3D Simulation to Validate Autonomous Intervention Systems Architecture for Disaster Management

Tullio Tanzi(✉) and Matteo Bertolino

LTCI, Télécom Paris, Institut Polytechnique de Paris, Paris, France
tullio.tanzi@telecom-paris.fr

Abstract. The use of autonomous robots either on the ground (i.e., Rover) or flying (i.e., Drone) constitutes a major progress in the support of a crisis. To work properly and to reach the desired level of autonomy, they have to be correctly configured though. Indeed, errors on robot configuration can lead to imprecise or erroneous data and, consequently, erroneous decisions can result from them. Before the beginning of the mission, it is important also to achieve a strong level of confidence about the usage of the sensors (for example, LIDARs) with respect to the context of the mission. Many aspects of these validations cannot be performed during the mission, for example verifying the behaviour of a rover following a strong collision with an external actor (such as debris) that can potentially damage or break some components. Moreover, during a real mission it is not always possible making huge modifications in the system configuration. In this respect, simulating the behaviour of the system in a virtual environment, similar to the real physical world, can constitute a good validation approach before the mission. These simulations allow to validate the behaviour and the configuration of the system as well as the most appropriate equipment of it.

Keywords: Autonomous system · Simulation · Models · Disaster

1 Introduction

The usage of autonomous systems for rescue operations constitutes a great improvement in the conditions of intervention on a site after a disaster-type event. However, in order to be truly effective, these systems shall be perfectly in-line with the objectives of the mission as well as the conditions in which they have to evolve. This means that they must be configured for each mission assigned to them, customizing them from a physical point of view (hardware) and in terms of embedded intelligence (software).

This raises a series of important questions: how to check the adequacy between the architecture of the system (in a broad sense) and the mission? How to do this check before operating on site?

Our objective is to avoid discovering an inadequacy between the autonomous system and the mission during the mission itself. Our work therefore focuses on this verification in order to adapt this configuration before the on-site departure of the equipment.

Published by Springer Nature Switzerland AG 2020
Y. Murayama et al. (Eds.): ITDRR 2019, IFIP AICT 575, pp. 196–211, 2020.
https://doi.org/10.1007/978-3-030-48939-7_17

1.1 Disaster Management

With the growing improving and reliability of technologies, we can rely on information technology (IT) to elaborate and refine a disaster recovery management strategy. A disaster recovery management strategy is intended to be the set of plans, processes and techniques to be implemented, with the final goal to saving lives as well as finding survivors or let the life restarts back to normal in the shortest possible time [1]. After a disaster, the impacted area could have suffered huge mutations due to terrain's alterations or presence of debris, for example. Some information can be obtained only after an in-place reconnaissance and not easily. The communication with the local inhabitants may be hindered by physical constraints or by the fact that people abandoned the area after the catastrophe. Even a local inspection presents strong limitations. For example, the poor visibility range can result in erroneous decisions based on incomplete or erroneous data. However, completeness and correctness information are necessary for the decision-making process that precedes the emergency intervention [2].

1.2 Benefits from Robot Usage

In this respect, the usage of Autonomous Systems (A.S.) such as rovers or drones can assist the recovering operations for many tasks. For example, A.S. can be used for Rapid-Mapping or to scan the affected area for finding survivors, among others. Figure 1 contains the 3D CAD view of our ArcTurius Rover [2], an A.S. for post-catastrophe humanitarian mission. It has been designed and developed by LabSoc, a research group on complex digital electronic systems from LTCI laboratory of Télécom Paris.

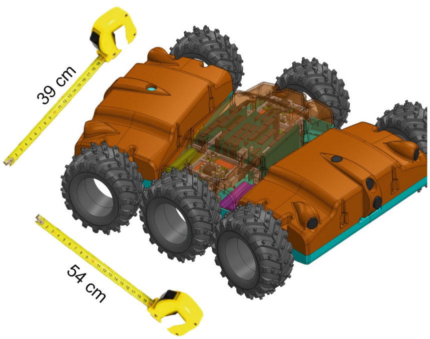

Fig. 1. ArcTurius Rover: autonomous system for post-catastrophe humanitarian mission (3D CAD view).

ArcTurius Rover has to work for several days in total autonomy, underground (subsoil and basement given by building ruins), searching for the presence of survivors.

Its design implies many challenges. Specifically, the environment of the mission prevents the rover by communicating with the operational centre by using radionavigation means. Moreover, the length of the mission in terms of time introduces the problem of energy consumption. Thus, the design and the configuration have to be strongly considered in order to achieve the desired behaviour while considering the power management. In the context of this article, ArcTurius is our reference in the study of the rover behaviour within a virtual environment. Towards 3D-based simulations, our goal is to validate the design and the configuration (specific to a particular mission) of ArcTurius in order to enhance the effectiveness of A.S. usage after a catastrophe. Our laboratory works in the design of techniques that facilitate the recovery process in an environmentally critical context. In this regard, we propose a new approach, based on 3D simulations of the real world that speeds-up the definition of a recovery management strategy. Our research work aims to get a role immediately after a disaster happens, while the rescue teams are approaching the place of the crisis. It targets two main objectives:

1. Finding the best design and the best configuration of the A.S. for the target mission. To work properly and to reach the desired level of autonomy, A.S. have to be correctly designed and configured. Indeed, errors on robot configuration can lead to imprecise or erroneous data and, consequently, erroneous decisions can result from them. Before the beginning of the mission, it is important to achieve a strong level of confidence about the usage of the sensors [2–4] (for example, LIDARs) with respect to the surrounding world. For instance, we can evaluate the positioning of a LIDAR in order to minimise the impact of external noise, or whether the terrain's discontinuity perturbs data acquisition [5].
2. Acquiring a better knowledge of an area that the rescue teams do not know or do not know more because of environmental alterations. While the recovery teams are reaching the crisis area, A.S. can autonomously operate performing the above mentioned tasks. Their result will help the definition of a strategy, even complex, or they can help modifying run-time an existing plan [2, 6].

The configuration of an A.S. depends by some key points that include: i) the configuration of the devastated area and ii) the main objectives of the Search and Rescue (SaR) mission such as damage assessment, people search and location, etc. The choice of the right equipment such as the set of sensors to be mounted as well as the best physical placement is a non-trivial task. A bad choice in this step may make the A. S. not reliable as expected.

2 Simulation

We propose a 3D-simulation, which the direction of the rescue team can perform before their arrival. Briefly, an initial map of the real world is taken, for example through satellites' data, then it is injected onto a graphical engine. We define through a model-driven engineering approach the physics of the terrain and the physics of the actors that populate the world. Environmental conditions are taken into account too. We need to model the design and the geometry of the A.S. under examination and to provide a description of its behaviour. In this respect, the modelling of sensors and actuators, part of the A.S. and that interact with the external world, plays a main role. Through a physical engine, we are allowed to rapidly testing algorithms, designing robots and simulating their behaviour in realistic scenarios. The 3D vision enhances and speeds-up the comprehension of the designers.

Figure 2 shows an overview of the proposed simulation system. It is composed by several computational blocks that communicate through the exchange of high-level messages. The main components are:

1. Graphical engine that includes the rendering of textures, lights, shadows, etc.
2. Physical engine that allows realistic physics simulation and that is able to interact with the graphical engine with computer animation API
3. A comprehensive description (architecture, physics and behaviour) of the A.S. under examination. Sensors and actuators are included in this component of the system
4. An engine able to generate realistic terrain data
5. An engine that created realistic environment actors and conditions close to the real conditions due to a catastrophe.

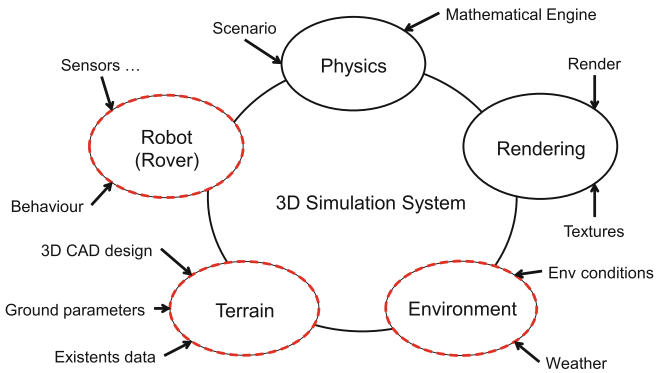

Fig. 2. 3D simulation system general architecture

The system in Fig. 2 is based on an existing system named Gazebo Simulator [7]. Gazebo is a well know 3D robot simulator that allows to rapidly testing robots in a physically realistic scenario. The input of the system is a precise modelling of the A.S. as well as the surrounding environment where the physical properties of each element are described. Starting from these models, Gazebo is able to perform reliable physical simulations and to represent them in a realistic rendering. Unfortunately, with respect to disaster management, the only Gazebo supports well only the blocks responsible for the physics and the rendering (blocks characterized by a black continuous line in Fig. 2), whereas features miss to obtain fast and realistic simulations in a disaster environment. Moreover, we need a way to establish the communication between all the components of the system. To do that, we built our system on Robot Operating System (ROS), deepened in Sect. 3.

2.1 Robot Modelling

In order to correctly reproduce the behaviour and the interaction of an A.S. in a surrounding world, we have to provide to the simulation system the following inputs: i) architecture, ii) physical properties, iii) behaviour and iv) sensors and actuators.

The architecture of an A.S. is intended to be the shape and the geometry of each component that is part of the A.S and the connections between them. In this respect, there are two possibilities that are characterized by different level of granularity. The first approach is object-oriented and it models the geometry of A.S. components through simplified shapes (box, cylinder, etc.) to whom are associated static and dynamic characteristics. Alternately, the A.S. architecture can be expressed in a more refined way through 3D designs. In this respect, the CAD models in Fig. 1 can be a possibility. In both cases, the input of the system is a XML-like file in which the description and the origin of each component is made explicit. Providing the architecture of each mechanical component is not enough though. Indeed, physical properties of them have to be defined as well as the characteristics of their interconnections. With physical properties we intend, for example, the characteristics features of rigid bodies: mass, inertia, the respect of kinematics laws, any kind of friction, coefficients that describe the reaction to an impact, etc. More physics parameters are provided, more the reliability of the simulation is enhanced.

Modelling the junctions between components enhance the reliability of the model too, for a dual reason. First of all, the type of interconnection permits to describe the physical movement of the system. In this regard, there is a difference between a fixed connection with no degrees of freedom and a hinge joint that rotates along the axis and has a limited range specified by the upper and lower limits. For example, the latter can be used to describe the movement of a wheel with respect to the chassis to which it is attached. Secondly, junctions description permits to play a key-role in the physical integrity of the system, because they permit to perform more realistic safety analysis. Supposing to run the A.S. in a extreme environment that, because to its intrinsic danger, can lead to partial damages to the rover architecture. In such conditions, we have to be sure to make the correct design choices in order to enhance the A.S. survivability. In this respect, parameters such as the maximum effort and the maximum velocity that a junction can endure are important. Indeed, if an A.S. is subject to an huge strength in correspondence to a contact point, we have to be sure to avoid a fragmentation of A.S. components.

According to the object-oriented paradigm, we modelled ArcTurius rover with a chain of 3 boxes, interconnected by a custom junction that allows flexibility to each body to the chain. Wheels are cylinders, connected to each body through a revolute joint. Part of the future work include the ArcTurius representation starting to 3D-CAD model, keeping unchanged the physical properties associated to each component and junction.

A possible goal of the system is checking and analysing the behaviour of the A.S. during its interaction with the surrounding world. Because of that, an important part of A.S. representation is its behaviour. Through ROS, the system accepts as an input a program that has the role to move the A.S. during the simulation. In the context of rover ArcTurius, the simplest version of such program moves the virtual rover within a virtual environment applying a force to the junctions that connect the wheels with the chassis of the rover. The movement decisions can be based on the data acquired during the simulation. Even though sensors are logically part of an A.S., they have a different nature. For this reasons, we treat them in Sect. 2.2.

2.2 Sensors and Noise

Sensor output can be captured during the simulation and analysed through third-part programs. Even for sensors an architectural description has to be provided, in the same manner as rigid bodies used to model A.S. architecture. Indeed, they are physical components. However, because of they interact with the external environment only by acquiring data, the description of physical properties of their design has a secondary role with respect to A.S. modelling. Thus, the behaviour of sensors plays a key-role for a realistic representation of the system. Except for few features (such as the acquiring frequency), each sensor has quite unique set of features that have to be taken into account separately.

Table 1. Modelling choices for Hokuyo LIDAR in ArcTurius

Parameter	Value
Range measurement	$[-\pi/3, \pi/3]$ [radians]
Update rate	20 [Hz]
Accuracy	±30 [mm] *(distance* < 10 m*)*
Measurement resolution	1 [mm]
Angular resolution	0.25 [degrees]
# rays per cycle	640
Noise	Gaussian: $\mu = 0.0$, $\sigma = 0.01$

We decided to use a LIDAR on ArcTurius rover, and we represented its behaviour on the system. With respect to sensor modelling, the parameters taken into account have been taken directly from the datasheet of a real LIDAR, in this case an Hokuyo UTM-30LX Scanning Laser Rangefinder [8]. The choice has been motived by the fact that this kind of product intrinsically supports the ROS paradigm [9]. As a matter of fact, a ROS package, ready to be integrate in real simulations, already exists and it is supported and maintained [10]. Starting from its maximum and minimum specifications [8], we customized Hokuyo LIDAR model according to ArcTurius design. Indeed, we work with a low data-rate in order to keep the consumptions moderated, increasing the time of the mission. Moreover, an excessive data-rate can lead to a larger number of corrupted data. Table 1 shows our modelling choices for some parameters of Hokuyo LIDAR.

About the latter, sensors are noisy components and we have to consider the noise to enhance the realism of the simulation. Currently, it is possible to model the inferences who affect them through a Gaussian distribution with a moment parameterization (i.e., providing mean and standard deviation of the distribution) [11]. In this case, assuming to work within 10-distance meters, it is possible to simulate an accuracy of 30 mm assuming that the 99.7% of the measurement is correct. This is achieved modelling the noise with a Gaussian distribution whose mean is 0 and standard deviation is 0.01.

Data coming from a virtual sensor, whatever it is, can be used with a double goal. Firstly, they can be used by the virtual A.S. for any purpose. For example, the program responsible for navigation associated to A.S. behaviour can take decisions starting from data produced by sensors. As already mentioned, ROS handles this data exchange. However, the deepening of all the algorithms that rely on sensor data is not part of this paper. Regardless of the usage that A.S. modules make of data produced by virtual sensors, engineers in charge to find the best configuration of an A.S. upon a disaster can rely to a 3D sensor date representation that enhance the comprehension of the simulation. Thus, the second stated goal is facilitating the understanding of the telemetry results checking how the A.S. is seeing at each simulation step. This can lead to a faster decision making, reducing the effort in the interpretation of telemetry results whenever possible. In this respect, Figs. 3 and 4 shows the capturing of laser data through RViz [12], a 3D visualization tool for ROS. Its usage will be deepened as well in Sect. 3.

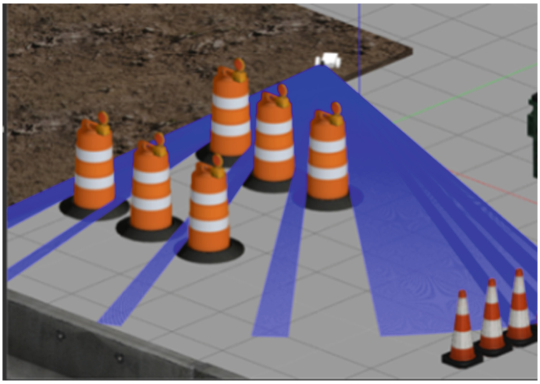

Fig. 3. Example of Sensor (LIDAR) model - Global view

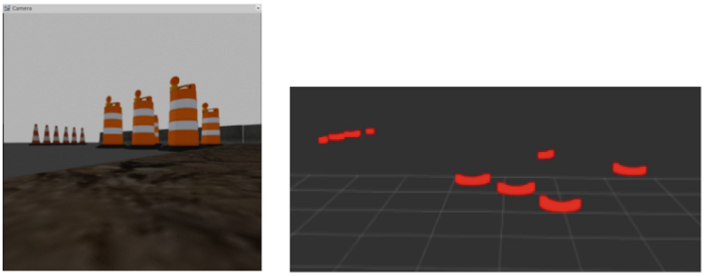

Fig. 4. Bottom, left: video view, right: 3D scene reconstruction with RViz.

2.3 Terrain Generation and Modelling

In SaR missions ground characteristics take a crucial role. Indeed, a correct representation of the ground permits to answer to many questions about the configuration of the A.S. during the simulation phase. Our simulation system is intended to speed-up the configuration of A.S. that support rescue teams during a mission. In this respect, rescue teams are mainly composed by rescuers and fire-fighters with limited engineering competences. The goal of this part is to avoid huge modification in A.S. configuration and mechanics once their arrival in catastrophe place. First of all, this would constitute a waste of time. Moreover, the environmental conditions do not always allow an easy intervention. A proper modelling of the terrain plays a key role in the preliminary evaluation of some issues related to the interaction of A.S. with the surrounding world.

For example, taking as a reference ArcTurius rover, we can evaluate if the depression of the terrain involve in a roll-over of it, or whether wheels of a different size are more appropriated for the mission, for example. Performing these kinds of simulations in advance permits rescuers to save time and to avoid a task for which they have limited competences.

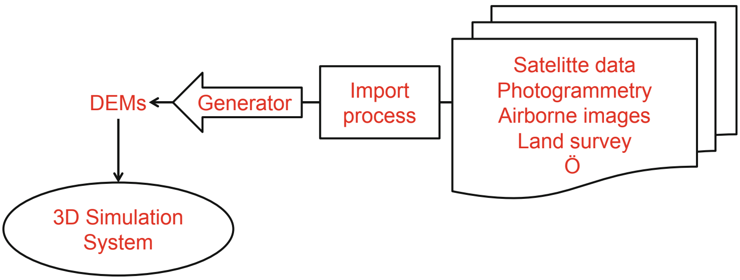

Fig. 5. Terrain input process to simulation system

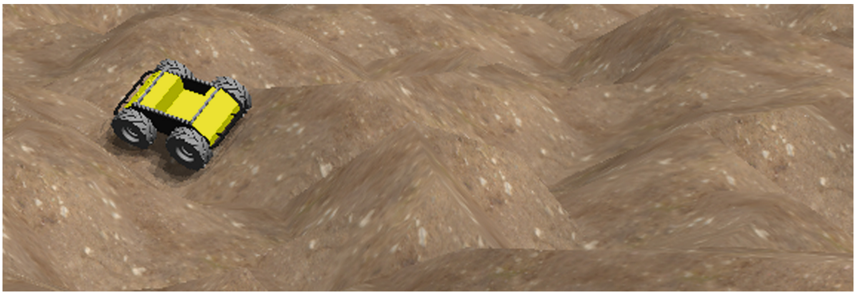

Fig. 6. South-west of Haiti imported in Gazebo-3D

Figure 5 shows the process to integrate realistic terrains in the simulation system. The starting point are real images obtained by different and real sources such as LIDAR, radar, cameras, photogrammetry, InSAR, land surveying and their combinations. There are processed to generate Digital Elevation Models (DEMs) that constitute the real input of the system [13]. DEMs show surface elevation data, sampled at regularly-spaced horizontal intervals. If the terrain elevation is represented as a grid of elevations (raster), DEM can be seen as a grey-scale height-map. In height-maps, elevation data are represented by associated the colour of a grey-scale pixel with an elevation. Specifically, a white pixel corresponds to the point of maximum elevation, whereas a black pixel represents a point characterized by the minimum depression for the considered ground. Figure 6 shows the result upon the application of the schema in Fig. 5 that we used in the context of our project. The terrain represents an area in the south-west of Haiti and it has been generated by merging data coming from ASTER [14], USGS NED [15] and SRTM30+ [16]. ASTER is a well-known survey of elevation of Earth, characterized by a high coverage and resolution (about 30 m). SRTM30+ provides coarse data (900 m of resolution) and they are used to the general contours of the land. USGS NED provides precise elevation data (up to 10 m of resolution) for zones close to United States geography.

Terrains generated from DEMs files do not usually contain external components that can be useful for simulation purposes, such as rocks, rubble, fails, etc. However, they can be integrated in the ground, according to the extended process present in Fig. 7. A possibility is to use a Drone that runs a low altitude to map them in a precise way. However, they can be generated and integrated in a realistic way. We can use mathematical functions, common in remote sensing applications, in order to estimate the roughness of the terrain. After, the physical properties of obstacles (such as density, bounciness, etc.) have to be defined, at least providing a range of values for each parameter. Finally, Monte Carlo method can be used to generate random distribution of these actors distributed over the terrain. To physically see the obstacles populating the world, a 3D description of them has to be provided too. Figure 5 schematically resumes the steps for the generation of terrains based on DEM models.

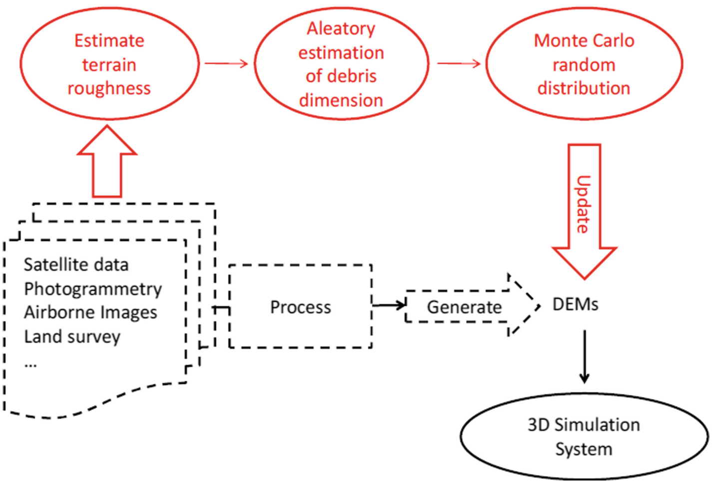

Fig. 7. Enhanced input integration process (DEM based example).

2.4 Environment Parameterization

Environmental conditions can dramatically change the results given by a simulation and they have to be modelled in the more realistic way possible. For example, a simulation that involves the usage of A.S. equipped with LIDARs can suffer from the light in particular times of the day. By properly modelling the light, it is possible to derive a different positioning of LIDARs with respect to the chassis of the rover depending by the time of the day. For example, lights can be described specifying position, orientation, attenuation factors and direction, among others. Many other perturbations can derive by the different state of the environment, and each of them would require a strong analysis performed by a physician and they are not deepened here. For example, the study of how weather conditions, such as rain, snow or wind, temperature, magnetic field, other forces act in the system or in the rest of the environment (e.g., the terrain). In this respect, lots of work has to be performed, both from a physical and a computer science point of view. In our view, this is a key-point for the simulation of A.S. in the domain of disaster management

Indeed, through simulation it is possible to evaluate the behaviour of an A.S. in a post-disaster area, also considering external environment conditions that can limit the domain of validity of the simulation. A huge realism is obtained at cost of a precise physical modelling, which is usually done by people expert in the domain.

3 Graphical Engine, Physical Engine and Communications

To assure the communication between the different parts of the simulation system we chose Robot Operating System (ROS) [17]. ROS is commonly used by the scientific community and despite its name, rather than a classical Operating System, it is more a framework that contains libraries, code, modules, configuration files, third-part software in support to robotics development. Its main goal it is to provide a standard communication interface to the different, heterogeneous elements that compose a robotics system. The latter are seen as "Nodes". Nodes can produce data, consume data or both. The exchange of data happens following a standard syntax, even though nodes can have a very different nature each other. In particular, the exchange of data may happen in a continuous way (e.g., sensors nodes that send their value to a 3D-visualizer node) or they can occur one-off (e.g., a master node that express a request and wait for a reply to a slave node). In the first case, the nodes exchange data through ROS messages sent over ROS topics, with a reference paradigm similar to the Publisher/Subscriber. In the second case ROS messages are sent over artefacts named ROS services, with a reference model Client/Server. In our respect, ROS is used to connect all the elements intrinsic to the simulation (such as rover, environmental actors, sensors, etc.) and the data analysis tools that we use for the evaluation of the measurements. However, ROS is more than a meta operating system that provides a standard connection way. As already mentioned, it includes many other modules that can be useful to robot simulation. In this article, we especially focused on two of them: RViz and Gazebo-3D.

RViz: RViz [12] is a 3D visualization environment for ROS. The decision-making for A.S. configuration can be difficult without exactly knowing what the A.S. is seeing.

Data obtained from telemetry can be difficult to be interpreted, not only in 3D but also in 2D. Indeed, these data often are a set of coordinates with information associated. Instead, visualising the world with rover eyes in 3D coordinates allows an easier debug of engineers in charge to configure an A.S. before a mission. Naturally, the telemetry continues to have a key-role during the operating mission of the A.S. There are two ways to inject simulation data into RViz:

1. It understands data from cameras, lasers, point-clouds, coordination frames, etc. sent through ROS topics
2. It is able to receive customized visual markers obtained by sending primitive shapes such as coloured cubes, arrows, etc.

A combination of two methods is accepted too. In fact, it is the most common method among algorithms in ROS navigation package [18].

Gazebo-3D: Many graphical engines are available in the domain. For several reasons, only hinted in this paper, we chose Gazebo-3D [7]. Gazebo-3D is a graphic engine that includes a physical engine such as ODE [19]. It is able to perform 3D physical simulations while displacing a rover within a surrounding virtual world. Moreover, it can be used to rapidly evaluate different kinds of algorithm, to design robots and to simulate their behaviour in realistic scenarios. For that, it is necessary to realise realistic models of A.S., sensors, actuators, environment conditions and external actors. Models are expressed with a XML-like syntax following SDF specifications [20]. They have to describe the physical properties (e.g., shape, collision domain, kinematics, friction, etc.) of the different actors that compose the simulation. It is possible to address a behaviour through the usage of plugins. For example, sensors shall transmit their measurements to a third-part analyser. Moreover, rover shall be able to move in a external environment and to interact with it, eventually basing its decisions on the values taken by the sensors. Gazebo-3D was chosen because of the experience of our laboratory on it and because it merges the graphical and the physical engine while maintaining a strong level of customization. Indeed, starting from the capabilities offered by Gazebo-3D, it is possible to define custom components able to interact with the simulator. The customization can occur both at graphical and behavioural level. Even though its expressiveness is very powerful, it is not useful whether not supported by a strong modelling of system features as well as an appropriate behaviour description of the various components through the usage of plugins. The latter can use Gazebo API to interact with the engines and can be developed in a custom way to describe the behaviour of rovers/drones, sensors, external agents, etc. Simulation data can be transferred from a virtual environment in Gazebo-3D to analysers through ROS.

4 Expected Results

The purpose of the system is exploiting the results of the 3D simulation in order to identify possible situations in which an A.S. may underperform (or not working at all) and to take countermeasures before the mission. This can enhance the utility of A.S. and reduce the errors during a mission. Beyond the adequacy assessment, the other expected results are the mission success expectancy and the A.S. survival probability after the mission.

The research of an optimal configuration is realized following the cyclic pattern *research, simulations, modification*. That means that there is always a set of simulations that verifies the configuration after an integration of a component. If the simulation results show that the configuration is not optimal after the integration, it is possible that the component is not suitable to achieve the proposed goal. Thus, variation in the component architecture or behaviour can be tested and that lead to invent characteristics of the instrument you need even before knowing them.

5 Illustration of the Concept

The vast domain of this approach allows the representation of several and different case studies. For example, a 3D simulation can be performed to evaluate the positioning of a sensor, in order to find the position that minimises the number of useless samples taken during the mission. Another possibility, deepened in this section, is the evaluation of the mechanical configuration of the A.S. (a simple representation of ArcTurius rover, in this example) with respect to a terrain characterized by an non-negligible roughness level. In this set of simulations, we integrated in the simulation system the landing site of Apollo 15 (Apennine Mountains region) [21], whose elevation models has been provided by NASA organization. It has been selected because the represented area is characterized by an irregular ground, suitable for an adequacy analysis of mechanical equipment (e.g., wheels, chassis, etc.). Figure 8 shows a 3D-reconstruction of the terrain.

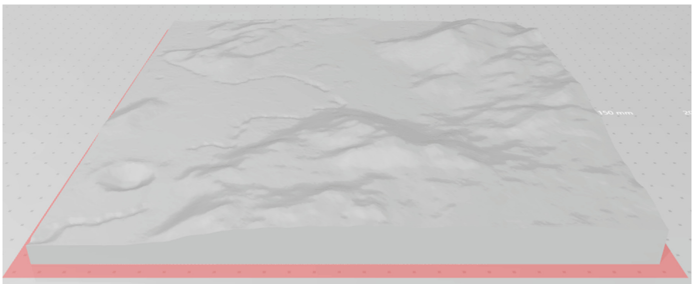

Fig. 8. 3D reconstruction of Apollo 15 landing site

In the simple representation of ArcTurius rover used for this simulation, the rover has been modelled with 3 boxes connected each other by a rigid junction, whereas wheels are represented using solid cylinders. Sizes and weights have been chosen according to original ArcTurius design [2]. In this set of simulations, we would like to verify whether ArcTurius rover is able to easily cross an irregular terrain such as the one that characterize Apollo region. If not, the results of the simulation can lead us to easily understand where is necessary to act.

Figure 9 shows the results of the first simulation. The simple representation of ArcTurius rover runs over the terrain in a straight line, but he fails to overcome a depression in the terrain. This is confirmed by the contacts points showed in Fig. 10. Indeed, the chassis of the rover touches the ground (red circle in Fig. 10) whereas a wheel is raised from the ground (the absence of contact points in the green square in Fig. 10).

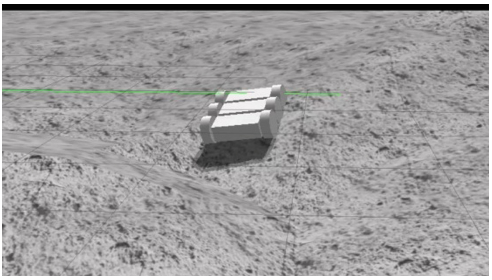

Fig. 9. Starving after a depression

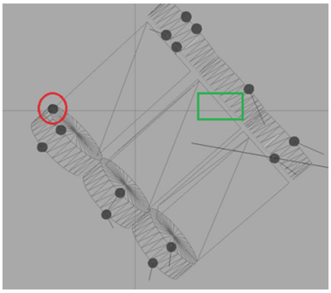

Fig. 10. Contact points (Color figure online)

This preliminary result leads us to take countermeasures in order to make the system adapted with respect to the surrounding environment. A first idea can be to enhance the power of engines, acting on torque parameter in the models. In fact, that leads the system in a success crossing of the site. However, this operation involves in a larger power consumption with the consequent reduction of the rover autonomy as well as the time mission.

For this reason, we wonder whether an architectural change can lead to a success, keeping the power consumption unchanged. We though to replace the wheels with other characterized by a bigger size and height. This lead to an architectural problem: the lack of free space implies to switch to an architecture characterized by 3 connected bodies and 4 wheels instead of 6, as depicted in Fig. 11. Also in this time, the autonomous system succeeded in the crossing of the area.

The research of a new configuration can involve several other critical issues that have to be taken into account before running the real rover in a post-disaster environment. With respect to the last illustration, the fact to have a standalone body can cause a balancing problem. Indeed, the body in the middle of the chassis has to be balanced in order to avoid awkward behaviours while crossing a non-straight terrain. In the last simulation, we injected a component in the body in order to cause a parasitic sway of the system. This is showed in Fig. 12, where we can notice the centres of mass of wheels (right and left side) and of the central body after the integration of a component, located in correspondence of the green box. In the bottom of the image, we can notice the contact points of the body to the ground, behaviour that shall be avoided in a real context.

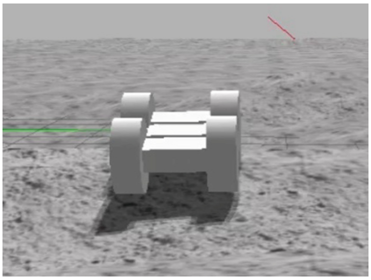

Fig. 11. Architectural variation

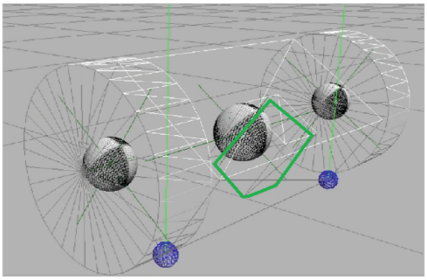

Fig. 12. Integration of a new component – Balancing problem (Color figure online)

Before the integration a new component in the system (such as sensors, batteries, etc.) we expect to perform this and many other kind of analysis. In this respect, the immediate visual returning given by the 3D system can enhance the understanding of engineers in charge to configure an autonomous system in order to make it adapted before a mission.

6 Discussion and Conclusion

The use of autonomous engines either on the ground (Rover) or flying (Drone) is without any doubt an improvement. They are able to attain unattainable and dangerous locations. Moreover, they are less sensitive to both environmental conditions such as meteorology and stressful situations for human beings. However, in order to achieve this autonomy several difficulties have to be resolved. In the first hours after a disaster, during the rescue team deployment, our 3D-based simulation system allows a real gain in terms of timing. In conclusion, from the proposed approach we can expect to:

- Increase the knowledge of the devastated area
- Validate the A.S. behaviour (moving, data acquisition, etc.)
- Validate the adaptation of the payload sensors configuration for this (these) mission (s)

This is a first work towards the proposal of a new approach than can support the support rescue missions that expect a contribution of A.S. Today we are able to generate terrain data based on satellites and DEM model, to customize them and to import them in the simulation environment. Moreover, we can correctly model rigid bodies and their physical properties, such as the mechanical parts that compose a rover with their interconnection points (junctions). We are able to associate a behaviour to the rover, to model the most common sensors (such as camera, LIDARs, odometers) and display their results. On the other hand, more work has to be done in the context of environment conditions. Part of the future work will include the reaction of the terrain (intended as the modification of its physical parameters) to a natural event such as rain on snow. This work will open the door to many others scenarios in the context of simulation in support to disaster risk reduction [22–24].

References

1. Mukhopadhay, B., Bhattacherjee, B.: Use of information technology in emergency and disaster management. Am. J. Environ. Prot. **4**, 101–104 (2015). https://doi.org/10.11648/j.ajep.20150402.15
2. Tanzi, T.J., Isnard, J.: Autonomous system for data collection: location and mapping issues in post-disaster environment. Comptes Rendus Physique (2019). https://doi.org/10.1016/j.crhy.2019.03.001
3. Chandra, M., Tanzi, T.J.: Drone-borne GPR design: propagation issues. In: Journées scientifiques de l'URSI-France (JS'17) (2017)
4. Servigne, S., Gripay, Y., Pinarer, O., Samuel, J., Ozgovde A., Jay, J.: Heterogeneous sensor data exploration and sustainable declarative monitoring architecture: application to smart building. In: First International Conference on Smart Data and Smart Cities, 30th UDMS, 9 September 2016, Split, Croatie, pp. 97–104 (2016). https://doi.org/10.5194/isprs-annals-iv-4-w1-97-2016
5. Tanzi, T.J., Roudier, Y., Apvrille, L.: Towards a new architecture for autonomous data collection. ISPRS Geospatial Week 2015: Workshop on civil Unmanned Aerial Vehicles for geospatial data acquisition, La Grande Motte (Montpellier), France, 1–2 October 2015

6. Stormont, D., Allan, V.: Managing risk in disaster scenarios with autonomous robots. J. Syst. Cybern. Inform. **7**, 66–71 (2009)
7. Koenig, N., Howard, A.: Design and use paradigms for Gazebo, an open-source multi-robot simulator. In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat.No.04CH37566), Sendai, vol. 3, pp. 2149–2154 (2004). https://doi.org/10.1109/iros.2004.1389727
8. Hokuyo UTM-30LX Scanning Laser Rangefinder. https://www.hokuyo-aut.jp/search/single.php?serial=169
9. ROS Components UTM-30LX. https://www.roscomponents.com/en/lidar-laser-scanner/87-utm-30lx.html
10. Package Summary hokuyo node. http://wiki.ros.org/hokuyo_node
11. Burgard, W., Fox, D., Thrun, S.: Probabilistic Robotics. The MIT Press, Cambridge (2005)
12. Kam, H.R., Lee, S.-H., Park, T., Kim, C.-H.: RViz: a toolkit for real domain data visualization. Telecommun. Syst. **60**, 1–9 (2015). https://doi.org/10.1007/s11235-015-0034-5
13. Digital Elevation Models. https://en.wikipedia.org/wiki/Digital_elevation_model
14. ASTER - Global Digital Elevation Map. https://asterweb.jpl.nasa.gov/gdem.asp
15. USGS DEM - National Elevation Dataset (NED). https://en.wikipedia.org/wiki/USGS_DEM
16. SRTM30+ - Shuttle Radar Topology Mission. https://www2.jpl.nasa.gov/srtm/
17. Quigley, M., et al.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software, vol. 3 (2009)
18. ROS navigation package. http://wiki.ros.org/navigation
19. Drumwright, E., Hsu, J., Koenig, N., Shell, D.: Extending open dynamics engine for robotics simulation. In: Ando, N., Balakirsky, S., Hemker, T., Reggiani, M., von Stryk, O. (eds.) SIMPAR 2010. LNCS (LNAI), vol. 6472, pp. 38–50. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17319-6_7
20. SDF data format. http://sdformat.org/
21. Apollo 15 landing site. https://nasa3d.arc.nasa.gov/detail/Apollo15-Landing
22. Apvrille, L., Tanzi, T.J., Roudier, Y., Dugelay. J.-L.: Drone "humanitaire": état de l'art et réflexions. Revue Française de Photogrammétrie et de Télédétection, N 213-04-26, pp. 63–71 (2017)
23. Tanzi, T.J., Chandra, M., Isnard, J., Camara, D., Sebastien, O., Harivelo. F.: Towards drone-borne disaster management: future application scenarios **III-8**, 181–189 (2016). https://doi.org/10.5194/isprs-annals-iii-8-181-2016
24. Chandra, M., Tanzi, T.J.: Wave propagation and radar system. Aspects for designing a "drone borne" GPR for humanitarian application. In: IEEE Conference on Antenna Measurements & Applications (CAMA), Antibes, France (2014)