3D Simulation for Disaster Management: toward a new approach

Tullio Tanzi, Ludovic Apvrille

3rd URSI AT-AP-RASC, Gran Canaria, 29 May – 3 June 2022

3D Simulation for Disaster Management: toward a new approach

Tullio J. Tanzi(1), Ludovic Apvrille(1)
(1) LTCI, Telecom Paris, Institut Polytechnique de Paris,
(2) France http://www.telecom-paris.fr

Abstract
Recent progress in modern technology can enhance the definition of disaster recovery management strategy. Rescue teams can rely on Autonomous Systems (A.S.) during recovery operations, dispatching to them various tasks. A.S. can reach locations that may be unattainable or dangerous for humans. However, before sending the autonomous system to the catastrophe area, it is important to verify its adequation to the environment and to the mission objectives. The simulation provides an assessment of this adaptation between the autonomous system and its expectations.

1 Autonomous system for support disaster Management

Recent progress in modern technologies can help in enhancing strategies for disaster recovery management. For instance, rescue teams can rely on Autonomous Systems (A.S.) during recovery operations, dispatching to them various tasks. A.S. can reach locations that may be unattainable or dangerous for rescuers [1-3].

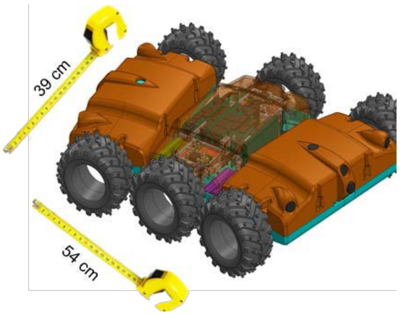


Figure 1. Autonomous System ArcTurius (Rover). 3D CAD view.

In this respect, the usage of autonomous systems (A.S.), such as Rovers or Drones, can assist recovering operations for many different tasks [4]. For instance, A.S. can be used for rapid terrain mapping or to scan the affected areas to find survivors, among others. The need for measurement systems operating in total autonomy has existed for a long time, but accurate enough measurement technologies were not yet available [5, 6]. In the past, acquired data were not frequently updated, probably because former technologies were only producing environment snapshots rather than permanent monitoring. Nowadays, wired and mobile communication networks can easily dispatch real-time measurements [7, 8].

We suggest relying on modeling and accurate simulations of A.S. in order to verify their adequation to targeted missions. The paper first elaborates on the architecture of A.S. before explaining how adequation can be efficiently studied.

2 Context

TTool is a tool helping in the design of embedded systems. TTool offers a UML/SysML interface, and simulation and formal verification (safety, security, performance). TTool has several development stages:

- Partitioning of embedded systems with DIPLODOCUS
- Design of embedded software with AVATAR
- Design of safe and secure embedded systems with the SysML-Sec environment

Gazebo allows Robot simulation based on roboticist's toolbox. Gazebo makes it possible to test algorithms, design robots, perform regression testing, and train AI system using realistic scenarios. Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. It uses a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces.

We have already use TTool to produce proofs concerning safety, performance or security properties of our various systems (Drone, Rover, etc.). We have also designed, based on the Gazebo framework, a 3D simulator to test and assess the behavior of our various autonomous systems in a representative environment of the disaster area and the conditions that prevail there.

2.1 Software and hardware architecture assessment

Autonomous vehicles and other robots have been proposed for disaster relief efforts. Our first case study describes the design of a rover which will search through rubble for disaster victims. The rover is equipped with telemetric sensors, located in the front, rear, top, and sides. These sensors allow the rover to detect obstacles and navigate the terrain autonomously [9]. The rover adjusts its acquisition behavior based on the situation. When it detects no obstacles in proximity, the rover decreases its sampling rate, assuming that no obstacles will suddenly appear in its path. When an obstacle is detected in close proximity, or within its "safety bubble", the rover adapts its behavior and increases its rate of acquisition. When the rover has detected obstacles in a very close proximity, exact distances to obstacles become more critical.

Precise distance calculation depends not only on the telemetric sensor measurements, but also on ambient conditions. Therefore, to obtain an exact measurement, temperature and pressure sensors must be used. Indeed, the rover must be able to respond to obstacles within a set time frame – i.e., a maximal latency – to avoid collisions. TTool can be used to closely define safety bubbles and time frames. We begin by modeling at the partitioning level using TTool/DIPLODOCUS. The rover consists of a main controller which receives data from a distance sensor and temperature sensor, which it uses to determine motor commands sent to the motor control, as shown in the functional view shown in Figure 2. The main controller behaviour and sampling rate of the distance sensor depends on the proximity of an obstacle (far away, intermediate, close).

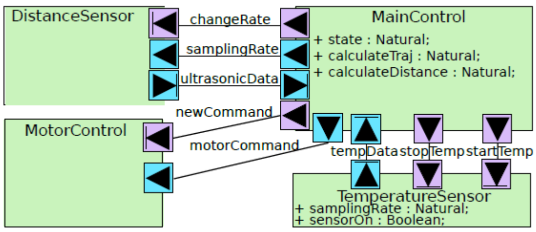


Figure 2. Functional Model of Autonomous System Arcturius (Rover). TTool view. [10]

After mapping these functions in a model of the rover architecture, we can use TTool simulator to evaluate the latency between e.g. the reception of a signal by the ***DistanceSensor*** and a corresponding motor reaction called ***motorCommand***. We assume the rover moves at 6 km/h, thus covering a distance of 100 meters per minute. Once simulation traces have been generated, TTool can determine the minimum / average / maximum latency between two events, as well as the standard deviations (Fig. 3).

Signal	Partitioning level				Software design level				
	min	max	avg	std dev	min	max	avg	std dev	SoCLib
s(tempData)-> r(tempData)	4	4	4	0	0	4	0.2	0.8	21.7
s(ultrasonicData)->r(ultrasonicData)	16	52	34	18	0	56	11.6	17.9	20.3
s(motorCommand)-> r(motorCommand)	5.7	10.3	8	2.3	0	16	4.9	3.3	32.3
s(ultrasonicData)-> r(motorCommand)	2	2	2	0	11	24	13.2	2.5	38.0
r(rultrasonicData)-> s(changeRate)	4	42	23	19	0	68	10.6	13.0	45.7

Figure 3. Calculated latencies. [10]

2.2 A.S. Behavior assessment

The use of autonomous systems constitutes a major progress in the support of a crisis. But, to work properly and to reach the desired level of autonomy, they have to be correctly configured though. Indeed, errors on A.S. configuration can lead to imprecise or erroneous data and, consequently, erroneous decisions can result from them. For this, 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 another 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.

To obtain this assessment, we propose a 3D-simulation, which can be used to find the best way to proceed for rescue teams. Briefly, an initial map of the real world is taken, for example through satellites' data, then it is injected into 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

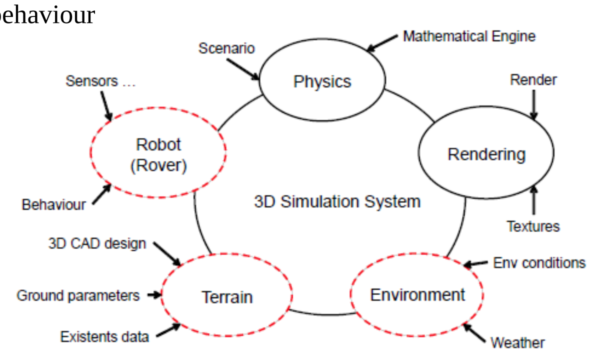


Figure 4. 3D Simulation System general architecture. [11]

. 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 4 shows an overview of the 3D-simulation framework. The latter is built upon several computational blocks that communicate through the exchange of high-level messages. The main components are:

1. A graphical engine that includes the rendering of textures, lights, shadows, etc.
2. A 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 creates realistic environment actors and conditions close to real disaster areas.

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 e.g., a rover structure, in order to find the position that minimizes 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 a non-negligible roughness level. For instance, we integrated in the simulation system the landing site of Apollo 15 (Apennine Mountains region) [11], whose elevation models have 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 5 shows a 3D-reconstruction of the terrain

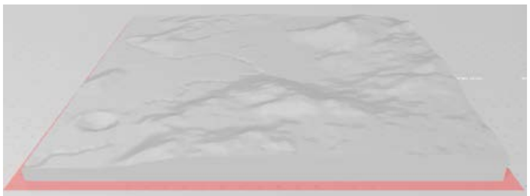
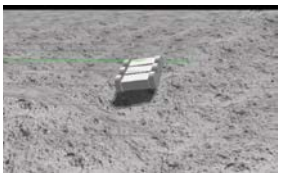


Fig. 5. 3D reconstruction of Apollo 15 landing site [11]

For 3D simulation, the ArcTurius rover has been abstracted as 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 [7]. 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 figure out which structural improvements are necessary.

Figure 6 shows the results of the first simulation. The simple representation of ArcTurius rover runs over the terrain in a straight line, but it fails to overcome a depression in the terrain. This is confirmed by the contacts points showed in Figure 7. Indeed, the chassis of the rover touches the ground (red circle in Figure 7) whereas a wheel is raised from the ground (the absence of contact points in the green square in Figure 7).



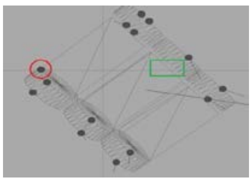
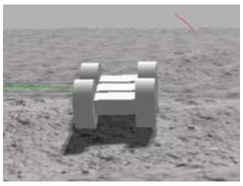


Fig. 6. Starving after a depression. **Fig. 7**. Contact points

This preliminary result leads us to improve the system with respect to the surrounding environment. A first idea can be to enhance the power of engines, acting on torque parameter in the models. By doing so, we were able to successfully cross the site. However, this modification resulted in a higher power consumption thus reducing both rover autonomy and time mission.

Thus, we investigated other solutions keeping the power consumption unchanged. Replacing wheels with bigger ones could help, but this leads 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 Figure 8. With this option, the autonomous system succeeded in crossing the area.



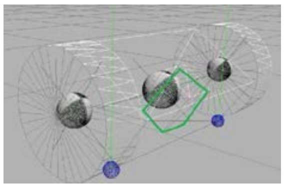


Fig. 6. Starving after a depression**. Fig. 7**. Contact points.

More generally, the identification of a better 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 behaviors 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 shown in Figure 9, where we can notice the center of mass of wheels (right and left side) and of the central body after the integration of a new component, located in the green box. In the bottom of the image, we can notice the contact points of the body to the ground: such situation shall be avoided in a real context. Before the integration a new component in the system (such as sensors, batteries, etc.) we expect to perform this and many other kinds 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.

3 Mixed 3D-digital simulation

As shown in previous section, 3D simulation can help designing a rover for a given mission. Yet, 3D simulation does not take into account the behaviour of the digital platform. For instance, if a decision of the 3D simulation were to integrate 6 wheels instead of 4, we need to check if the current digital platforms has the necessary computation power to drive in real-time the six wheels. This remarks also applies to sensors: improving the frequency of data acquisition can lead to extra computations, to contentions on buses or memories. This leads us to state that it is mandatory to design both physical and digital parts all together for a given mission. To do so, we propose to join our modeling and simulation solutions, as shown in Figure 10. The physical simulation generates values that are fed into the sensors simulated in TTool. When TTool simulator receives new sensor values, it executes the corresponding functions, and outputs values to actuators. Values from actuators are then taken as input to the 3D simulation in order to impact the dynamics of the Rover e.g. acceleration, turns, etc.

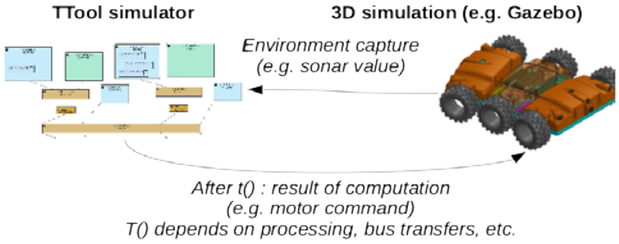


Fig. 10. Combining TTool and 3D simulation

For instance, when a sonar generates a new value – according to a given frequency – this value is transferred to the digital platform simulator. The latter stores this new value in a memory, and triggers the corresponding task, running e.g. on a processor [12]. Yet, the latter may be occupied running another more urgent task, so the processing of the sonar value is delayed. Then, the sonar task executes (again, it takes some time, depending on the processor capability), and an output is written to the memory of an actuator: the processor-to-actuator bus may be occupied, thus delaying again the computation. Finally, the overall digital platform time taken to produce the result impact the physics since e.g. the motor is activated into the 3D simulator only once the corresponding delay has elapsed.

4 Conclusion

Risk management requires new agile and autonomous systems to increase the effectiveness of rescue operations. A basic formulation of the paradigm to be solved could be: "to provide good information, with the good person, the good moment, for the good decision". In practice, that consists in providing to safety actors the technical solutions that enable them to be informed, to have tools of assistance (real-time cartographic support, simulations…), to transmit their directives, to verify the current course of operation. This requires the deployment of sophisticated solutions relying on two skills:

- Skills of rescuers who know how to act with effectiveness and are conscious of the risks that the crises can generate,
- ITC skills for actors involved in the definition and implementation of technical solutions likely to help the experts in their works.

To take good design decisions, actors in the IT domain must have solutions to quickly and efficiently reproduce disaster zones. The solution can join together physical and digital aspects of autonomous systems in order to customize them for given areas or missions.

5 References

1. P. Wilkinson and D. Cole, “The Role of the Radio Sciences in the Disaster Management,” Radio Science Bulletin, vol. 3358, pp. 45–51, 2010.
2. F. Lefeuvre and T. Tanzi, “International Union of Radio Science, International Council for Science (ICSU), Joint Board of Geospatial Information Societies (jBGIS),” in United Nations office for outer Space Affairs (OOSA), 2013.
3. D. Camara. Cavalry to the Rescue: Drones Fleet to Help Rescuers Operations over Disasters Scenarios. In IEEE Conference on Antenna Measurements & Applications (CAMA).Antibes, France, 2014
4. Daniel & Allan, Vicki Stormont, "Managing Risk in Disaster Scenarios with Autonomous Robots," Journal of Systemics, Cybernetics and Informatics, vol. 7, 2009.
5. T. Tanzi & J. Isnard, "Autonomous system for data collection: Location and mapping issues in post-disaster environment." ComptesRendus Physique, 2019.
6. Y. Kawasumi, Communications for rural and remote areas, ITU News magazine, 05, 2004
7. T. Tanzi & J. Isnard. "Introduction to Public safety networks" Chapter of book, In Public safety Book volume 1: Overview and Challenges. Published by Wiley-ISTE. 2015.
8. T. Tanzi and F. Lefeuvre, “Radio Sciences and Disaster Management,” C.R. Physique, vol. 11, pp. 114–224, 2010.
9. Tanzi, T., Chandra, M., Isnard, J., Camara, D., Sebastien, O., and Harivelo, F. (2016). Towards ”drone-borne” disaster management: Future application scenarios. In ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, volume III-8, pages 181–189.
10. Daniela Genius, Letitia W. Li, Ludovic Apvrille, Tullio Tanzi. Multi-level Latency Evaluation with an MDE Approach. 6th International Conference on Model-Driven Rngineering and Software Development (MODELSWARD 2018), Jan 2018, Funchal, Portugal. ⟨10.5220/0006535902950302 ⟩. ⟨hal-01670546⟩
11. Tullio J. Tanzi, Matteo Bertolino. Towards 3D Simulation to Validate Autonomous Systems Intervention in Disaster Management Environment. 4th IFIP Conference on Information Technology in Disaster Risk Reduction (ITDRR- 2019), October 9 - 10th, 2019. Kyiv, Ukraine
12. A. Enrici, L. Apvrille, R. Pacalet, "A Model-Driven Engineering Methodology to Design Parallel and Distributed Embedded Systems", ACM Transactions on Design Automation of Electronic Systems (TODAES), Volume 22 Issue 2, January 2017