

## Repositório ISCTE-IUL

---

Deposited in *Repositório ISCTE-IUL*:

2023-02-07

Deposited version:

Accepted Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Marques, F., Santana, P., Guedes, M., Pinto, E., Lourenço, A. & Barata, J. (2013). Online self-reconfigurable robot navigation in heterogeneous environments. In 2013 IEEE International Symposium on Industrial Electronics. Taipei: IEEE.

Further information on publisher's website:

10.1109/ISIE.2013.6563831

Publisher's copyright statement:

This is the peer reviewed version of the following article: Marques, F., Santana, P., Guedes, M., Pinto, E., Lourenço, A. & Barata, J. (2013). Online self-reconfigurable robot navigation in heterogeneous environments. In 2013 IEEE International Symposium on Industrial Electronics. Taipei: IEEE., which has been published in final form at <https://dx.doi.org/10.1109/ISIE.2013.6563831>. This article may be used for non-commercial purposes in accordance with the Publisher's Terms and Conditions for self-archiving.

---

### Use policy

Creative Commons CC BY 4.0

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a link is made to the metadata record in the Repository
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

---

# Online Self-Reconfigurable Robot Navigation in Heterogeneous Environments

Francisco Marques<sup>1</sup>, Pedro Santana<sup>1,2</sup>, Magno Guedes<sup>3</sup>, Eduardo Pinto<sup>1</sup>, André Lourenço<sup>1</sup>, José Barata<sup>1</sup>

<sup>1</sup>CTS-UNINNOVA, New University of Lisbon

<sup>2</sup>DCTI, University Institute of Lisbon (ISCTE-IUL)

<sup>3</sup>R&D Unit, INTROSYS SA

**Abstract**—This paper presents a robot navigation system capable of online self-reconfiguration according to the needs imposed by the various contexts present in heterogeneous environments. The ability to cope with heterogeneous environments is key for a robust deployment of service robots in truly demanding scenarios. In the proposed system, flexibility is present at the several layers composing the robot’s navigation system. At the lowest layer, proper locomotion modes are selected according to the environment’s local context. At the highest layer, proper motion and path planning strategies are selected according to the environment’s global context. While local context is obtained directly from the robot’s sensory input, global context is inspected from semantic labels registered off-line on geo-referenced maps. The proposed system leverages on the well-known Robotics Operating System (ROS) framework for the implementation of the major navigation system components. The system was successfully validated over approximately 1 Km long experiments on INTROBOT, an all-terrain industrial-grade robot equipped with four independently steered wheels.

## I. INTRODUCTION

Over the years, autonomous robot navigation has been receiving a considerable amount of attention. This interest owes to the relevance navigation has in what regards the actual deployment of robots for the execution of real world tasks. Navigation in off-road environments [1], urban environments [2], [3], and other environments, are rather well studied topics. Typically, these previous studies show that different environments demand for different navigation strategies. Thus, a robot operating in heterogeneous environments needs to be able to determine the current environmental context and then smoothly switch to the proper navigation strategy accordingly. This context aware strategy switching process is still an open problem, to which this paper contributes with a novel solution.

This paper addresses the problem by proposing a navigation system capable of online self-reconfiguration in the face of context changes. Self-reconfiguration occurs through the activation, deactivation, and parameterisation of a set of pre-existing navigation-related modules. To foster scalability and reusability, the self-configuration process is built on the well-known Robotics Operating System (ROS) [4], which provides well-specified interfaces to lookup and interact with the navigation system’s composing nodes. In the proposed system, self-reconfiguration is triggered and guided according to both local and global context information. Local context information is obtained directly from the robot’s sensory input. Global context is obtained from a set of semantic labels registered off-line on geo-referenced maps. Thus, this approach relies on the increasingly widespread availability of labelled



(a)

Fig. 1. The all-terrain robot, INTROBOT [5], used in the field trials.

satellite imagery and on the increasingly ubiquity of high throughput communication channels.

In its current state, the proposed system considers two semantic labels, namely, *open space* and *narrow space*. Based on these two labels, planning and localisation strategies are selected and parameterised. Moreover, obstacle avoidance parameters and representations of the robot’s footprint are also adapted according to the local context. With such flexibility, the system is able to trade-off, in a context-aware way, between accuracy of the planned motions and computational cost. As a result, the complexity of the planner matches the complexity of the environment, which is key to enable a proper management of computational and energetic resources.

The use of context to enable the self-reconfiguration of control systems has recently received some attention. Most notably, overall scene statistics have been used to predict the presence of a given environment, which, in turn, triggered a given set of behaviours [6] or localisation strategies [7]. These solutions require that the robot learns online a visual classifier prior to a proper self-reconfiguration. Conversely, the solution presented in this paper considers the use of semantic maps overlaid on large scale satellite imagery. This allows the system to respond promptly without expending the cost of learning. Nevertheless, both approaches are complementary, rather than mutually exclusive. In addition, by not relying on distal information (e.g, vision) to determine the global context, the system is able to operate even in the presence of strong perceptual aliasing situations.

Although there are also some examples on the use of overhead imagery in the support of navigation outdoors, these do not cover the wider problem of selecting proper navigation strategies. See for instance the case in which overhead imagery was shown to be useful in the task of learning classifiers for navigation cost assessment [8]. Another related topic is the one of topological mapping, in which easily distinguishable regions of the environment are the pivots of the employed world representation. By not relying directly in sensory feedback to infer global context, the system proposed in this paper is immune to perceptual aliasing, which is a well-known problem in topological mapping. Rather than an alternative to topological mapping, the proposed system is complementary. Concretely, it can provide topological mapping processes with priors on the global context, thus reducing the complexity of handling strong perceptual aliasing situations.

To validate the proposed system, field trials were conducted with an industrial-grade robot, the INTROBOT<sup>1</sup> (see Fig. 1). INTROBOT is a 0.8 m × 1.5 m × 0.7 m all-terrain robot equipped with four independently steered wheels and its sensor package includes a tilting 2-D laser scanner, a stereo vision head, an inertial measurement unit (IMU), and a GPS device. Throughout the 895 m long field trials, the proposed system showed to be able to self-reconfigure in such a way that collision-free and goal-directed smooth transition between environments was ensured.

The paper is organised as follows. Section II presents the proposed system. Then, experimental results are presented in Section III. Finally, conclusions and future research avenues are drawn in Section IV.

## II. SELF-RECONFIGURABLE NAVIGATION SYSTEM

The goal of making the navigation system self-reconfigurable is to allow picking the set of navigation, localisation, and control components best suited to the situation at hand. This proposal builds on the observation that no single component is able to cover the wide set of situations a robot may face. This section starts by describing the set of state-of-the-art components that are to be selected online by the system, according to the context. Then, the section ends with the description of the context aware selection mechanism itself.

A key enabler for the proposed system was the emergence of meta operating systems for robotic controllers, such as the Robotics Operating System (ROS) [4]. With ROS, a robot's control system is defined as a dynamic graph of nodes that interact via well-specified messages. Moreover, ROS being a product of community development, it provides a set of state-of-the-art components as ROS nodes. This paper exploits the dynamic nature of ROS graphs in order to include in a seamless way context aware adaptation of the control system. In this work, the control system is assumed to have at its disposal ROS nodes for robot localisation, map building, SLAM, obstacle avoidance, motion control, motion planning, and path planning (see Fig. 2). In this context, self-reconfiguration means setting a ROS graph with the nodes and, respective connectivity, that better suits the needs of the current context.

For the sake of completeness, an overview of the considered nodes is provided next.

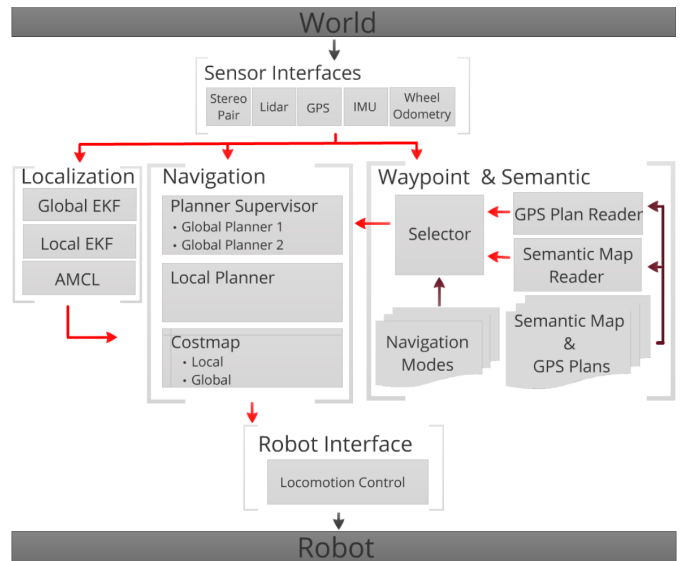


Fig. 2. Overview of the different nodes and interactions within the navigation system. Each bracket contains a category of nodes. Red arrows represent data exchange via ROS messages. Black arrows represent low-level sensor/actuator messages. Dark red arrows represent parameterisation messages.

### A. Obstacle Avoidance ROS Node

Obstacle avoidance holds the specific ability of taking the robot towards a given goal location without colliding against any obstacle, i.e., according to the local context observed from the robot's onboard distal sensors (e.g., laser scanner). In the proposed system, this is attained with the trajectory roll out method [9]. In short, it finds the set of feasible arc trajectories, given the robot's kino-dynamic constraints and obstacles distribution. From the feasible trajectories, one is selected based on an objective function that trade-offs speed, distance to obstacles, bearing with respect to the goal location, among other.

In the proposed system, the original trajectory roll out method was adapted in order to allow online self-reconfiguration of the robot's footprint representation and locomotion capabilities (see Fig. 3). This is important because INTROBOT is able to execute four different locomotion modes [5], such as moving with a double Ackerman geometry or turning around its geometric centre in place.

To properly reconfigure the locomotion capabilities of the robot, a hysteresis-based cost of changing among locomotion modes has been included in the aforementioned objective function. The goal is to reduce motion time and energy consumption when the local context induces an oscillatory behaviour. For instance, the presence of noise in the localisation process may take the robot to intermittently consider to be too far or too close from the desired path to be followed. As a result, without the presence of the hysteresis, the robot would spend time and energy by intermittently switching between an Ackerman-based locomotion mode (when aligned with the path) and a turning in place locomotion mode (when unaligned with the path).

<sup>1</sup><http://www.introbot.pt/>

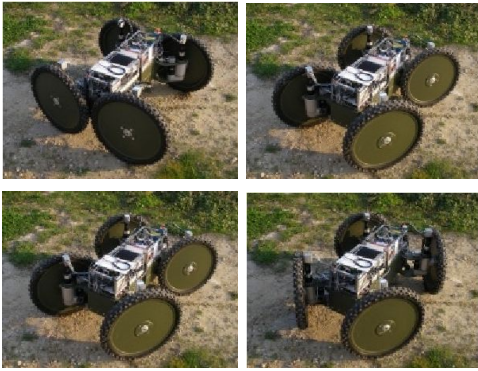


Fig. 3. INTROBOT's four locomotion modes [5].

### B. Motion and Path Planning ROS Nodes

Obstacle avoidance alone, being a local method, does not ensure progression towards the final goal. To circumvent the myopic nature of local methods, the output of a motion planner or of a path planner is usually used to bias the local method towards the goal location. This way, *cul-de-sac* situations and others are properly handled. While *motion planning* allows explicitly accounting for the kino-dynamic properties of the robot in the planning process, *path planning* abstracts robot motion as linear trajectories between waypoints. Depending on the situation, one may be better suited than the other; that is, the speed-accuracy trade-off must be solved according to the context. For instance, a path planner usually suffices for producing fast motion outdoors. Conversely, a robot that needs to pass through a door may need a fine motion planning procedure. The high computational cost of motion planning is compensated by the fact that when passing through doors robots are allowed to move slowly.

In this work, two planning strategies are considered. The first is a path planner known as Nafvn Planner, which is a Dijkstra algorithm [10]. The second is the SBPL Lattice planner [11], which introduces fine motion planning skills. This latter planner propagates a set of linear and nonlinear motion primitives, which have been customised for the INTROBOT.

The modules for obstacle avoidance, motion planning, and path planning are available to the system as separate ROS nodes.

### C. Localisation ROS Nodes

Depending on the situation, the navigation goal must be specified differently. For instance, in open outdoor areas, global positioning systems are usually adequate, thus allowing the user to specify a GPS position as final destination for the robot. Conversely, indoors and narrow outdoor areas hinder a clear sky view and, as a result, of a reliable GPS signal. As a consequence, localisation is typically defined with respect to a local map's frame of reference. In this case, localisation is often done by matching observations with the map itself.

In this work, GPS, IMU, and wheel odometry are fused with an Extended Kalman Filter (EKF) for localising the robot in open outdoor spaces. To build a local occupancy grid from laser readouts, used for obstacle avoidance purposes,

a second EKF is used. The second EKF only accounts for motion information, so that the global positioning error does not influence the registration of laser scans on the local map. The laser scanner is assumed to be fixed horizontally, thus sensing a 2D horizontal plane aligned with the robot's motion.

For indoor and outdoor narrow spaces, a Monte Carlo localisation process [12] is used. This process uses particle filters to track multiply localisation hypotheses, given horizontal laser scans, and a generated offline occupancy grid. The map was generated offline with a SLAM algorithm [13] while teleoperating the robot in the environment. To allow a smooth transition between the two localisation methods, a common metric frame of reference is set.

The modules for localisation in open spaces and localisation in narrow spaces are available to the system as separate ROS nodes.

### D. Context-Aware Graph of Nodes

This section describes how the previously described state-of-the-art components, abstracted as ROS nodes, are selected according to the context, thus enabling operation in heterogeneous environments. The selection is done by a *selector node* that is able to change the graph of nodes according to context-related messages provided by a *context extraction node*. Changing the graph of nodes means adjusting its connectivity, which, in turn, may require initiating and terminating nodes.

The context extraction node casts messages with the current context, given a set of a priori geo-referenced semantic map of the environment, and the current robot's position in world coordinates. Currently, the semantic map is generated offline by a human operator using a Web-based control centre based on Google Maps API <sup>2</sup>. This control centre provides the user with a set of tools for path specification, labelling regions in the environment, and register local occupancy grids (generated offline with SLAM) as overlays of the satellite imagery, i.e., making them geo-referenced. The same interface serves the purpose of remote monitoring the robot's operation and setting the robot's navigation goal, which is defined as a set of geo-referenced waypoints. To facilitate computations, the selector node transforms all GPS coordinates into Universal Transverse Mercator (UTM) coordinates, thus enabling the use of a Cartesian coordinate system. Asynchronous communications between robot and control centre are maintained via a shared folder in a file sharing server, which avoids the complexity of establishing and maintaining peer-to-peer communication channels.

As mentioned, the selector node changes the graph of nodes as the semantic label of the robot's current location changes. Concretely, in narrow spaces, map-based localisation and motion planning are used, whereas in open spaces GPS-based localisation and path planning are used. To avoid jitter in the transition caused by mis-localisation, a probabilistic test is performed prior to engaging in the actual transition. In addition to initiate and terminate nodes, the selector node needs also to perform some preparatory tasks. For example, entering a narrow space area triggers the loading of the corresponding

<sup>2</sup><https://developers.google.com/maps/>



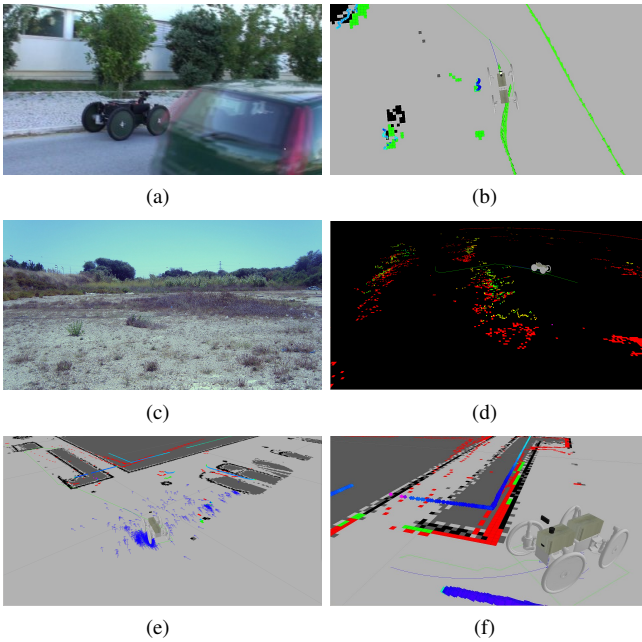


Fig. 4. Snapshots of the field trials. (a) Snapshot of car passing by the robot during the narrow-space field trial. (b) Robot’s internal map while avoiding a pedestrian on the narrow-space field trial (obstacles represented by green squares). (c) Site of the open-space field trial. (d) Robot’s internal map while on the open-space field trial filled, with several obstacle false positives due to tall vegetation (red and green squares). (e) Convergence of the localisation process during the transition phase between environments. Particles represented as blue arrows. (f) Robot’s internal map while reaching the final goal below the balcony (obstacles green squares).

offline generated local occupancy grid and setting the proper planner’s and obstacle avoidance’s parameters.

### III. EXPERIMENTAL RESULTS

This section presents the experimental setup and parameterisation used to assess the robustness and efficiency of the self-reconfigurable navigation system. The tests were carried out in the facilities of the New University of Lisbon organised in three different scenarios. Overall, the robot travelled 895 m at an average speed of  $0.5 \text{ ms}^{-1}$ .

The navigation stack was tested on a Intel(R) Core i7-2670QM CPU @ 2.20 GHz with 6 Gb of RAM, running a 64-bit Linux distribution Ubuntu 11.10 (Oneiric Ocelot). Low level control is handled by an on-board embedded PC.

#### A. Autonomous Navigation

This section presents the outdoor field trials used to evaluate the performance of the autonomous navigation module when presented with real world situations. A first test was carried out on a narrow space around a building. This test aimed at validating the ability of the navigation system to take the robot from a given location to another without recurring to GPS information, i.e., by localising itself based solely on offline generated maps. The ability of the system to rely on GPS was validated on a second test performed on an open environment. In this case the robot was asked to execute an offline specified route. Finally, the transition

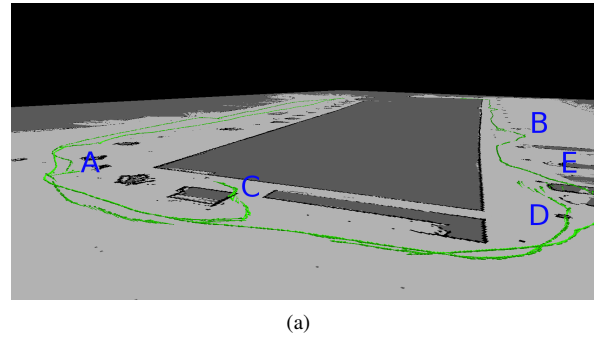


Fig. 5. Robot’s path (green plot) executed during the narrow-space field trial, overlaid on the offline generated map. Light grey, dark grey, and black represent free space, unknown space, and obstacles, respectively. Capital letters represent key situations in the course of the test. Situations A-B: avoidance of dynamic obstacles. Situations C-D: due to a slight mis-localisation of the robot, the path overlays the obstacles present in the off-line generated map. As the local map is updated the robot is still able to avoid the obstacles safely. Situation E: obstacles present in the off-line generated map but absent in execution time. This explains why the robot’s path apparently crosses these obstacles.

between both modes of navigation was analysed on a third test. The robot in this case was asked to move from a given location to another, where the former was located on an open space and the latter on a confined space. In this case the robot needed to be able to detect the context change and act accordingly in its navigation system. All tests were successful, thus showing that the environment-specific navigation methods perform appropriately, as well as the mechanism enabling the transition between them. A detailed description of these three tests is given next.

1) *Navigation in Narrow Environments:* In the narrow space navigation scenario, the robot needs to navigate on a sidewalk around a building present on a normal urban setting. The building is surrounded by trees, grass patches, and parking lots. In the frontal face of the building, a staircase and a ramp can be found. The back of the building faces dirt patches with light vegetation. Due to the presence of the tall and close-by walls, the navigation setup in this case is based on a Monte Carlo localisation process, i.e., based on a priori learned maps.

Throughout the run, several goals were set for the robot to pursue. Most of them were random, whereas other were picked in order to compel the robot to perform specific behaviours, such as circumnavigating the building.

From ten goals provided to the robot, only one was not successfully reached because the robot’s power source develop a glitch mid-run. When restarted an incorrect initial position estimate was made leading to slight mis-localisation that influenced the final pose in the next intermediate goal.

Fig. 5 depicts the path executed by the robot throughout the trial. Overall, the robot moved, on average, at  $0.5 \text{ ms}^{-1}$ , covering 396 m.

Let us now look into some of the key moments of the trial. Initially, the robot was placed below a balcony, where despite the closeness of the walls, it was able to move safely in the environment. While moving towards its goals, the robot had to often navigate around unexpected obstacles, such as people. To ensure safety, the obstacles present in the robot’s internal

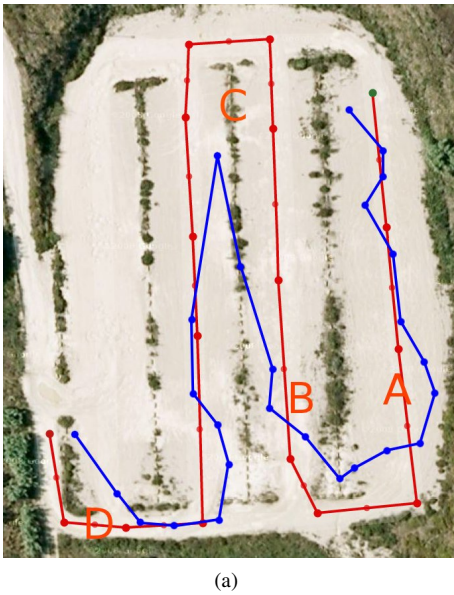


Fig. 6. Offline waypoint path (red plot) and actual robot's path (blue plot) executed during the open-space field trial, overlaid on the satellite aerial imagery. Capital letters represent key situations in the course of the test. Situation A: path deviation due to obstacle false positives generated by dense vegetation. Situation B: avoidance of a dynamic obstacle. Situation C: waypoints missed due to the presence of dense vegetation inducing obstacle false positives. Situation D: path deviation due to presence of dismantled steel fences in the desired path.

map of the environment were inflated. The outcome of this inflation were paths further from walls and other obstacles. In one situation, the inflation of the obstacles took the robot to the road. Note that the robot, in its current state, has no semantic information regarding roads and sidewalks and, so, it is unable to distinguish both in the path planning process. Although this is not a problem as the robot is still able to perceive any potential obstacle, it could be avoided with a smaller inflation. See for instance that it was able to detect a passing car (see Fig. 4(a)) and a pedestrian (see Fig. 4(b)). While the car did not cross the robot's desired path the pedestrian did triggering an avoidance behaviour.

2) *Navigation in Open Environments:* In the second field trial, the robot was asked to move in an open space. Concretely, the robot was requested to follow a path generated by the offline path design tool. The objective was to simulate a real world application of surveillance or recon in an off-road environment. The area chosen for this second trial is a rough dirt clearing with small elevations and with about 6300m<sup>2</sup>, filled with small concrete objects, large patches of ground and medium height vegetation and also some dismantled steel fences.

The absence of structure hampers the use of off-line generated maps. However, as a clear view of the sky is available, GPS can be used to provide global localisation information. Local maps are nevertheless created by the robot in order to help in the obstacle avoidance behaviour (see Fig 6).

The path followed by the robot despite successful, has some differences from the ideal route, caused by different factors. First, the satellite imagery is somewhat outdated,

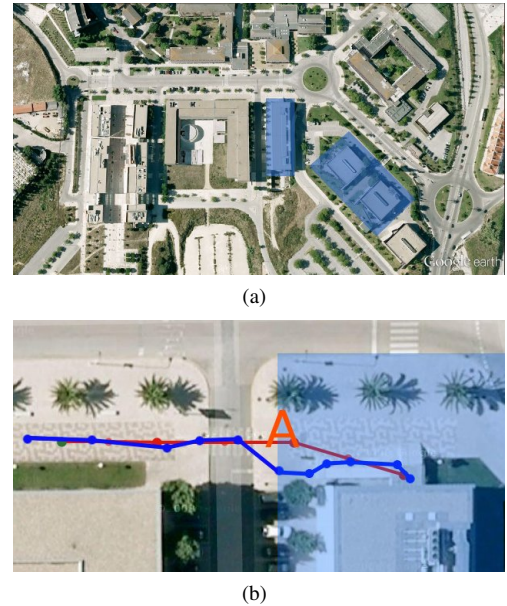


Fig. 7. Environments-transition field trial. (a) Semantic map used in the trial. (b) Offline waypoint path (red plot) and actual robot's path (blue plot) executed throughout the trial, overlaid on satellite imagery. Letter A represents the robot's transition from open space to narrow space. The blue overlay represents the area of the environment labelled as narrow space. The remainder of the map is labelled as open space.

which causes the operator to select a route that cannot be tracked by the robot. Dense vegetation growth and debris show up as unexpected obstacles that need to be avoided by the navigation stack (see Fig. 4(c) and Fig. 4(d)). To avoid that the presence of an obstacle induces the robot to be indefinitely pursuing an unreachable waypoint, this was taken as reached as soon as the robot was 10 m distant from it. In turn, this behaviour took the robot to skip certain goals, as it occurred at approximately half of the run, which also explains another detour from the expected path. These challenges show the ability of the navigation stack to produce creative solutions as exceptions emerge. That is, in challenging environments, robustness is a much more relevant feature than behavioural optimality.

In this trial, the robot travelled 221 m with an average speed of 0.7 ms<sup>-1</sup>. As for the previous trial, in this one, dynamic obstacles were also present in the run.

3) *Transition Between Narrow and Open Environments:* The previous sections described a set of experiments that allows us to show the ability of the system to navigate in narrow and in open environments. The navigation system was set differently for each case. Not only the localisation processes were different, but the navigation algorithms differed as well. This section presents experimental results showing that the robot is able to self-reconfigure its navigation system in order to adapt to context changes based on the off-line generated semantic map depicted in Fig-7(a). That is, depending on the configuration of space, i.e., narrow vs. open, the system automatically changes both localisation and navigation processes (see previous sections).

To validate the model, the robot was initiated on a open-

labelled area and asked to follow a set of waypoints, being the last ones in a narrow-labelled area. In fact, the final goal in this trial is the same as the initial position of the robot in the narrow-space trial, i.e., underneath the building's balcony. As expected, the robot moved along the open area using GPS information, crossed the road, and entered the narrow-labelled region (see Fig 7(b)). At that point, the robot recognises the context change and triggers a map-based navigation process, i.e., discards the GPS information for its localisation. Using this localisation method, the robot is able to navigate near the building and reach its goal underneath the building's balcony.

As aforementioned, after crossing the road the robot entered in the narrow space area. At that point, the robot started the transition phase, which includes starting the map-based localisation method and changing some of the navigation parameters. One of the most important stages of the transition phase is the first estimate of the robot's position in the map. This is seamlessly handled by the Monte Carlo localisation process, which tracks multiple localisation hypotheses while sensory evidence on the most likely location is progressively integrated. In the method employed, each localisation hypothesis is represented by a particle and the full distribution represented by a set of particles. Fig. 4(e,f) illustrates these particles converging on a solution as sensory feedback is generated. The outcome is an accurate localisation of the robot in the offline generated map, which, in turn, allows the robot to accurately navigate in the confined environment where the goal is located.

As in the previous experiments, the robot encountered dynamic obstacles along its course. As expected the robot slowed down in the presence of these obstacles in order to maintain safety.

#### IV. CONCLUSIONS

A self-reconfigurable navigation system designed for autonomous robots operating in heterogeneous environments was presented. Self-reconfigurability is attained by relying in the use of contextual information and in the use of a well-defined control system architecture. Contextual information is obtained from offline generated semantic maps. Currently, these maps are prepared by a human operator, but in the future these are expected to be automatically generated and made available as internet services. The proper abstraction level required from the control system architecture is obtained from an extensive use of the well-known Robotics Operating System (ROS).

The developed system has been deployed in an industrial-grade all-terrain robot, the INTROBOT. Experiments showed the benefits of using a context-aware self-reconfigurable navigation system when dealing with heterogeneous environments. This is particularly important to ensure smooth transitions between different environments. Hence, the proposed system contributes to bring robots from labs to real world applications.

The successful use of the system provides additional evidence on the benefits of considering ROS as a backbone of the robot's control system. The well-defined specification provided by ROS enables the development of vertical supervisory systems, as the one herein proposed. Furthermore, with the advance of ROS-like architectures, the feasibility of general purpose learning-based supervisory systems become a reality.

Besides developing the system towards a general purpose context-aware supervisory node, we also expect to improve the INTROBOT's navigation stack in several ways. For instance, we expect to include visual odometry [14] for improved motion estimation in off-road environments. Based on [15], we expect to include GPS-IMU information, when available, to bias the map-based localisation process.

#### ACKNOWLEDGMENT

This work was partially supported by the QREN-funded project ROBOT-QREN and by the CTS multi-annual funding, through the PIDDAC Program funds.

#### REFERENCES

- [1] K. Konolige, M. Agrawal, M. R. Blas, R. C. Bolles, B. Gerkey, J. Solà, and A. Sundaresan, "Mapping, navigation, and learning for off-road traversal," *Journal of Field Robotics*, vol. 26, no. 1, pp. 88–113, 2009.
- [2] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [3] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Etinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhne *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.
- [4] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proceedings of the 2009 ICRA Open-Source Software Workshop*, 2009.
- [5] M. Guedes, P. Santana, P. Deusdado, R. Mendonça, F. Marques, N. Henriques, A. Lourenço, L. Correia, J. Barata, and L. Flores, "Ares-iii: A versatile multi-purpose all-terrain robot," in *Proceedings of 2012 the International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2012.
- [6] P. Santana, C. Santos, D. Chaínho, L. Correia, and J. Barata, "Predicting affordances from gist," *From Animals to Animats 11*, pp. 325–334, 2010.
- [7] J. Collier and A. Ramirez-Serrano, "Environment classification for indoor/outdoor robotic mapping," in *Proceedings of the 2009 Canadian Conference on Computer and Robot Vision (CRV)*. IEEE, 2009, pp. 276–283.
- [8] D. Silver, B. Sofman, N. Vandapel, J. Bagnell, and A. Stentz, "Experimental analysis of overhead data processing to support long range navigation," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2006, pp. 2443–2450.
- [9] B. Gerkey and K. Konolige, "Planning and control in unstructured terrain," in *Proceedings of 2008 Workshop on Path Planning on Costmaps (ICRA)*, 2008.
- [10] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
- [11] M. Likhachev and D. Ferguson, "Planning long dynamically feasible maneuvers for autonomous vehicles," *The International Journal of Robotics Research*, vol. 28, no. 8, pp. 933–945, 2009.
- [12] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy *et al.*, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *The International Journal of Robotics Research*, vol. 19, no. 11, pp. 972–999, 2000.
- [13] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.
- [14] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2004, pp. 1–652.
- [15] I. Miller, M. Campbell, and D. Huttenlocher, "Map-aided localization in sparse global positioning system environments using vision and particle filtering," *Journal of Field Robotics*, vol. 28, no. 5, pp. 619–643, 2011.