ramezani, Martim Brandao, Benoit Casseau, Ioannis Havoutis, and Maurice Fallon, Oxford Robotics Institute, University of Oxford



## Abstract

Inspection and monitoring of assets are repetitive and expensive tasks and have higher risk when facilities are located offshore. Robotics holds the promise of improving the efficiency and safety of such platforms by allowing inspection and continuous remote monitoring of difficult-to-access facilities.

Legged robots, such as quadrupedal robots, are promising machines to achieve this goal: they have high maneuverability both indoors and outdoors, they are designed for accessing and navigating facilities that are built for humans (e.g. stairs, step-over piping, narrow passageways) and can carry a variety of sensors targeted at inspection and monitoring tasks.

In this paper we introduce our approach for autonomous inspection of oil & gas platforms using legged robots. Our approach is being developed as part of the ORCA Hub (Offshore Robotics for Certification of Assets), a UK robotics research hub. We envision a highly autonomous robotic system that conducts inspections with minimal intervention by human operators. The robot can navigate through facilities, as shown in Figure 1, accomplishing crucial tasks such as 3D mapping, monitoring of thermal build-up using thermal cameras, pressure sensing and also using color cameras to detect people and to carry out general visual inspection. We demonstrate and evaluate the system's perception, locomotion and inspection capabilities on a training facility that realistically simulates an oil rig at the Fire Service College, Moreton-in-Marsh, UK and an industrial area in the Offshore Renewable Energy Catapult Facility, Blyth, UK. We show the result of both autonomous and real-time teleoperated missions, and analyze the accuracy and efficiency of the system.

## Introduction

Regular inspection of offshore assets is crucial for the oil and gas industry. The oil and gas facilities can potentially be dangerous places due to the exposure to toxic liquids or gasses and damage or leakage of pipelines. Hence, they not only pose a risk to the personnel's health but also damage might go unnoticed if not regularly monitored, possibly leading to long-term damage to the environment.

**Figure 1—Top: Fire Service College, Moreton-in-Marsh, UK is a training facility for first responders. Bottom: ANYmal operating in different parts of the facility.**

Robots are becoming increasingly popular for inspection of environments including various physical and chemical hazards. Drones, for instance, can fly over industrial sites and provide a stream of images/videos online. Flying robots are often used for reconnaissance and patrolling. Recent works such as (Chermprayong, 2019) have enabled drones to be used in more applications such as repairing oil pipes in an aerial repair mission. Despite the impressive progress in the technology of maneuverability, navigation and stability of such robots, their stability in harsh weather conditions still remains a challenge. In addition, small-scale aerial robots are limited in payload, which makes them incapable of carrying accurate navigation sensors such as laser scanners. Large-scale drones, on the other hand, have less maneuverability for flying through narrow passages.

One of the robots specifically built for autonomous inspection of oil and gas sites is the VIKINGS robot (Merriaux, 2018). Exploiting two articulated caterpillar tracks, the robot can climb up and down staircases or pass over obstacles. In addition, the robot is equipped with a variety of sensors and a telescopic mast aiming at measuring difficult-to-access areas. The robot exhibited an impressive performance at the ARGOS Challenge (TOTAL, 2018).

Legged robots have drawn attention in the industry in the past decade. Stasse et al. provided a proof of concept targeting at introduction of bipedal robots in an industrial environment -an aircraft factory (Stasse, 2019). Schwartz et al. employed a low-cost humanoid robot for floor tiling by developing an end-effector to lift the tiles and place them on the floor precisely (Schwartz, 2016).

Quadrupedal robots can use their four limbs to create wide bases of support and to better distribute load over more contact points. They exhibit higher stability than humanoids, especially when traversing harsh terrain. Benefiting from a low center of gravity and having more contact with the ground during motion, quadrupeds will be suited for use in challenging environments from underground in mines and tunnels to overground in onshore and offshore facilities including narrow passages and stairs. Gehring, for example, demonstrated how a quadruped robot can be used for routine autonomous inspection (Gehring, 2018).

To utilize robots in autonomous inspection tasks, they must not only be equipped with a highly accurate navigation system, but also need to be able to walk reliably and circumvent obstacles to avoid collisions. Therefore, in this work, we discuss two components of the robot platform: localization and mapping, and motion planning. For localization and mapping, we propose a Simultaneous Localization and Mapping (SLAM) system described in the following section. It can build accurate 3D maps of a facility and to identify where the robot is within the map at each point in time. For path planning, we propose a system which enables our robot to plan routes and quickly replan when obstacles appear.

We first overview our robot's mapping and path planning subsystems. Our robot is equipped with accurate sensors for localization and mapping, which we integrate with the kinematics of the robot to simultaneously map the facility and localize within it, without the requirement of prior information. Information from this map is used to plan the robot's movement through the facility's rooms, corridors and outdoor areas. The map uses a pose-graph representation of the facility which allows it to discover new routes between locations and obtain shortest paths to goals. Another aspect of our research analyses the terrain to identify its traversability and can automatically switch between different specialized locomotion controllers depending on the terrain (flat ground, rough ground, stairs etc.). This allows the robot to save battery consumption by only using power-intensive controllers when necessary -while keeping safety and robustness at top priority. The integration of our SLAM-system with our path planning is the first contribution of this paper. Finally, we demonstrate our test procedures in a mock-up oil rig and an industrial indoor area, namely a hangar, to display the usefulness of our system in oil rig and industrial settings, the major contribution of the paper, before conclusions are presented in the final section.

## ANYmal Quadruped Robot

We use the ANYbotics' quadruped robot version B of "ANYmal" (Hutter, 2016) which weighs 33 kg and is capable of carrying payloads up to 10 kg at a speed of 1 m/s. Our quadruped integrates a multitude of high frequency sensors, namely an Inertial Measurement Unit (IMU), joint encoders and torque sensors. Lower frequency sensors include cameras and a LiDAR. Our SLAM-system uses the IMU, joint encoders as well as LiDAR as the primary sensor. The robot has 3 joints per leg, leading to a total of 12 degrees of freedom and allowing for a variety of locomotion styles or "gaits".

### Walking Control System

The ANYmal has a combination of specialized "controllers" to command the robot to walk in a given direction. In this work we use the term "controller" to refer to an algorithm that is responsible for computing and executing a motion in all of the robot's motors in order to make progress in a certain direction (e.g. forward, 30 degrees rotation, etc.). In particular we currently use a fast and dynamic "trotting" controller, which is specialized for fast locomotion on flat ground (Gehring 2016), and a slower "walking" controller which is specialized for locomotion on staircases and rough terrain, making use of the vision/depth sensors for careful footstep placement (Fankhauser, 2018).

The trotting controller uses an optimization-based method with dynamics constraints to compute joint torques that satisfy a desired body velocity. The walking controller, on the other hand, involves multiple layers of optimization: one for footstep placement using terrain statistics, and one for torque computation given the desired footstep and static stability constraints.

We use onboard depth cameras to build a dense and precise 3D model of the terrain directly in front of the robot. We then use this model to estimate the roughness of the terrain and therefore choose when to switch between controllers (Brandao, 2019). When using the "walking" controller, our method plans exact footstep placements from this model, so that enough clearance from obstacles and holes on the ground guarantees a safe execution of a direction of motion.

## Mapping System

Our navigation system leverages a map of the facility, which is generated in advance by teleoperating the robot across the facility. The robot constantly localizes itself within the facility (i.e. recognizes its position and orientation within the map) by an algorithm that matches what is seen by the robot's laser sensor to the stored map, and fuses this information with other sensors such as inertial units and motor data. The system also leverages this map to plan long-range paths to goals that can be anywhere within the facility (e.g. a different room in a different floor). The system uses onboard sensors for local adjustments to these paths that take into account what is directly visible to the robot (e.g. terrain changes or new objects).

### Simultaneous Localization and Mapping

Autonomous operation of robots requires accurate localization and mapping. SLAM algorithms are used to construct and update a map of an unknown environment while concurrently estimating the motion of the robot. To build a SLAM system, the robot needs to be equipped with sensors such as laser scanners or cameras.

Visual SLAM (vSLAM) algorithms track visual features in a sequence of images or video frames to perform localization. By simultaneously estimating the 3D position of the features, vSLAM provides a map of the environment. However, the map is quite sparse if only salient features are used (Leutenegger, 2013). To use the maximum visual information, so-called direct techniques (Caruso, 2015) have gained in popularity that instead of extracting features from images, they compute depth (Newcombe, 2011) in an incremental manner and estimate the camera's trajectory from direct alignment of the images/frames. Nonetheless, we do not focus on the use of cameras in this work since vSLAM algorithms are highly dependent on illumination conditions of the environment. We would like to have a SLAM system which works reasonably well in windowless and unlit environments that is likely in offshore facilities, thus our primary sensor for localisation and mapping is a LiDAR scanner.

### Point Cloud Registration

LiDAR-SLAM systems often leverage the Iterative Closest Point (ICP) algorithm (Besl, 1992). Given an initial pose of the sensor corresponding to the capture of the point clouds, the ICP algorithm minimizes the distance between the correspondences of two LiDAR clouds (i.e. the measured cloud and the map) iteratively. By minimizing the distance between correspondences, the relative transformation between consecutive clouds are precisely estimated. It provides the 6 DOF pose of the robot, i.e. 3D position and orientation with respect to a frame called odometry frame.

To obtain a precise initialization for the sensor pose, we exploit the integration of kinematic information of the robot with the inertial measurements. Each leg of our quadruped contains three joint encoders on knee, shoulder and hip. The measurements of an IMU, which is attached to the robot's main body, together with the joint information are tightly coupled within a Kalman filtering method (Bloesch, 2017). This sensor fusion, called kinematic-inertial odometry, operates at a high rate, equal to the frequency of IMU or encoders, and can provide the necessary pose prior for ICP registration.

### Pose Graph Optimization

As described so far, our LiDAR SLAM system is only an odometry system that incrementally estimates the current pose of the robot with respect to the previous pose. This process, which is known as dead reckoning in navigation, is prone to drift over time due to cumulative errors. To solve this issue, we use a pose graph which consists of nodes corresponding to the robot's pose and edges connecting the nodes. The consecutive edges are the transformation constraints estimated by our ICP algorithm and form the front-end component of our SLAM structure. Benefiting from the pose graph, if the robot has a loop in its trajectory and revisits a part of the environment that it explored in the past, a new edge is added to the pose graph, called a loop-closure, which poses a new constraint on a pair of poses. This forms the back-end component of the SLAM architecture.

Having added the odometry and loop-closure constraints to the pose graph, the robot's poses are estimated in a global optimization which removes the drift and generates a globally consistent map. In our recent work (Ramezani, 2020), we fully describe our SLAM-system and its evaluation in simulation and real-life settings.

With such a system in place, we are able to teleoperate the robot in the environment while the SLAM-system is running on the robot to generate the entire map of the environment on-the-fly. The generated map can later be used for the conduction of autonomous or teleoperated inspection missions.

### Localization within a Prior Map

Having generated the map of the environment, we are able to define a series of walking goals for the robot to execute (to carry out a complete survey mission). Execution of these goals relies on the robot accurately localizing itself within the map. To do so, we align the current information of the LiDAR with the prior map generated in advance. This way, the robot achieves the predefined poses and can execute a mission made up of survey tasks such as reading dials or taking photos of hot spots with a thermal camera.

## Navigational Path Planning

From the map generated by the SLAM system, which consists of a large point cloud, we generate a simplified 3D model of the environment for fast path planning. We first reconstruct a precise 3D mesh from the point cloud using a Ball pivoting algorithm (Bernardini, 1999), and then generate a simplified "navigation mesh" for path planning using the approach described in (Brandao, 2020). The approach consists of voxellization and wavefront expansion as proposed in (Mononen, 2014). Navigation meshes consist of a set of simple connected polygons representing the walkable areas of the environment. This kind of structure is also used in commercial computer games for extremely fast path querying -even at a kilometer scale. In our system, planning a path from one point to another in an oil rig facility typically takes less than 1ms. Path planning in navigation meshes is efficient because these structures consist only of a set of walkable triangles and a graph representing their connectivity.

Each triangle is also associated with a choice of controller and a cost-of-transport (i.e. cost per distance traveled). The cost of a 1 meter path over a "walking" area is therefore different from the cost over a "trotting" area. Path planning in this domain then corresponds to running A* search on the underlying graph, which provides a sequence of triangles to walk over. This is followed by the computation of a minimum-distance path that connects the triangle sequence. See (Brandao, 2020) for more details.

Due to the assignment of a controller to each triangle in the navigation mesh, the paths obtained using the navigation mesh take into account the controllers that need to be used in each point of the facility, so as to be as fast or as safe as possible. For example, for speed-focused missions, we set a cost-of-transport proportional to controller speed, and thus the method prefers long paths on flat terrain to short paths that go in and out of a building through stairs. Figure 2 shows an example of such behavior, where the longer path returned by our algorithm can be 4 times faster than one that does not take controller-choice into account. This is, again, because our "trotting" controller is considerably faster – up to 8 times – than our walking controller.

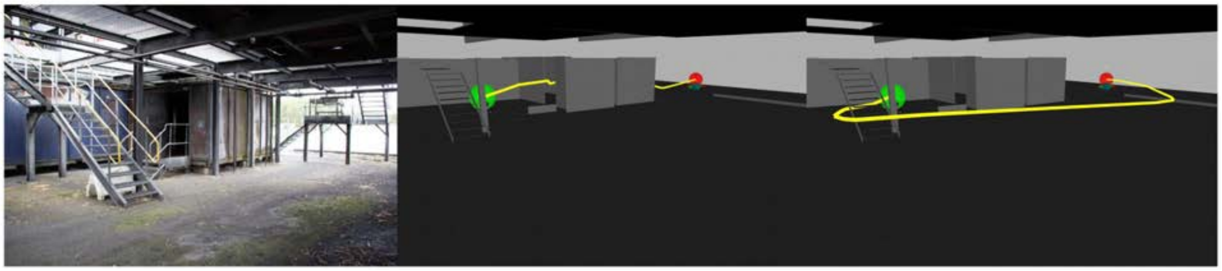

Figure 2—The advantage of controller-reasoning within long-range planning. Left: a part of the facility. Middle: a least-distance path goes through buildings which requires more dangerous stair-climbing. Right: the controller-aware path is longer but 3.7 times faster to execute as it uses only trotting.

**Replanning in the presence of obstacles**

Sometimes new objects are added to a facility or they are temporarily left in places that can obstruct passages or make a terrain untraversable. Our vision-based controller-switching scheme described in the beginning of the section can adapt to local changes that require a new choice of controller or different footstep placements. However, it is not able to deal with complete blockages of passages that require a completely new path (e.g. through a different door and sequence of rooms) and would fail in such a situation.

For this purpose, we employ an extra sub-system that is responsible for verifying whether locally following a long-range path is possible. Using the onboard depth and LiDAR sensors we check whether any collisions exist along the planned long-range path. If there is a collision, or it is not possible to find a set of footsteps that track the path without collision, then we update the geometry of our global navigation mesh. We do this by adding a virtual obstacle to the mesh on the appropriate location, thus triggering the use of an alternative path that avoids the problematic section of the map. Typically, this will consist of navigating a different set of corridors and rooms to arrive at the same goal position.

**Experimental results**

Our experiments were carried out in a mock-up oil rig which is part of the Fire Service College located at Moreton-in-Marsh, UK and in a hangar including power generators located at Blyth, UK with the focus on localization and mapping as well as path planning.

## Localization and mapping system

In one scenario, the robot was teleoperated in the Fire College test area, both indoors and outdoors. Figure 3 shows the map which was generated online. The pose of the robot is indicated by axes where red is the x and green is the y axes of the base frame. The red strips connecting some of the robot poses are the loop-closures detected between the current pose of the robot and a pose stored in the pose graph. This experiment, which took 40 minutes, demonstrates that our SLAM system has minimum drift due to the fact that all the structures/features are accurately aligned with each other even after long excursion of the robot in the test field.

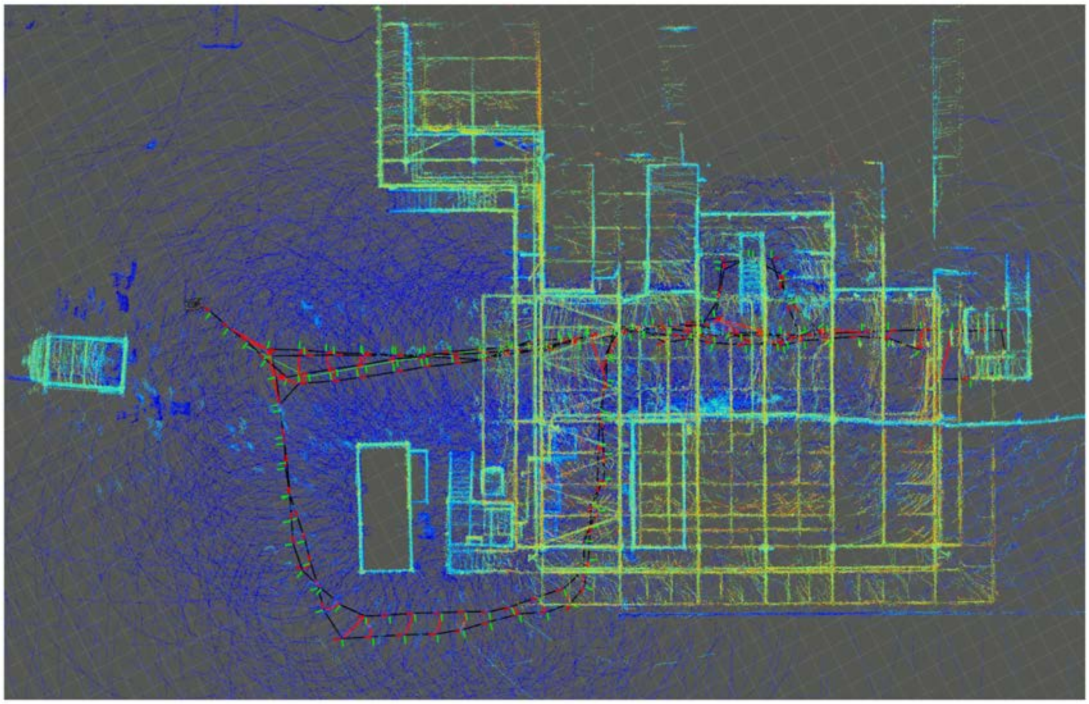

Figure 3—The map of the Fire Service College generated by the SLAM system. Red lines indicate loop closures between the current pose of the robot and a pose captured in the past.

To evaluate the accuracy of the localization, we tracked the robot with a laser tracker (Leica-TS16) at 10 Hz whenever the robot was in the line of sight. Figure 4 compares the trajectory of the robot estimated by the SLAM approach versus the ground truth. For the portion of the trajectory in which the ground truth is available, we computed the Root Mean Squared Error (RMSE) which is 6 cm in this case, verifying that the drift on this experiment is less than 0.06% over the course of the trajectory. This demonstrates that our SLAM system can produce an accurate map of the environment.

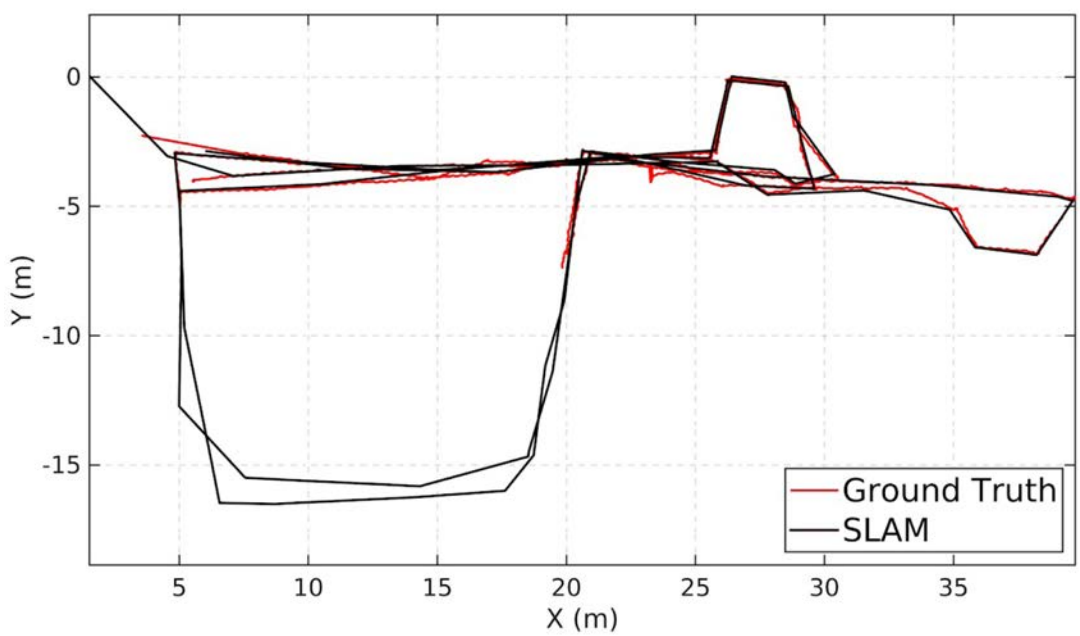


Figure 4—The trajectory of the robot estimated by the SLAM algorithm (black) and compared with the ground truth determined from a laser tracker (red, only available when the robot is in line of sight of the tracker).

### Path Planning System

After generating a global map of the environment using the SLAM system, we created a mesh of the environment based on the map. In the Fire Service College (Figure 5), the challenges were climbing up and down staircases and passing through doorways from one room to another. The mesh classifies the operating environment into untraversable areas, areas without any mesh (Figure 5, top-left), traversable areas that the robot can trot through quickly (indicated as light blue in Figure 5, top-left) and the areas that the robot needs to walk carefully (indicated as red).

As seen in Figure 5, the robot started trotting from a circle marked by a chalk on the ground. It succeeded in switching from fast trotting to slow walking once it approached to the staircase. After passing through the narrow passages and entries, it exited the structure accomplishing the entire path while succeeding in switching between the controllers.


**Figure 5—The robot is commanded to navigate through a sequence of connected shipping containers in the Fire College oil rig facility. It trots on flat ground and then uses a walking controller for safe navigation on stairs and narrow passages.**

In another scenario, the objective was to demonstrate the capability of the system to replan the path to find an alternative when an original path is suddenly blocked by obstacles. The robot first planned the path based on the original mesh (yellow path in Figure 6, top-left) and started executing the path. We then blocked one of the passages using two metal barrels (indicated by a red circle in Figure 6, top-left). Once the robot recognized the route blockage using the LiDAR measurements, it succeeded in replanning the path (Figure 6, top-right). Also, in the middle of this path, a steel barrier with a height of 10 cm was located as seen in Figure 6, down-left. When the robot approached this barrier, it managed to switch from fast trotting to slow walking. Finally the robot arrived at the goal and accomplished the task of measuring temperature of a pipe with the thermal camera.

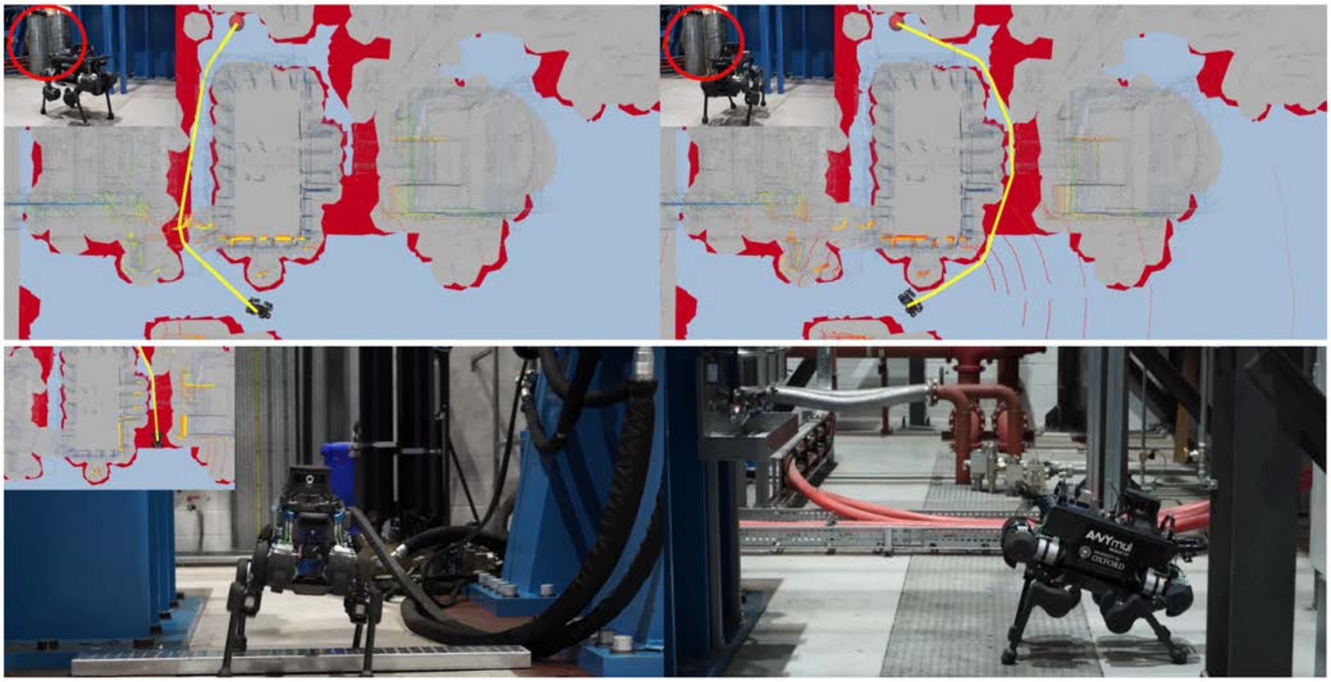

**Figure 6—Upper Left: The robot initially plans a faster path as enter the corridor on the left. Upper Right: The system identifies a new obstacle and; Lower Left: starts executing a new path through that is feasible (to the right). Lower Right: The robot reaches the inspection destination (a heat source) to which it points its thermal sensor.**

## Conclusions

In this work, we introduced a robot system which is able to carry out inspections in industrial facilities. We developed a SLAM system with LiDAR as the primary sensor. The SLAM system creates an accurate global map of the environment. Our path planning system can provide sufficient maneuverability of the robot to overcome obstacles and barriers, or to replan if a difficult obstacle is sensed in the way of the robot. Field experiments demonstrated the success of our system. In the future, we would like to enable the robot to locate itself within the map automatically. This would be beneficial for tasks taking more than one day and the robot needs to continue operation by localizing globally in the map.

## Acknowledgments

We would like to thank our colleagues in Oxford Robotics Institute (ORI) for their help in conducting the experiments. This research is supported by the Innovate UK-funded ORCA Robotics Hub (EP/R026173/1) and the EU H2020 Project MEMMO. M. Fallon is supported by a Royal Society University Research Fellowship.

## References

Chermprayong, P., Zhang, K., Xiao, F., and Kovac, M. 2019. An integrated delta manipulator for aerial repair: A new aerial robotic system. *IEEE Robotics & Automation Magazine* **26** (1): 54–66.

Merriaux, P., Rossi, R., Boutteau, R., Vauchey, V., Qin, L., Chanuc, P., Rigaud, F., Roger, F., Decoux, B., Savatier, X. 2018. The VIKINGS Autonomous Inspection Robot: Competing in the ARGOS Challenge. *IEEE Robotics & Automation Magazin* **26** (1): 21–34.

Stasse, O., Morsillo, F., Geisert, M., Mansard, N., Naveau, M. and Vassallo, C. 2014. Airbus/future of aircraft factory HRP-2 as universal worker proof of concept. IEEE-RAS International Conference on Humanoid Robots: 1014–1015.

Schwartz, M. 2016. Use of a Low-Cost Humanoid for Tiling as a Study in On-Site Fabrication. 36th Annual Conference of the Association for Computer Aided Design in Architecture-Posthuman Frontiers: Data, Designers, and Cognitive Machines, ACADIA: 214–223.

Gehring, C., Fankhauser, P., Isler, L., Diethelm, R., Bachmann, S., Potz, M., Gerstenberg, L., Hutter, M. 2019. ANYmal in the Field: Solving Industrial Inspection of an Offshore HVDC Platform with a Quadrupedal Robot. Conference on Field and Service Robotics.

Hutter, M., Gehring, C., Jud, D., Lauber, A., Bellicoso, C.D., Tsounis, V., Hwangbo, J., Bodie, K., Fankhauser, P., Bloesch, M. and Diethelm, R. 2016. Anymal-a highly mobile and dynamic quadrupedal robot. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): 38–44.

Gehring, C., Coros, S., Hutler, M., Bellicoso, C. D., Heijnen, H., Diethelm, R., Bloesch, M., Fankhauser, P., Hwangbo, J., Hoepflinger, M., and Siegwart, R. 2016. Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot. *IEEE Robotics Automation Magazine* **23** (1): 34–43.

Fankhauser, P., Bjelonic, M., Bellicoso, C. D., Miki, T., and Hutter, M. 2018. Robust rough-terrain locomotion with a quadrupedal robot. IEEE International Conference on Robotics and Automation: 1–8.

Brandao, M., Fallon, M., and Havoutis, I. 2019. Multi-controller multi-objective locomotion planning for legged robots. IEEE/RSJ International Conference on Intelligent Robots and Systems.

Leutenegger, S., Furgale, P., Rabaud, V., Chli, M., Konolige, K. and Siegwart, R. 2013. Keyframe-based visual-inertial slam using nonlinear optimization. *Proceedings of Robotics Science and Systems (RSS)*.

Caruso, D., Engel, J. and Cremers, D.. Large-scale direct slam for omnidirectional cameras. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS): 141–148.

Newcombe, R.A., Lovegrove, S.J. and Davison, A.J. 2011. DTAM: Dense tracking and mapping in real-time. International conference on computer vision: 2320–2327.

Besl, P. J., McKay, N. D. 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intellignece* **14** (2): 239–256.

Bloesch, M., Burri, M., Sommer, H., Siegwart, R. and Hutter, M. 2017. The two-state implicit filter recursive estimation for mobile robots. *IEEE Robotics and Automation Letters* **3** (1): 573–580.

Ramezani, M., Tinchev, G., Iuganov, E., Fallon, M. 2020. Online LiDAR-SLAM for Legged Robots with Robust Registration and Deep-Learned Loop Closure. IEEE International Conference on Robotics and Automation (ICRA).

Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. 1999. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics* **5** (4): 349–359.

Brandao, M., Aladag, 0. B., Havoutis, I. 2020. GaitMesh: controller-aware navigation meshes for long-range legged locomotion planning in multi-layered environments. *IEEE Robotics and Automation Letters*.

Mononen, M 2014. Recast navigation. Google Project: http://github.com/recastnavigation/recastnavigation.