

Prof. Tullio J. TANZI

Prof. Ludovic APVRILLE

{firstName.lastName}@telecom-paris.fr



Gran Canaria Spain
29 May - 3 June, 2022



3D Simulation for Disaster Management: Toward a new approach

Design genesis of an autonomous post-disaster response system

Context: disasters

- **Disaster** means **chaotic scenario**
- Lives in danger means **time matters**
- Chance to survive strongly decreases after 72 h



Problematic and our proposal

- **Intervention in (large) devastated areas**
 - Global (and fast): covering an area asap
 - Rapid mapping
 - Detecting victims
 - Use of EM from personal objects
 - GPR
 - Shall handle hostile environments
- **Our idea: designing a rover with a mission-configurable payload**
 - Autonomous - Resilient to communication loss
 - Mapping
 - Optical and IR images
 - Electromagnetic features (radars, including GPR)

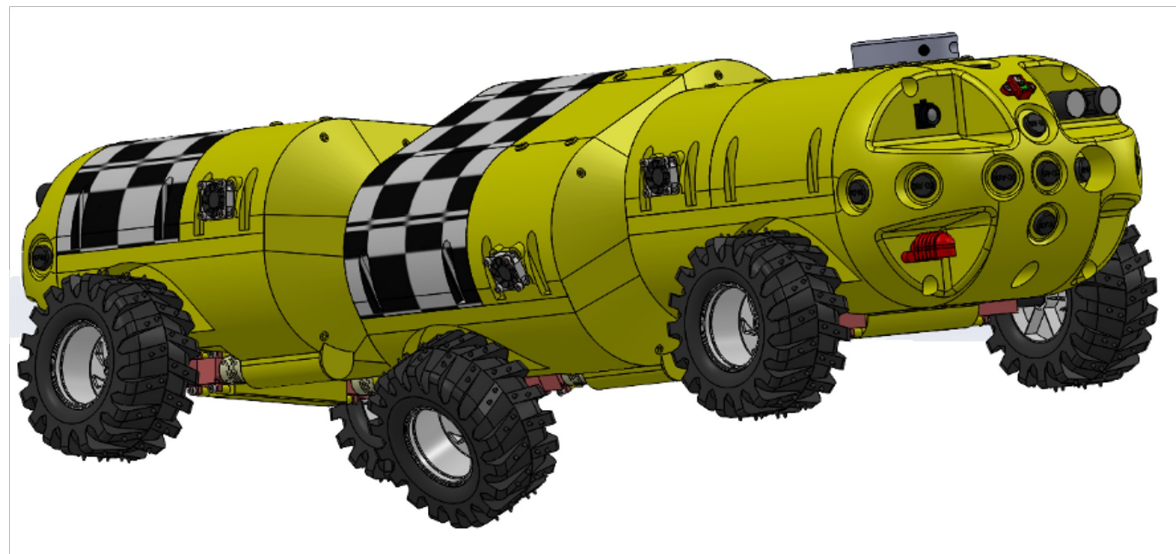


TECHNISCHE UNIVERSITÄT
CHEMNITZ

Prof. Madhu Chandra

Let's welcome ArcTurius!

- **Modular**
- **Customizable payload**
 - Configurable slots for custom sensor
- **Embedded power: from 1 to 3 kWh**
- **Weight: 10 kg for classical configuration, up to 15 kg**



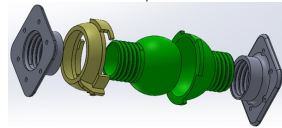
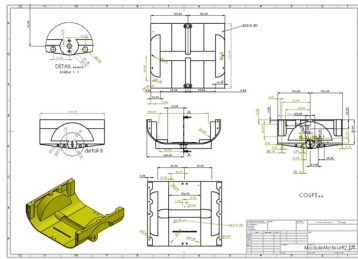
ArcTurius: typical sensors

- **Inertial unit**
 - 6 and 9 DOF accelerometer
- **Temperature, pressure, humidity**
 - Internal (LiPo, motors), external (environment)
- **Magnetometer**
- **Surroundings capture (LIDAR, Sonar, camera, etc.)**
- **Wheel rotation control for better traction control**
 - 3592 ticks per wheel revolution, 2 encoders per wheel
- **Power consumption tracking**
- **Attitude (anti-overturn control)**



ArcTurius: development process

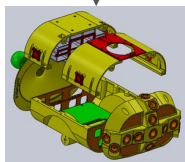
3D design



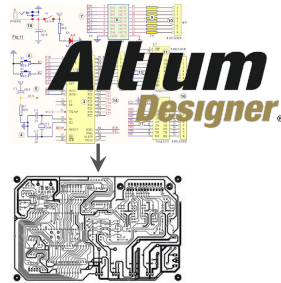
Mechanical Validation



3D Print



HW design



Embedded system

SW design

SysML Model

Safety verification

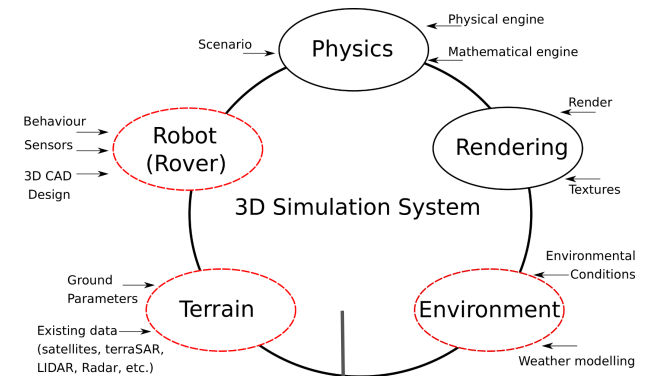
Code Generation

C/C++

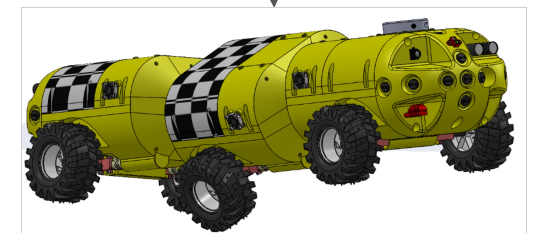


Supervisory SW

System Simulation

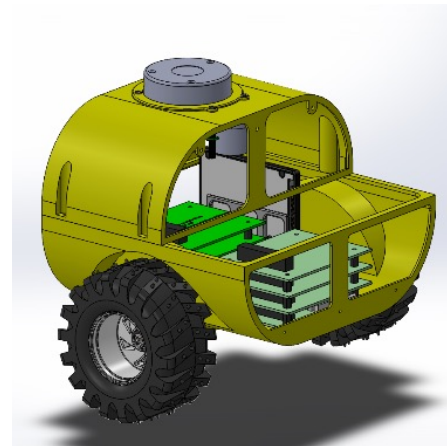
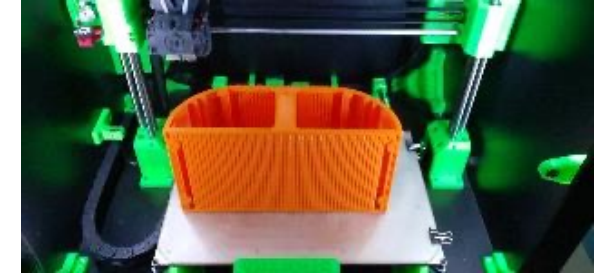
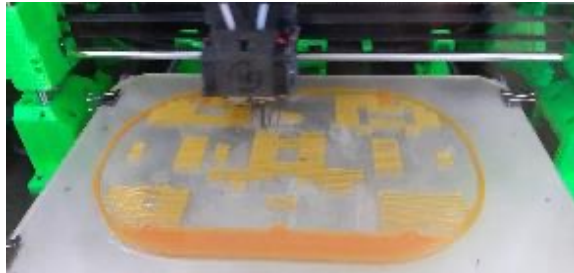


Configuration



Mission-specific system

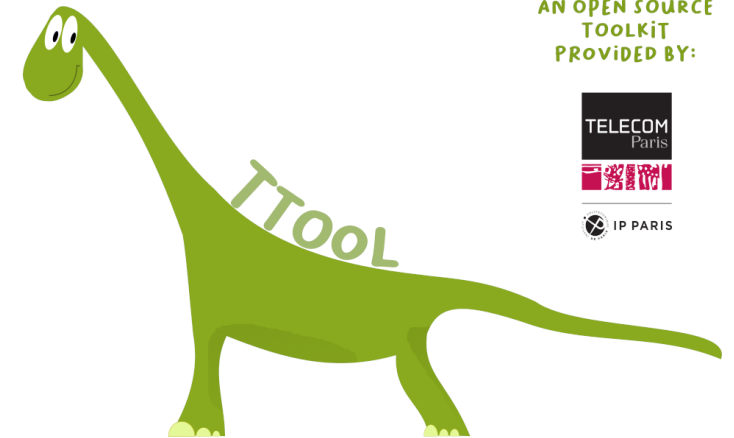
ArcTurius: 3D design and printing



- **Prototype: PLA, PETG**
- **Final system: carbon-loaded nylon**

ArcTurius: software design

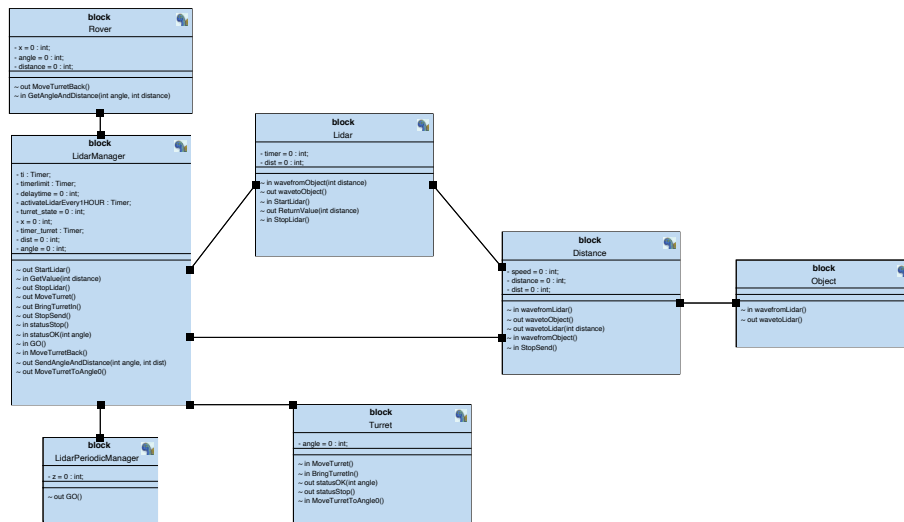
- **Objectives**
 - Designing the control software
 - Critical parts
 - Verifying safety properties
- **Our approach: using TTool**
 - Open-source
 - Well-known modeling language : SysML
 - Safety verification with a press-button approach
 - Simulation
 - Formal verification



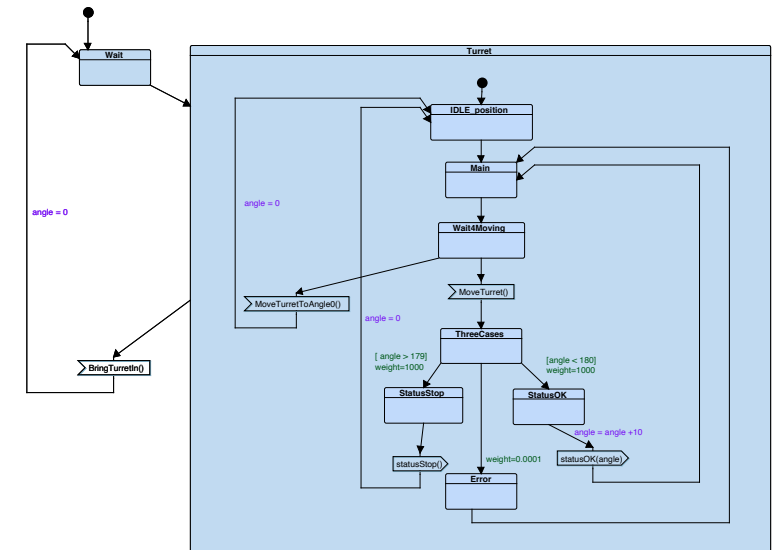
ArcTurius: software design of the LIDAR control

- Objectives

- Understanding how to **efficiently control the LIDAR**
- Measuring distance, handling of the angle at which the scan should occur, bringing back the LIDAR turret in case of upper obstacle
- **Verifying safety properties:** values returned by the LIDAR, liveness of the LIDAR control



Architecture



Behavior

ArcTurius: software design of the LIDAR control (Cont.)

- **How to express safety properties?**

```
Safety Pragmas  
A[] Lidar.dist<10  
LidarPeriodicManager.Start --> LidarPeriodicManager.Start  
LidarManager.Wait4Response --> LidarManager.WaitingForTurretActivation || LidarManager.MainLoop
```

- **Verification**

- Takes around 10 minutes
- Reachability graph: 4 millions of states, 8.6 millions of transitions
 - Manual analysis (after minimization), automated analysis (deadlocks)

```
Safety Pragmas  
✓ A[] Lidar.dist<10  
✓ LidarPeriodicManager.Start --> LidarPeriodicManager.Start  
✓ LidarManager.Wait4Response --> LidarManager.WaitingForTurretActivation || LidarManager.MainLoop
```


ArcTurius: software design of the LIDAR control (Cont.)

- **Model-to-C code generation**

```
/* Main loop on states */
while(__currentState != STATE__STOP__STATE) {
  switch(__currentState) {
    case STATE__START__STATE:
      traceStateEntering(__myname, "__StartState");
      __currentState = STATE__Main;
      break;

    case STATE__Start:
      traceStateEntering(__myname, "Start");
      makeNewRequest(&__req0__LidarPeriodicManager, 656, SEND_SYNC_REQUEST, 0, 0, 0, 0, __params0__LidarPeriodicManager);
      __req0__LidarPeriodicManager.syncChannel = &__LidarPeriodicManager_GO__LidarManager_GO;
      __returnRequest__LidarPeriodicManager = executeOneRequest(&__list__LidarPeriodicManager, &__req0__LidarPeriodicManager);
      clearListOfRequests(&__list__LidarPeriodicManager);
      traceRequest(__myname, __returnRequest__LidarPeriodicManager);
      __currentState = STATE__Wait4AnotherStart;
      break;

    case STATE__Wait4AnotherStart:
      traceStateEntering(__myname, "Wait4AnotherStart");
      waitFor((500)*1000000, (500)*1000000);
      __currentState = STATE__Main;
      break;

    case STATE__Main:
      traceStateEntering(__myname, "Main");
      __currentState = STATE__Start;
      break;
  }
}
```

ArcTurius: supervisory software

The screenshot displays the ArcTurius supervisory software interface, which is divided into several functional windows:

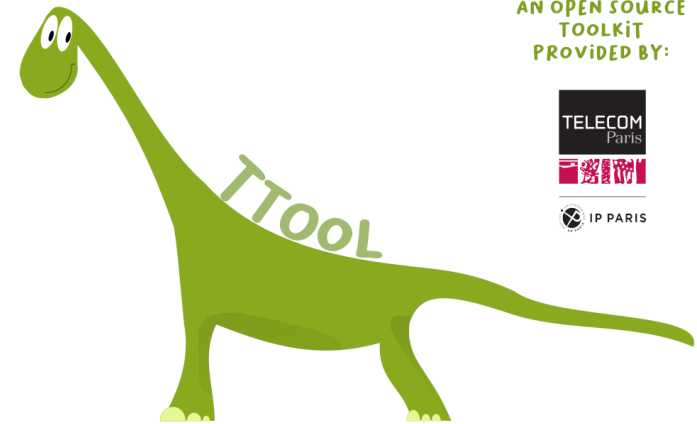
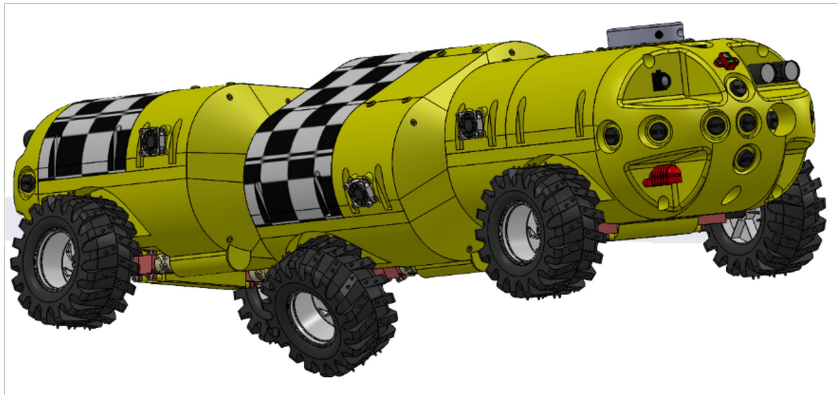
- Interface de contrôle de Tellus:** A control panel with buttons for 'Télémesure', 'Contrôle à distance', and 'Données'. It also includes a 'Commandes disponibles' section with options like 'Fermer', 'Déconnecter', and 'Données', and a 'Langues disponibles' section with flags for French, English, German, and Italian.
- Télémesure Tellus:** A central dashboard showing the rover's status. It features a top-down view of the rover with labels for 'Antenne Sonar', 'Antenne Frontale', 'Antenne Arrière', 'Antenne Vert. Avant', and 'Antenne Vert. Arrière'. Below this is a 'Gestion de l'énergie' section with five LIPo battery status indicators (LIPo 01 to LIPo 05) and three voltage regulator outputs (6 Volts, 5 Volts, 3.3 Volts). An 'Environnement' section shows gauges for 'Températures (°C)', 'Luminosité (Lux)', 'Humidité (%)', and 'Bruit dB(A)'. A 'STOP' button is prominently displayed.
- Contrôle à distance Tellus:** A navigation control panel with a 'Horizon Artificiel' (artificial horizon) and a 'Compas Magnétique' (magnetic compass). It includes a 'Cap Actuel' (current heading) display and a 'Changer le Cap' (change heading) control with a 'Nouveau Cap' (new heading) input field and an 'Appliquer' (apply) button.
- Données Lidar Slamtek:** A circular radar plot showing the range and intensity of laser returns from the environment.
- Données Lidar affichées:** A window showing the 'Signaux du Rover (Default Input Device)' as a blue waveform plot of 'Niveau Audio' (audio level) over 'Echantillons' (samples).
- IMAG0009.AVI:** A video player window showing a live feed of the rover in a laboratory setting.

Conclusion and perspectives

- **Objective: enhancing response to disasters**
- **ArcTurius: an almost ready-to-use rover**
 - Configurable
 - Interdisciplinary project
 - Expected to be used by operational teams to better handle risk
- **What's next?**
 - Currently: software integration, integration tests
 - Outdoor tests and global validation planned for summer 2022
- **Necessary to have a good collaboration between rescuers and researchers**

Thank you!

Any questions?



<https://ttool.telecom-paris.fr>