3D modelling and simulation of a crawler robot in ROS/Gazebo

Article · December 2016
DOI: 10.1145/3029610.3029641

Maxim Sokolov
Innopolis University

Aidar Gabdullin
Innopolis University

Ilya Afanasyev
Innopolis University

Roman Lavrenov
Kazan (Volga Region) Federal University

3D modelling and simulation of a crawler robot in ROS/Gazebo

Maxim Sokolov[1,a], Roman Lavrenov[2], Aidar Gabdullin[1], Ilya Afanasyev[1] and Evgeni Magid[2]

[1]Robotics Institute, Innopolis University, 1, Universitetskaya str., Innopolis, 420500, Russia
[2]Intelligent Robotics Department, Kazan Federal University, 35, Kremlyovskaya str., Kazan, 420008, Russia

Abstract. Modelling and animation of crawler UGV's caterpillars is a complicated task, which has not been completely resolved in ROS/Gazebo simulators. In this paper, we proposed an approximation of track-terrain interaction of a crawler UGV, perform modelling and simulation of Russian crawler robot "Engineer" within ROS/Gazebo and visualize its motion in ROS/RViz software. Finally, we test the proposed model in heterogeneous robot group navigation scenario within uncertain Gazebo environment.

1 Introduction

A long-standing goal of robotics, among others, is allowing robots to work in unreachable or dangerous environments, and one of the most demanded is search-and-rescue or simply rescue robotics [1]. A broad range of rescue robotics applications includes exploration of volcano craters and mine tunnels, operations in nuclear and chemically polluted environments, human teams replacement and support in scouting, mine clearing, antiterrorist operations etc. Rescue robots extend capabilities of human rescuers while increasing their safety, and execute tasks, which are beyond our natural abilities. One of particular rescue robotics applications is urban search and rescue (USAR) domain, which is probably one of the most dangerous environments with victims often buried in unreachable locations [2,3]. Rescue robots were intensively tested for the first time in real world environment during September 11, 2001 disaster at the World Trade Center (WTC). In debris environment of the collapsed towers, radio controlled and tethered robots of different sizes were deployed eight times by CRASAR [4]. The catastrophe at Fukushima Dai Ichi nuclear plant emphasized importance of USAR research, pointing out specific case of high radioactive contamination, which prevents human personal from executing any tasks within such environment and introduces high requirements for mobile robot's equipment, communication and sensory capabilities.

Currently, one of actual problems of rescue robotics deals with interaction of unmanned ground robots (UGV) and unmanned aerial vehicles (UAVs) in USAR scenarios. A heterogeneous group of robots should work as a team in order to improve information acquisition about environment and solve tasks of transportation logistics, reconnaissance, and search-and-rescue. This could significantly improve team's performance due to different sensory system features of robots and different operational space [5]. Our research investigates collaboration aspects between a heterogonous group of a UGV and a number of small-size UAVs, focusing on operation in uncertain environments within USAR scenarios. The environment could be completely unknown or may be represented by a partially outdated and/or imprecise map. The robots need to perceive surrounding environment collaboratively, infer their own states, and plan high-quality actions under various uncertainties. Our main objective is to develop collaboration strategies between robots in order to operate autonomously with partial and uncertain information regarding environment. We build upon recent progress in simultaneous localization and mapping (SLAM) and belief space planning, and explore multi-robot belief space planning approaches [6,7]. The proposed heterogeneous robotic team should collaboratively resolve point-to-point path planning task, which finds a path from some starting configuration S to target configuration T [8]. An objective of a path planner is to provide a continuous path from S to T, which avoids all obstacles and minimizes some positive cost metric. Classical path planning methods usually involve a global or a local path planner. A global path planner uses a precise map of environment, which is available to the planner a priori, and path planning process generates a hazard-free path in the off-line mode; thus, its main concern is to provide a computationally effective scheme for a particular cost metric [9]. On the opposite, a local path planner is a purely reactive on-line planner that collects sensory information and acts based on its analysis [10,11]. Each of these two classical approaches has its own strong and weak features; while in the past robotics researchers were mainly developing them separately, a more effective approach would mix them together to strengthen their advantages and compensate pitfalls.

---

[a] Corresponding author: m.sokolov@innopolis.ru

We develop navigation algorithm prototypes for heterogeneous robotic team navigation, which combine together global and local approaches, within MATLAB environment. Next, physical properties of the robots are modelled in ROS/Gazebo environment in order to verify the algorithms' prototypes in a realistic simulation. Such verification within the simulator gives an opportunity to perform intensive testing of the algorithms and saves significant amount of work, which is required for experimental part of the research, increases awareness and safety of further experiments [12,13].

This paper presents in physical modelling and simulation of Russian robot "Engineer" within ROS environment. The modelling takes into account such properties as robot dimensions, mass, centre of gravity, moments of inertia and motion speed. Furthermore, in order to gather information from (simulated) environment we use the already built-in models of sensors, which have one-on-one correspondence with our hardware. Navigation algorithms are implemented in C++ language and are incorporated into the model. Moreover, due to realistic properties of ROS, most of the simulated C++ code further will be directly transferred from the simulation onto the real robot for an experimental stage.

The rest of the paper is organized as follows. Section 2 introduces system setup and simulation environment. Section 3 describes "Engineer" robot modelling in ROS/Gazebo. Section 4 presents results of robot navigation and tests in 3D Gazebo environment. Finally, we present conclusions and describe our future work in the last section.

2 System setup

2.1 Crawler-type ground robot "Engineer"

Mobile ground robot "Engineer" (Fig. 1) is a crawler-type robot, which is designed and produced by Russian company "Servosila"[1]. This robot type was specially designed for search and rescue operations within natural and anthropogenic disasters or special missions (e.g., firefighting assistance, tunnels/pipelines inspection and maintenance, etc.). The robot is waterproof, dust-proof and is capable of working under rain and snow conditions. It has special chassis, which are optimized to overcome various obstacles that can appear in urban environment after a disaster. The robot has a metal body with radiation-hardened electronics and a sensors pack. The total ready-for-mission robot weight without sensors[2] is about 16.3 kg (Table 1 presents the weight parameters of the main robot equipment). The robot is able to climb stairs, traverse doorways and narrow passages, penetrate into buildings and raise its head, where most of the sensors are placed, thus enabling an operator to perform visual inspection by looking inside windows of parked vehicles, ground floors, etc. The robot is equipped with a robotic arm that can grasp, push and pull potentially dangerous objects or open different doors employing grippers in a remotely controlled mode. The robot is equipped with a bright headlight for operations in low illumination conditions. "Engineer's" variety of sensors includes GPS receiver, 3D and/or 2D laser scanner, stereo vision system with two head cameras, a rear-view camera, a front-view camera with optical zoom, a block of inertial sensors. Due to large number of degrees of freedom, rich sensors' set and a powerful on-board computer (Table 2), "Engineer" robot become a popular research and education platform for laboratories and universities.

Figure 1. Servosila "Engineer" crawler-type mobile robot. Courtesy of Servosila company.

Table 1. Weight parameters of "Engineer" robot.

Equipment	Weight parameter
Robot chassis with two main reversible tracks, two traction motors and motor control electronics	8.8 kg
On-board control and power system	2.1 kg
Sealed connector for external payloads or external computer	0.1 kg
LiFePo battery	3.7 kg
Power supply for standard robot battery (with cable)	1.6 kg

Table 2. "Engineer" robot's on-board computing system configuration.

System	Parameter
Processor	Intel i5, 4th Generation
RAM	4 GB
SSD	32 GB
OpenCL technology	Supported
Radio channel	WiFi

2.2 Modelling and simulation environment: Gazebo integrated with ROS

Robot simulators allow to design, simulate and test robotic applications in relevant physical environment independently on availability of real hardware, saving development time and cost. We performed "Engineer"

---

[1] Servosila company designs and produces mobile robots and equipement: www.servosila.com/en/

[2] Sensors pack may include various sensors, which could be installed by the manufacturer on user request

robot modelling and simulation in Gazebo[3] 2.2.3 robot simulator, which is integrated with ROS Indigo[4]. We utilize ROS as a convenient open-source framework for a mobile robot development, because it provides such operating system services as low-level device control, hardware abstraction, message passing between processes, and package management. Multiple interconnected processes running in ROS are represented as a graph, which main element is (obviously) a node. Nodes can receive and transmit data from sensors, control actuators, states, and handle different messages.

ROS-based software can be divided into three groups:

1. Languages and platform-independent tools used to create and distribute software.
2. Client libraries such as *roscpp*, *rospy* and *roslisp*.
3. Client libraries' packages for a variety of sensors, robots and particular algorithms.

Gazebo uses Open Dynamics Engine as a physics engine and Object-Oriented Graphics Rendering Engine for 3D rendering, has a free license for research, provides user-friendly graphical user interface and allows describing robot dynamics with Python language.

3 Robot modelling in ROS/Gazebo

3.1. Modelling main body of the robot

We completed a first draft model prototype of "Engineer" main body, which is a crawler mobile base with two additional front flippers. Figure 2 presents a schematic drawing of "Engineer" crawler robot.

Modelling chain track belt-drive mechanism is a difficult task as far as it contains multiple details, which we should build into a model and animate. In order to create a simplified model without modelling the physics of track-surface contact, typical slipping and skidding of tracks, and simulate permanent contact of model's crawlers with locomotion surface we added eight *imaginary* pseudo wheels, four on each side. These wheels are placed in the robot tracks vicinity: one pair corresponds to the flippers front, one pair corresponds to the joints that link flippers to main crawlers, and the rest two pairs are responsible for the robot's main body (Fig.3, on the right). The imaginary wheels are linked to the main robot body with ROS joints, thus connecting two parts of a robot model and allowing rotation of one part relative to another. Moreover, such built-in rotation control interface for both angle and speed already exists (*joint_position_controller, joint_velocity_controller*). This solution is not unique, and a number of research laboratories, including Fraunhofer Institute for Intelligent Analysis and Information Systems IAIS[5] and Technical University of Darmstadt[6] [14], have previously used pseudo wheels to simulate caterpillar motion. Increasing the number of imaginary wheels together with their size decreasing, respectively increases the correspondence of such model to a real caterpillar physics, including surface contacts, friction calculation etc. Obtaining a simple wheel control, we synchronize the wheels' speed with ROS request and response messages; in particular, a function reads a value of a driving wheel speed and transmits this value to other wheels.

Next, we simulated front flippers motion, which help navigating the robot through the rough terrain (e.g., stairs, slopes, and debris). Using ROS joints to attach the flippers to the main body we solved the issue of having different sizes of flipper wheels and their matching with caterpillars. The limitation on flipper rotation angles was set by standard ROS modelling tools. The wheels' synchronization was performed through required rotation speed calculation from the wheels' radii ratio. This allows visualizing a flipper rotation in RViz software[7] (Fig.3, on the right). Finally, the main body model is capable of performing three basic motions: raising and lowering flippers and locomotion through the right and left four-wheel sets rotation.

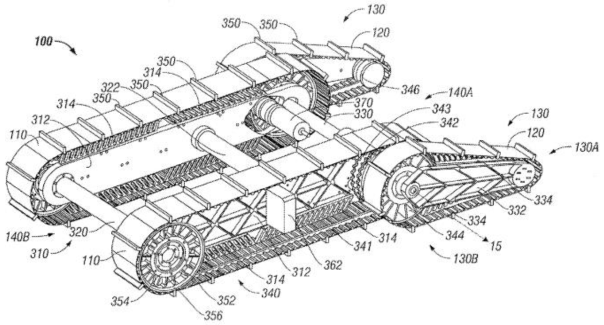

Figure 2. The main body schematic drawing of "Engineer" crawler robot. Courtesy of Servosila company.

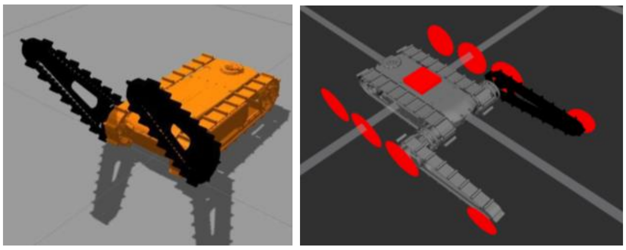

Figure 3. Simulation of «Engineer» main body in Gazebo (left) and in RViz (right). Imaginary wheels are marked in red.

3.2 Modelling upper part of the robot

The robot upper part consists of a 3DoF manipulator with a gripper and a head over it, which were modelled using CAD models (Fig. 4). All parts of the model are

---

[3] Gazebo robot simulator, www.gazebosim.org

[4] Robot Operating System (ROS) - a set of software libraries and tools for robots' development, www.ros.org

[5] Elmasry M, 2012, A tracked robot climbing stairs, www.youtube.com/watch?v=dhgcoXviDjo

[6] Team "Hector", 2013, Hector tracked vehicles common, ROS packages are available at https://github.com/tu-darmstadt-ros-pkg/hector_tracked_vehicles_common

[7] RViz - 3D visualization tool for ROS, wiki.ros.org/rviz

connected with ROS joints and controlled by angles, which are limited according to the robot specifications.

While modelling the robot's upper part, we faced a problem with manipulator control, which was caused by its weight distribution: the gripper and the head weight 2 kg together, while the total weight of robot's upper part is 4 kg. Thus, when the robot stretches its head to maximum height involving all manipulator joints rotations, a large cumulative torque appears due to the large mass of the head and the long lever of the manipulator. However, if only the second (middle) joint rotates, the torque reduces significantly. To optimize the robot manipulator control with a PID controller in our future work we consider calculating optimal PID coefficients for damping down motion-induced oscillations.

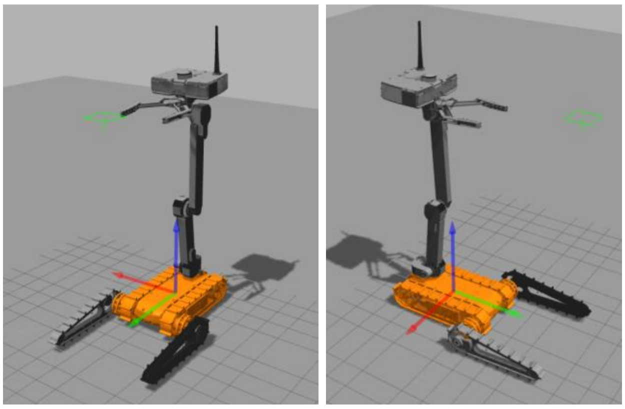

Figure 4. The complete robot model in Gazebo.

4 Navigation in Gazebo environment

4.1 Navigation task

For a practical model application, a capability of integration with ROS navigation stack is critical. To test the created ROS/Gazebo-based model of "Engineer" we used a standalone point-to-point navigation algorithm, run ROS driver and applied robot position commands. We integrated the designed model into a simulated system presented in [15], and tested a quality of crawler simulation in Gazebo simulator. This system consists of a simulated environment with obstacles, a UGV and two generic multirotor-type unmanned aerial vehicles (UAVs, Fig. 5). Within this system, two UAVs collaboratively investigate environment and perform its mapping. Next, the obtained map is applied by the UGV for navigation with an imbedded ROS navigation algorithm. ROS packages *hector_quadrotor* and *husky_navigation* are used for control and navigation of UAVs and Clearpath Husky UGV respectively.

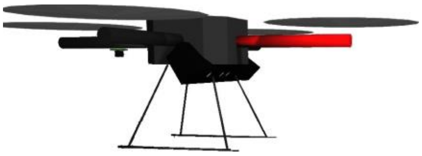

Figure 5. ROS/Gazebo generic UAV simulation.

We apply NavFn[8] and DWA_planner[9] packages as a global and a local path planner respectively. We substitute Voronoi diagram planner of the original system in [15] with NavFn, which provides a fast-interpolated path planning with Dijkstra algorithm. DWA_planner exploits Dynamic Window Approach [16], which takes into account robot dynamics and thus performs safe maneuvers without obstacles' collision.

4.2 Simulation and testing

We have integrated our "Engineer" robot model into the simulated system of [15] instead of original model of a wheeled UGV "Husky". The robot was tested in a flat surface simulated environment (Fig.6-7) and, since the front flippers were not used for navigation, we permanently fixed them at zero degree position contacting ground surface. We used only main body of the robot in simulated environment, because the robot's upper part does not affect to the navigation. To bypass obstacles in local navigation mode the robot uses a LIDAR sensor. Results of robot navigation is presented in Fig.6 and 7. One of our future challenges is to verify its performance in a 3D map with obstacles.

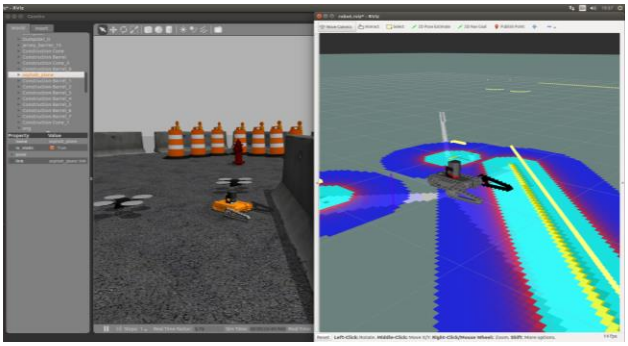

Figure 6. "Engineer" robot and two multi-rotor UAVs simulation in Gazebo (left) and RViz (right).

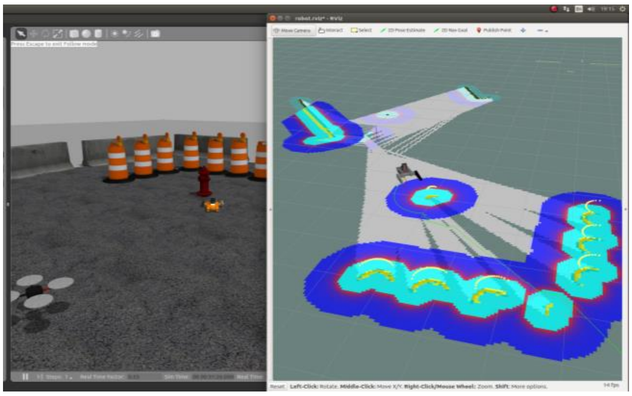

Figure 7. Point-to-point navigation Gazebo (left) and RViz (right). Green line is a global path; red line is a local path.

---

[8] ROS NavFn package, http://wiki.ros.org/navfn

[9] ROS DWA planner, http://wiki.ros.org/dwa_local_planner

5 Conclusions and Future work

Modelling and animation of crawler UGV's caterpillars is a complicated task, which has not been completely resolved in ROS/Gazebo simulators. In this paper, we proposed an approximation of track-terrain interaction of a crawler UGV with a set of imaginary pseudo wheels for each caterpillar with speed synchronization via ROS messages. We created a first version of "Engineer" crawler robot model in ROS/Gazebo environment, which corresponds to the real robot. We used CAD models provided by the manufacturing "Servosila" company, assembled them into a workable 3D simulation, and visualized robot motion in ROS/RViz software. The proposed model successfully repeats real "Engineer" robot physics during its motion and supports crawler locomotion and upper part (manipulator) control. The model was integrated with a ready-to-use ROS navigation stack and tested within ROS/Gazebo and ROS/RViz environments in a group navigation scenario together with two UAVs.

In the next stages of our research, we will extend the proposed approaches by applying adaptive PID controller with optimal coefficients to damp motion-induced oscillations and to control optimal motion of heavy upper part of the robot. We also plan to extend path planning to uneven 3D terrains where front flippers will be used to overcome traversable obstacles. Our "Engineer" robot model and original software files are available for a download in the public domain of GitHub[10].

Acknowledgements

This work was partially supported by the Russian Foundation for Basic Research (RFBR) and Ministry of Science Technology & Space State of Israel (joint project ID 15-57-06010). Part of the work was performed according to the Russian Government Program of Competitive Growth of Kazan Federal University. Special thanks to "Servosila" company (Russia) for providing CAD models of "Engineer" robot.

References

1. R. R. Murphy, Human-robot interaction in rescue robotics, *IEEE Trans. on Systems, Man, and Cybernetics, Part C*, **34**, 2 (2004)
2. A. Jacoff, E. Messina, and J. Evans, A standard test course for urban search and rescue robots, *NIST special publication SP* (2001)
3. T. Kimura, W. C. Vie, and Y. Ukai, Development of a usar robot considering camera view angle and grouser shape of crawler. *IEEE Int. Conf. on Robotics and Biomimetics* (2009)
4. M. R. Blackburn, H. R. Everett, and R. T. Laird, After action report to the joint program office: Center for the robotic assisted search and rescue (CRASAR) related efforts at the world trade center (No. SSC/SD-TD-3141). Space And Naval Warfare Systems Center San Diego CA (2002)
5. T. Kamegawa, N. Sato, M. Hatayama, Y. Uo, and F. Matsuno, System Integration for Grouped Rescue Robots System Using Robohoc Network. *Workshop on Robots and Sensors integration in future rescue INformation system ROSIN'10* (2010)
6. V. Indelman, L. Carlone, and F. Dellaert, Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The Int. J. of Robotics Research*, **34**, 7 (2015)
7. T. Regev, and V. Indelman, Multi-Robot Decentralized Belief Space Planning in Unknown Environments via Efficient Re-Evaluation of Impacted Paths. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2016)
8. H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki and S. Thrun, Principles of Robot Motion: Theory, Algorithms, and Implementations. MIT Press (2005)
9. E. Magid, D. Keren, E. Rivlin and I. Yavneh, Spline-Based Robot Navigation. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2006)
10. E. Magid, T. Tsubouchi, E. Koyanagi, and T. Yoshida, Building a search tree for a pilot system of a rescue search robot in a discretized random step environment. *J. of Robotics and Mechatronics*, **23**, 4 (2011)
11. E. Magid, T. Tsubouchi, E. Koyanagi, and T. Yoshida, Static balance for rescue robot navigation: Losing balance on purpose within random step environment. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems* (2010)
12. I. Shimchik, A. Sagitov, I. Afanasyev, F. Matsuno and E. Magid, Golf cart prototype development and navigation simulation using ROS and Gazebo. *Int. Conf. on Mechanical, System and Control Engineering, MATEC Web Conf.*, **75**, 09005 (2016)
13. I.M. Afanasyev, A.G. Sagitov, E.A. Magid, ROS-based SLAM for a Gazebo-simulated mobile robot in image-based 3D model of indoor environment. *LNCS*, **9386** (2016)
14. S. Kohlbrecher, F. Kunz, D. Koert, C. Rose, P. Manns, K. Daun, O. von Stryk, Towards Highly Reliable Autonomy for Urban Search and Rescue Robots. Robot Soccer World Cup. Springer International Publishing (2014)
15. A. Gabdullin, A. Buyval, R. Lavrenov, E. Magid, ROS-modelling of interaction between UAV and ground robots for solving path planning problem in static environment, *3rd Workshop on Unmanned Vehicles with Artificial Intelligence Elements*, in Russian (2016)
16. D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, *IEEE Robotics and Automation Magazine*, **4**, 1 (1997)

---

[10] https://github.com/Somal/Robot_Engineer/