ETH zürich

Building a Legged Robot with ROS

Programming for Robotics: Case Study

Péter Fankhauser

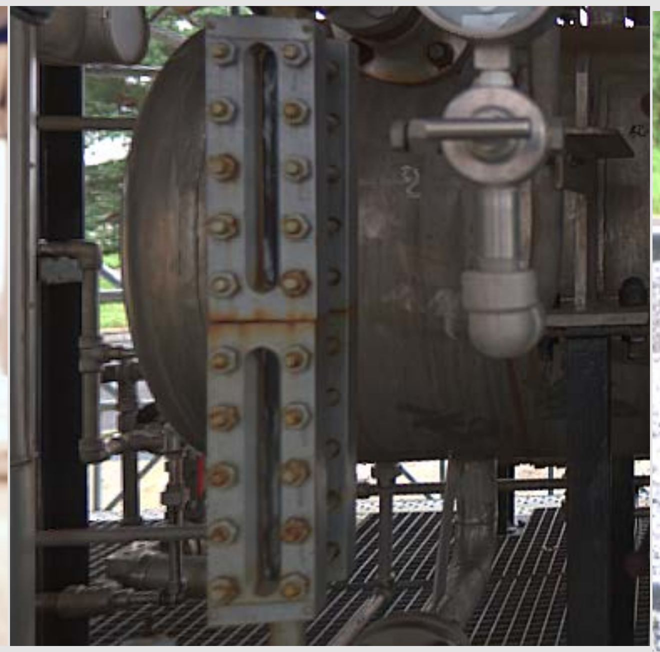
ANYmal RESEARCH

RSL Robotic Systems Lab


ANYmal

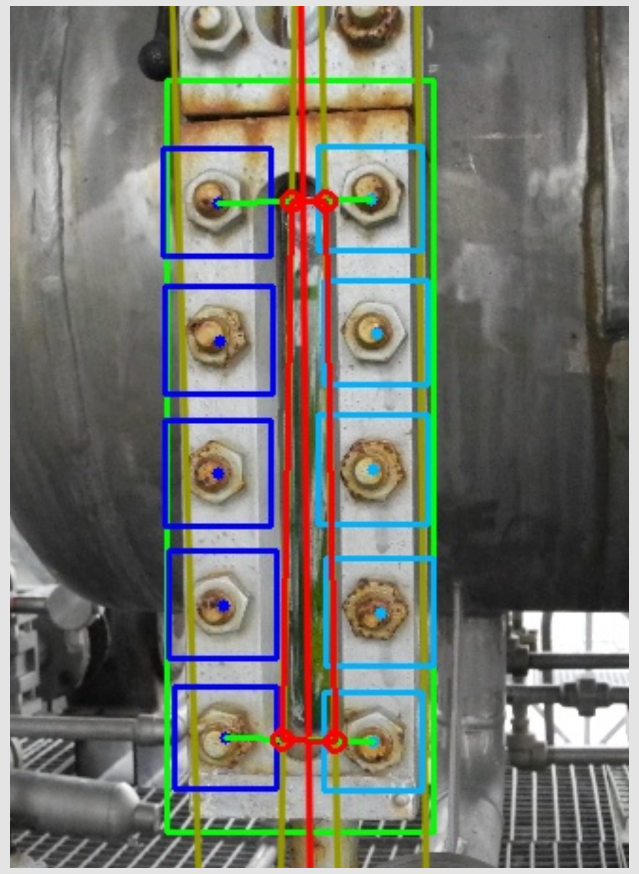
Autonomous Inspection of Industrial Sites

Visual inspection

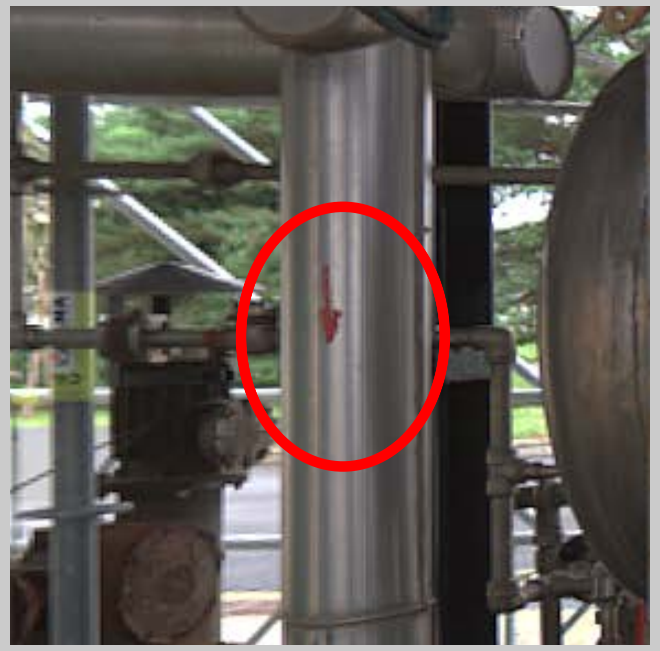




Pressure & Level gauges Valves

Zoom-camera

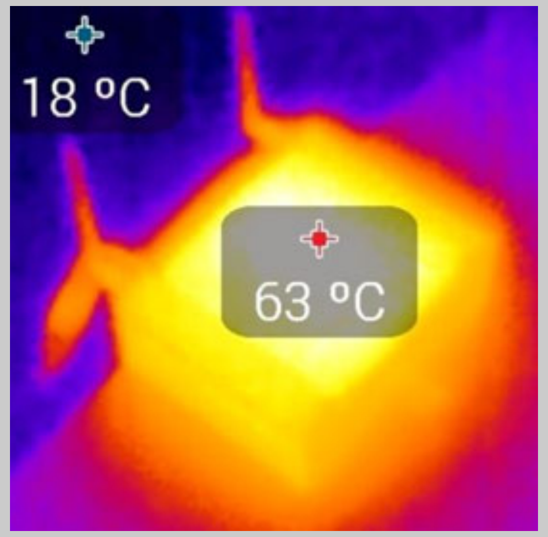


Thermal Inspection



Thermal points

Thermal-camera

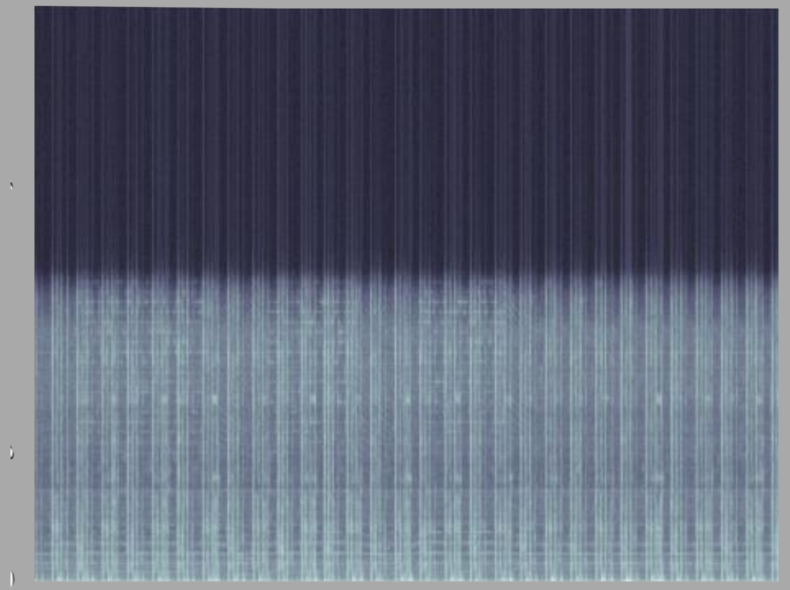


Auditive Inspection

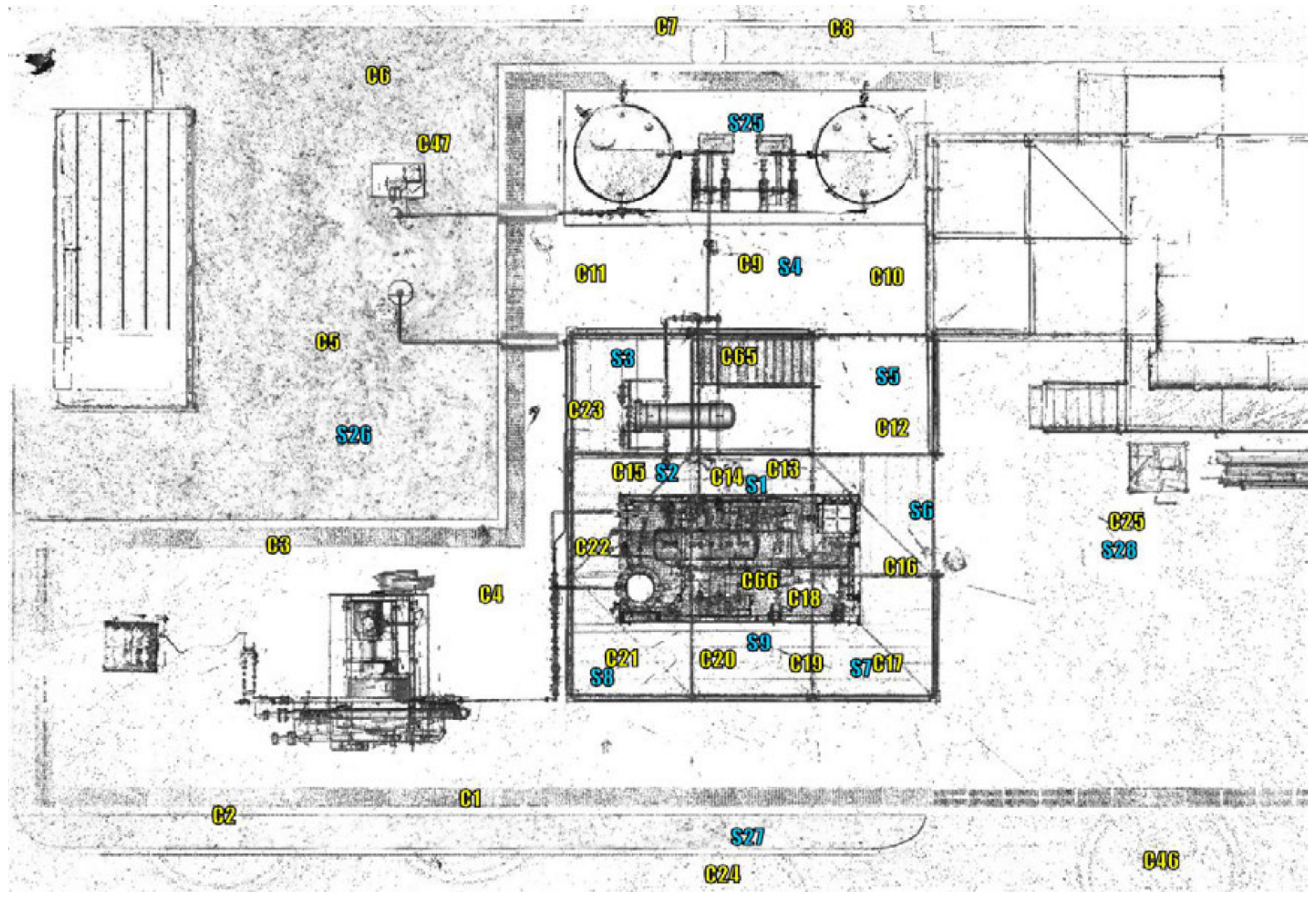


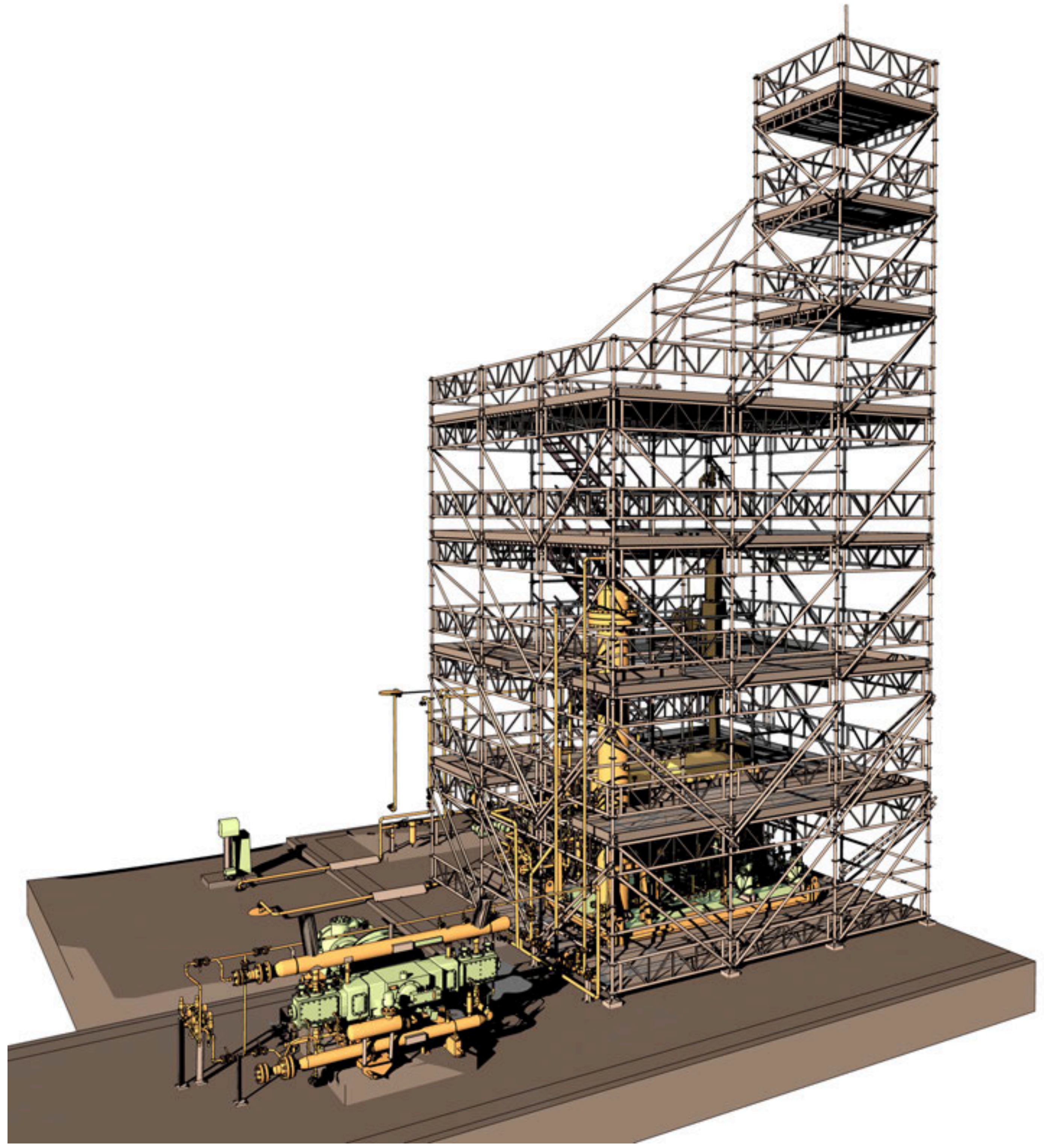
Pumps Gas leaks Platform alarm

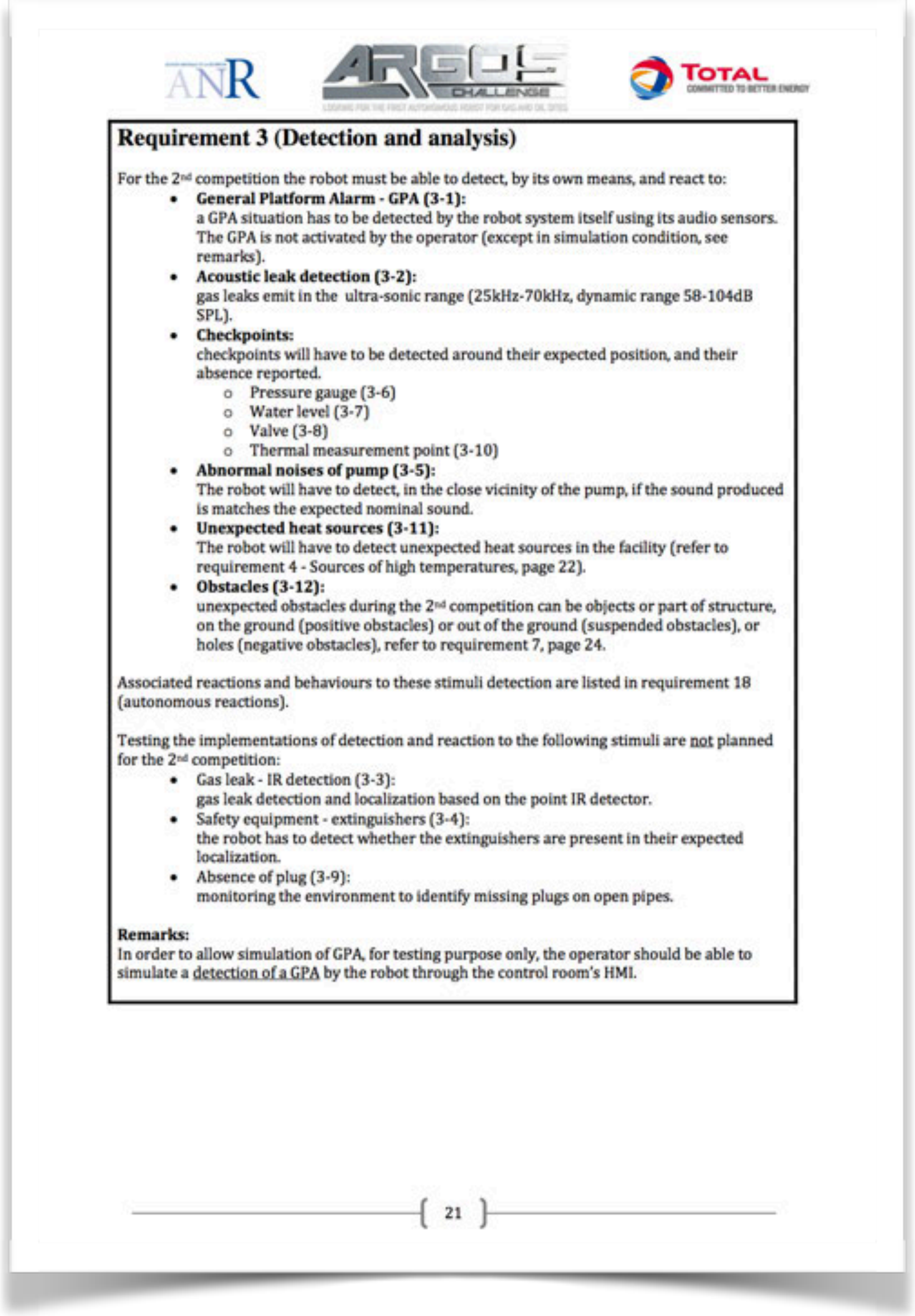
Microphones (audible and ultra-sonic)



Autonomous Inspection of Industrial Sites







Requirement 3 (Detection and analysis)

For the 2nd competition the robot must be able to detect, by its own means, and react to:

- **General Platform Alarm - GPA (3-1):**
  a GPA situation has to be detected by the robot system itself using its audio sensors. The GPA is not activated by the operator (except in simulation condition, see remarks).
- **Acoustic leak detection (3-2):**
  gas leaks emit in the ultra-sonic range (25kHz-70kHz, dynamic range 58-104dB SPL).
- **Checkpoints:**
  checkpoints will have to be detected around their expected position, and their absence reported.
  - Pressure gauge (3-6)
  - Water level (3-7)
  - Valve (3-8)
  - Thermal measurement point (3-10)
- **Abnormal noises of pump (3-5):**
  The robot will have to detect, in the close vicinity of the pump, if the sound produced is matches the expected nominal sound.
- **Unexpected heat sources (3-11):**
  The robot will have to detect unexpected heat sources in the facility (refer to requirement 4 - Sources of high temperatures, page 22).
- **Obstacles (3-12):**
  unexpected obstacles during the 2nd competition can be objects or part of structure, on the ground (positive obstacles) or out of the ground (suspended obstacles), or holes (negative obstacles), refer to requirement 7, page 24.

Associated reactions and behaviours to these stimuli detection are listed in requirement 18 (autonomous reactions).

Testing the implementations of detection and reaction to the following stimuli are not planned for the 2nd competition:

- Gas leak - IR detection (3-3):
  gas leak detection and localization based on the point IR detector.
- Safety equipment - extinguishers (3-4):
  the robot has to detect whether the extinguishers are present in their expected localization.
- Absence of plug (3-9):
  monitoring the environment to identify missing plugs on open pipes.

Remarks:
In order to allow simulation of GPA, for testing purpose only, the operator should be able to simulate a detection of a GPA by the robot through the control room's HMI.

21

ANYmal
A High-Performance Electrically Driven Quadruped

Outdoor operation
Water-proof & ruggedized design

Fast locomotion
Dynamic gaits at 1 m/s

Modular payload
10 kg payload

Safe and lightweight
30 kg and force-controlled

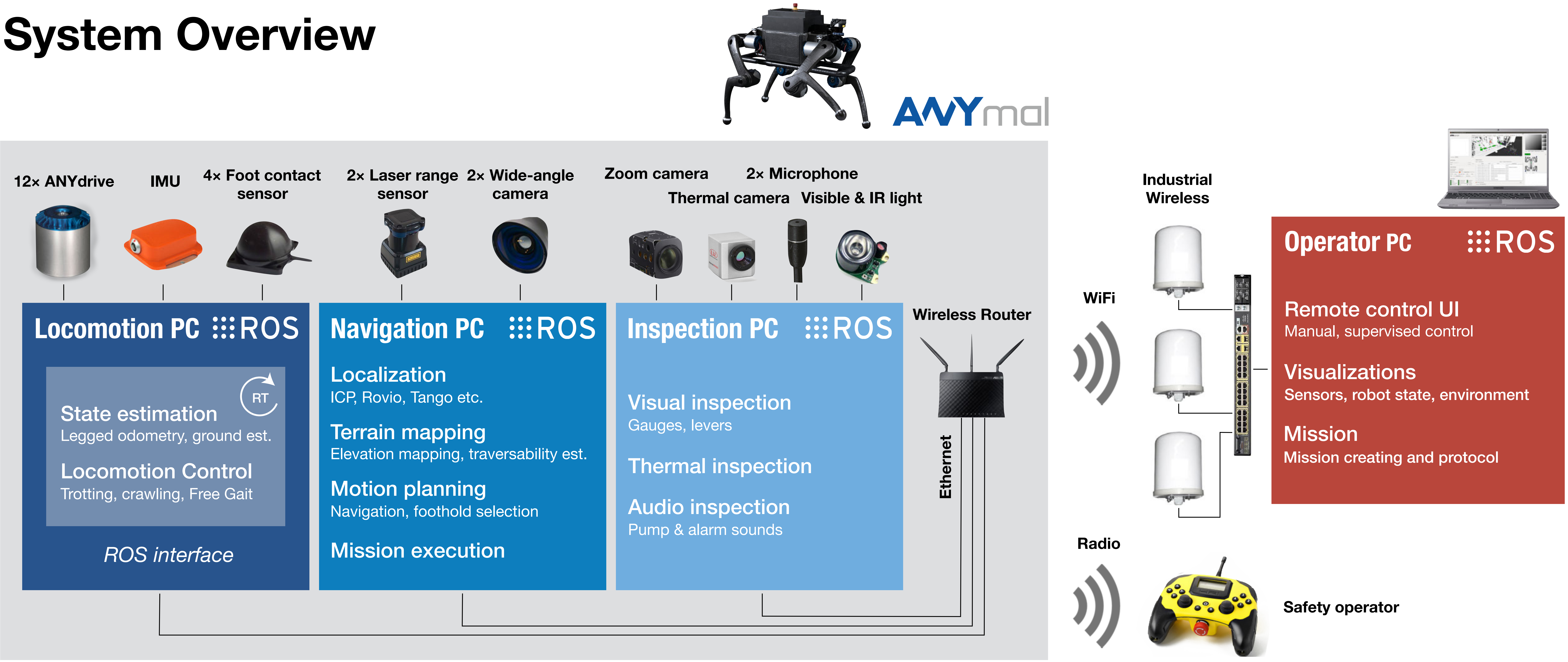
Extreme mobility
360° rotation of all joints

Full autonomy
Laser sensors for navigation

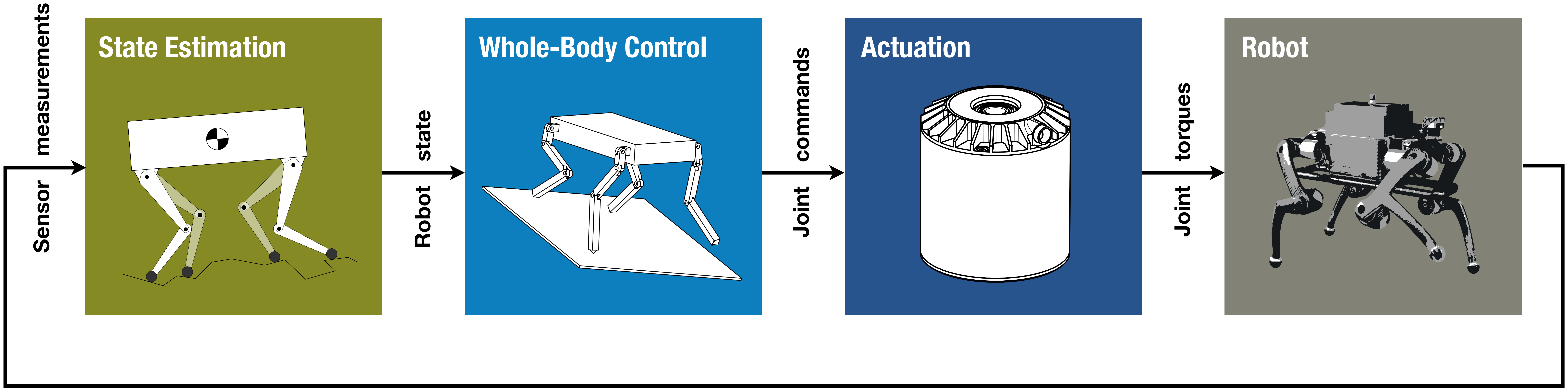
Long endurance
3 h operation

ANYbotics

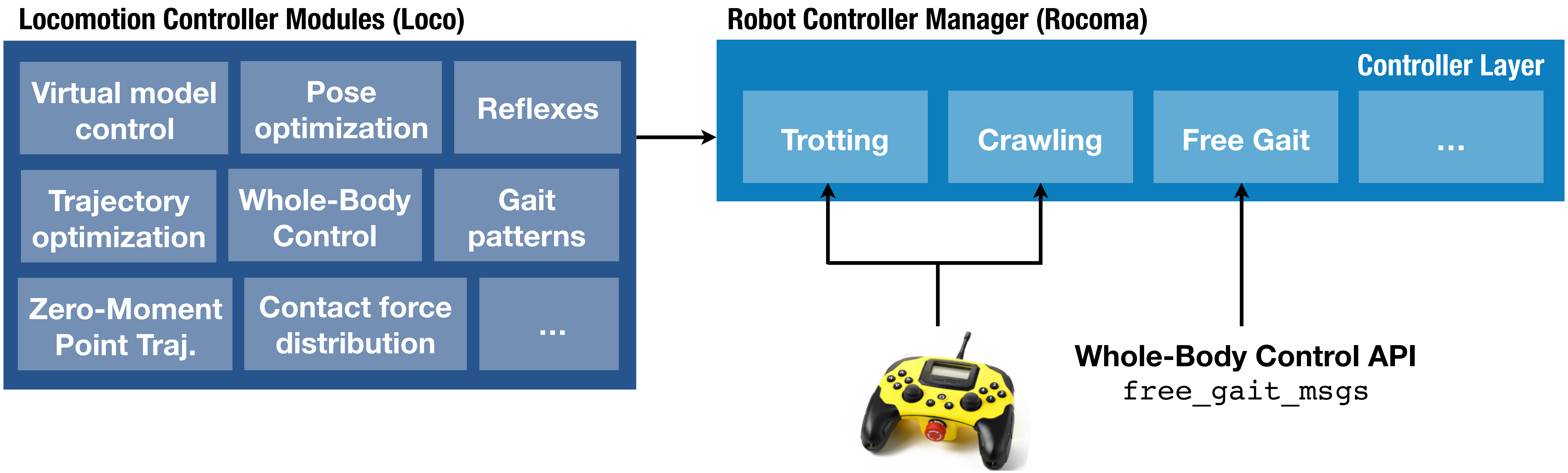
System Overview



Locomotion


Sensor measurements
State Estimation
Robot state
Whole-Body Control
Joint commands
Actuation
Joint torques
Robot

Locomotion
Whole-Body Control

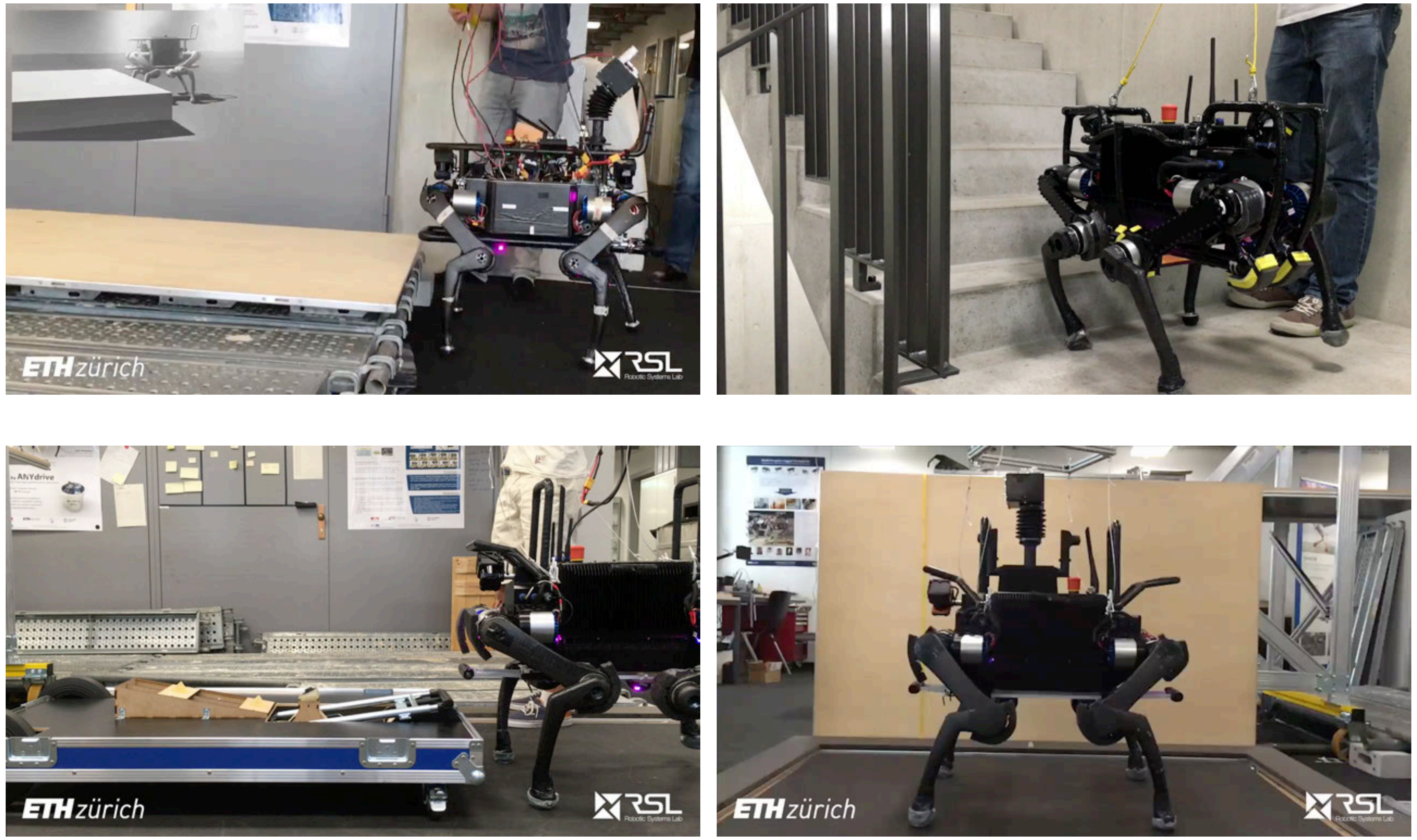


C. Gehring, S. Coros, M. Hutter, D. Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. A. Hoepflinger, and R. Siegwart, "**Practice Makes Perfect: An Optimization-Based Approach to Controlling Agile Motions for a Quadruped Robot.**", in IEEE Robotics & Automation Magazine, 2016.

C. Dario Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, M. Hutter, "**Emerging Terrain Adaptation from Hierarchical Whole Body Control**,"
in IEEE Internal Conference on Humanoid Robots (Humanoids), 2016.

Open Source
github.com/leggedrobotics/free_gait

Locomotion
Free Gait – An Architecture for the Versatile Control of Legged Robots



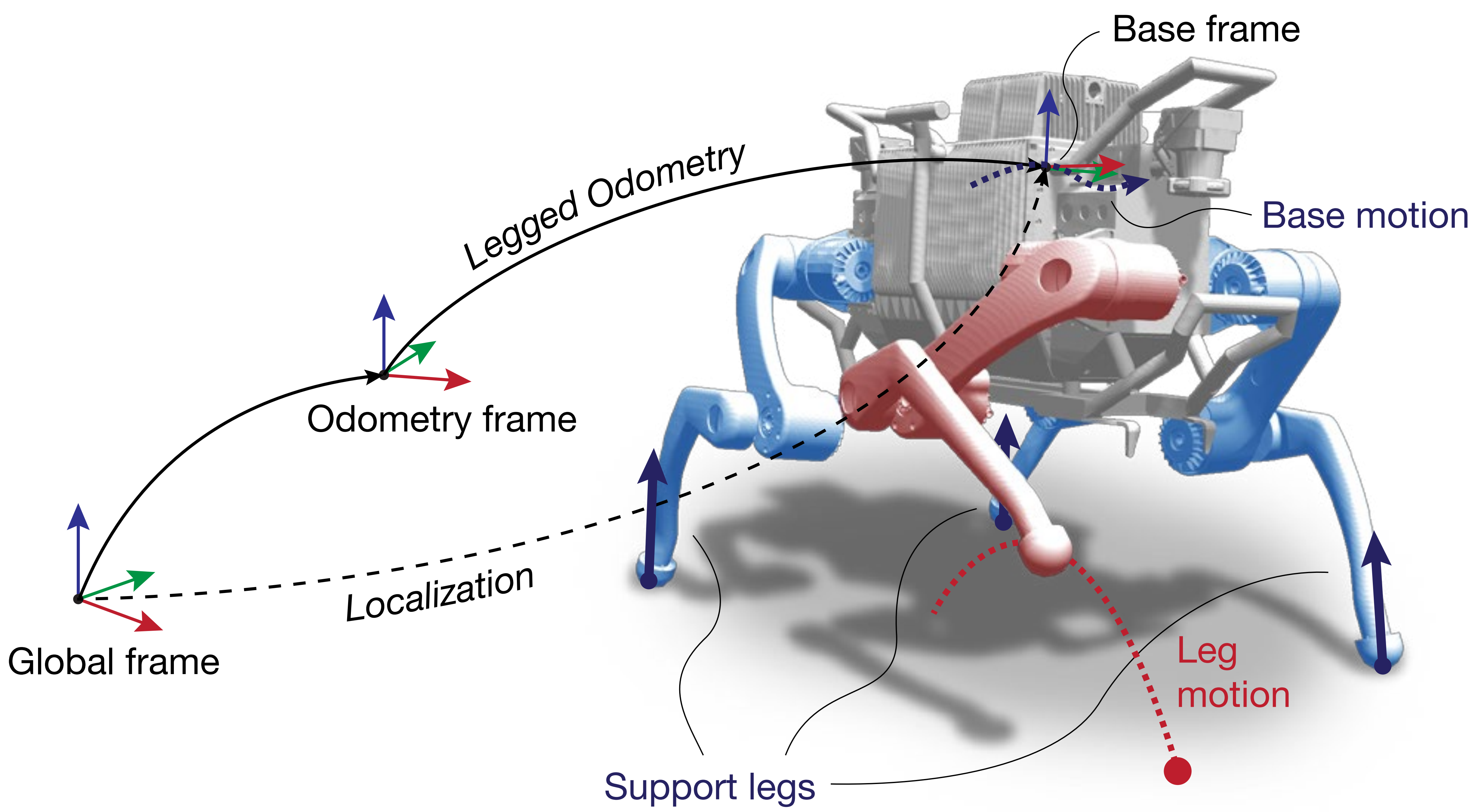
- Abstraction Layer for Whole-Body Motions (Free Gait API)

P. Fankhauser, D. Bellicoso, C. Gehring, R. Dubé, A. Gawel, and M. Hutter, **"Free Gait – An Architecture for the Versatile Control of Legged Robots,"** in IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2016.

Open Source
github.com/leggedrobotics/free_gait

Locomotion

Free Gait – An Architecture for the Versatile Control of Legged Robots



- Abstraction Layer for Whole-Body Motions (Free Gait API)
- Robust motion execution in task space

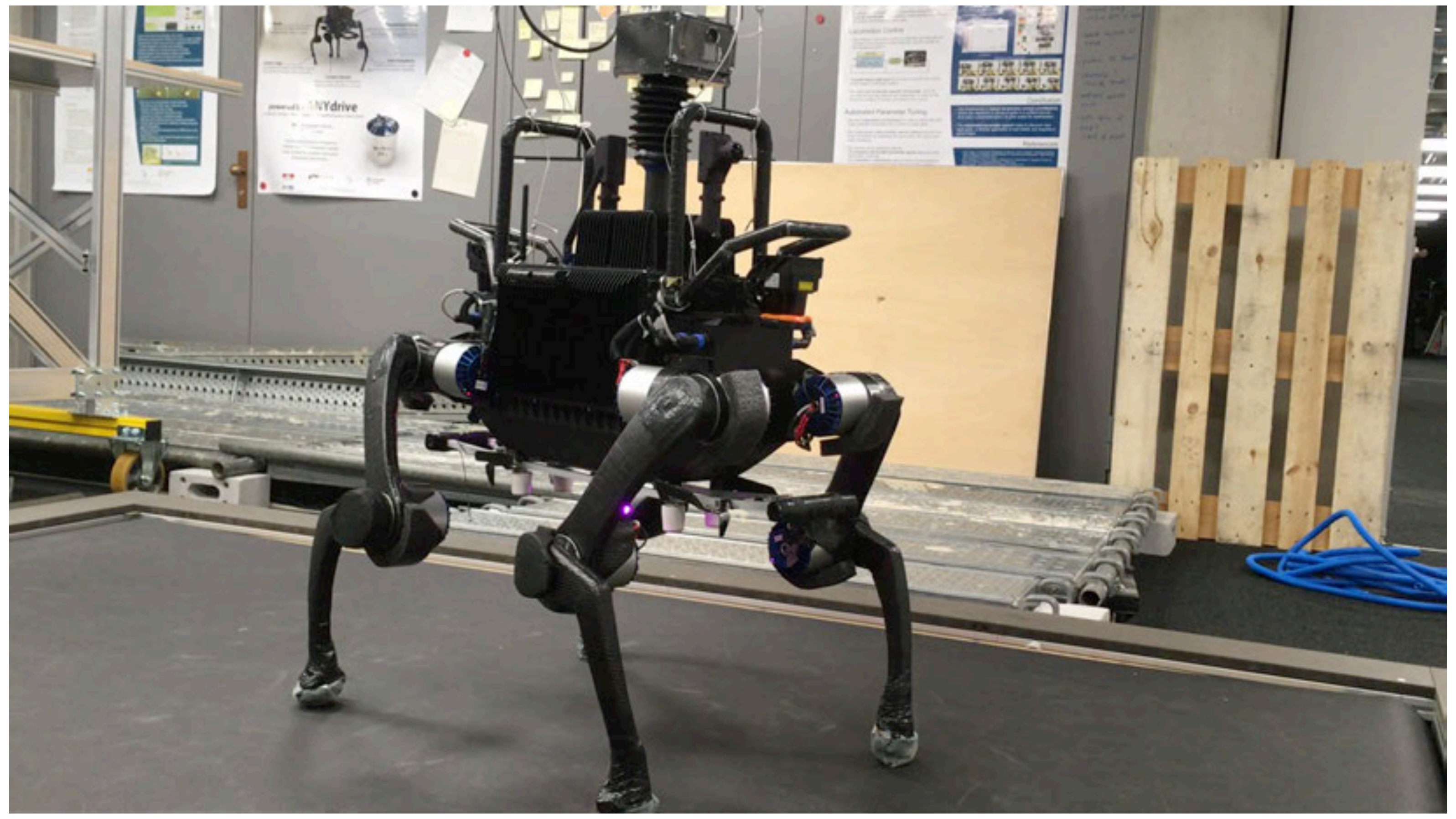
P. Fankhauser, D. Bellicoso, C. Gehring, R. Dubé, A. Gawel, and M. Hutter, **"Free Gait – An Architecture for the Versatile Control of Legged Robots,"** in IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2016.

Open Source
github.com/leggedrobotics/free_gait

Locomotion

Free Gait – An Architecture for the Versatile Control of Legged Robots

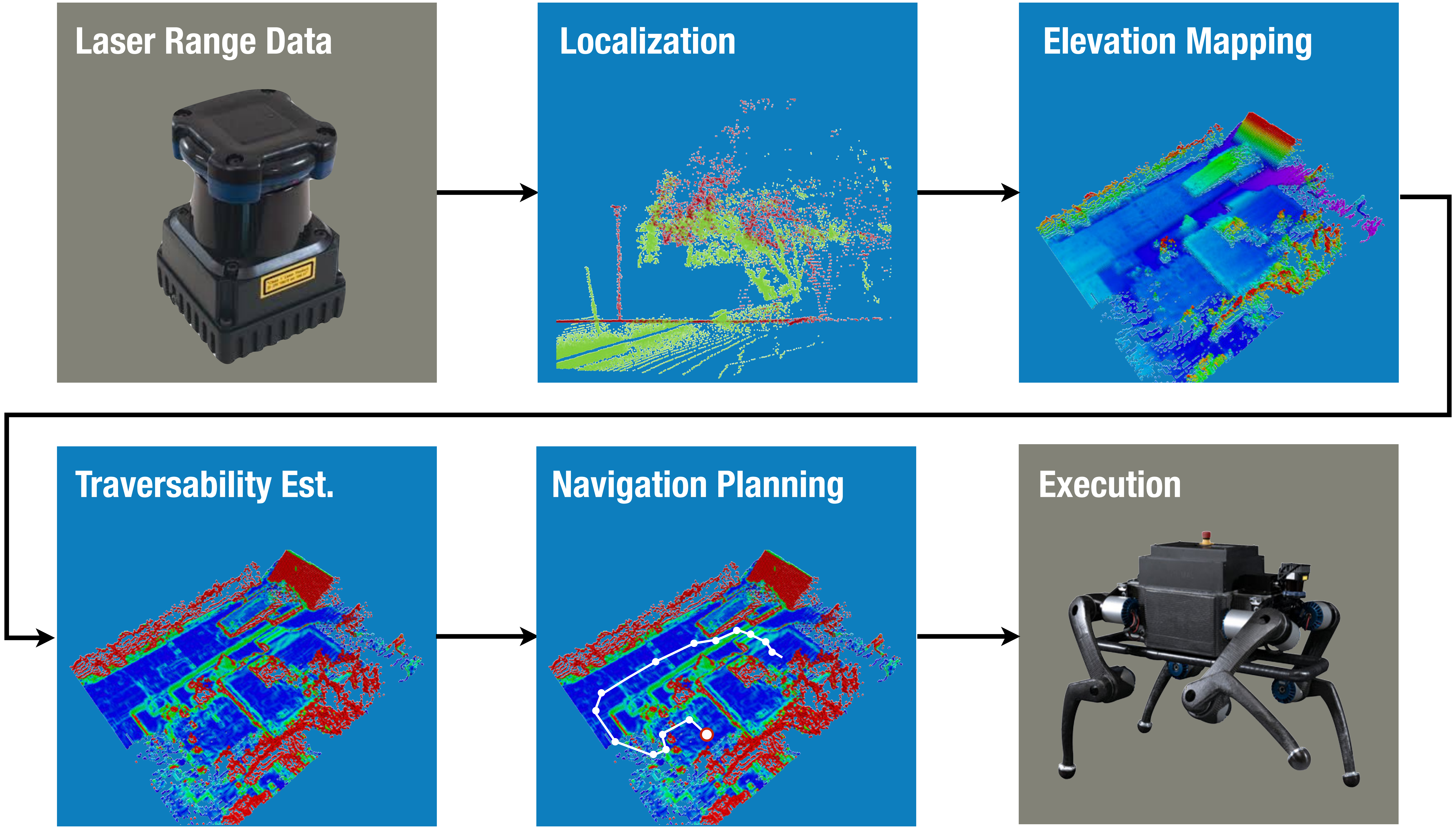
```yaml
steps:
 - step:
   - base_auto:
 - step:
   - end_effector_target:
      name: RF_LEG
      ignore_contact: true
      target_position:
       frame: footprint
       position: [0.39, -0.24, 0.20]
 - step:
   - base_auto:
      height: 0.38
      ignore_timing_of_leg_motion: true
   - end_effector_target: &foot
      name: RF_LEG
      ignore_contact: true
      ignore_for_pose_adaptation: true
      target_position:
       frame: footprint
       position: [0.39, -0.24, 0.20]
 - step:
   - base_auto:
      height: 0.45
      ignore_timing_of_leg_motion: true
   - end_effector_target: *foot
 - step:
   - footstep:
      name: RF_LEG
      profile_type: straight
      target:
       frame: footprint
       position: [0.32, -0.24, 0.0]
 - step:
   - base_auto:
```



- Abstraction Layer for Whole-Body Motions (Free Gait API)
- Robust motion execution in task space
- Implemented as ROS Action (with frameworks for YAML, Python, C++)

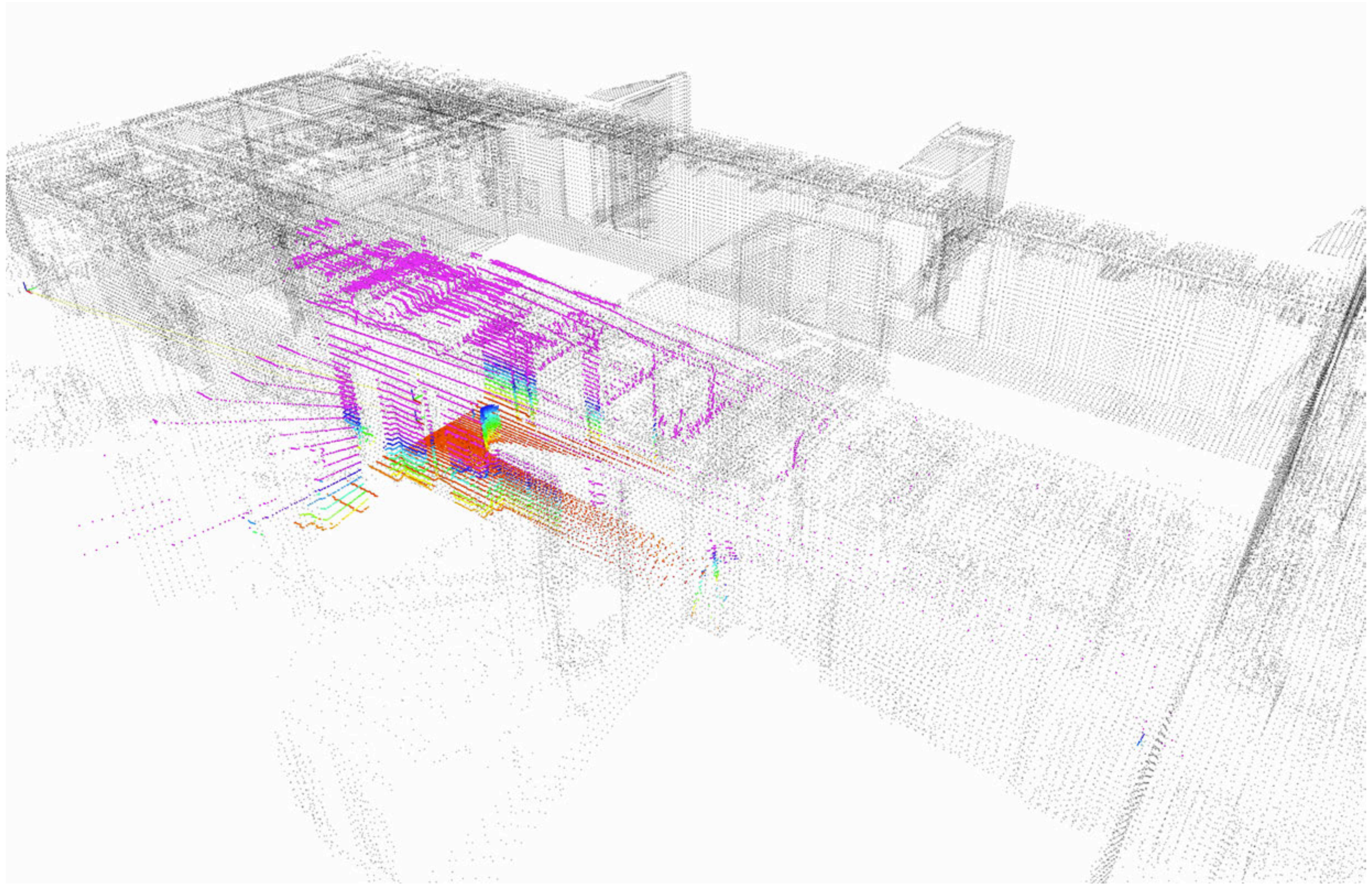
P. Fankhauser, D. Bellicoso, C. Gehring, R. Dubé, A. Gawel, and M. Hutter, **“Free Gait – An Architecture for the Versatile Control of Legged Robots,”** in IEEE-RAS International Conference on Humanoid Robots (Humanoids), 2016.

Navigation


Laser Range Data
Localization
Elevation Mapping
Traversability Est.
Navigation Planning
Execution

Navigation
Laser-Based Localization (Iterative Closest Point (ICP))

Open Source
github.com/ethz-asl/ethzasl_icp_mapping

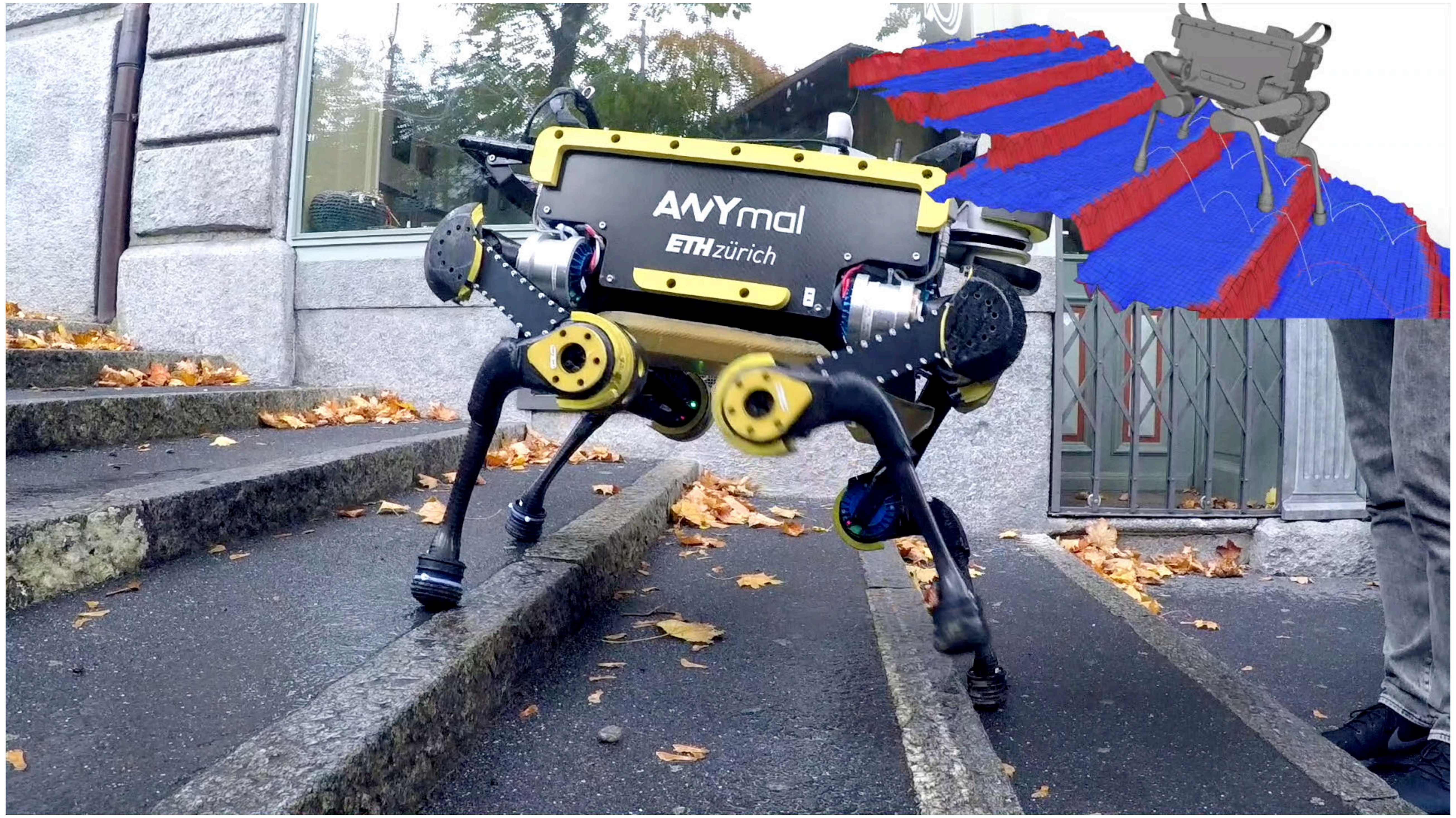


- Point cloud registration for localization in reference map
- Full rotation of LiDAR is aggregated for point cloud
- Use of existing maps or online mapping

Pomerleau, F., Colas, F., Siegwart, R., Magnenat, S., "**Comparing ICP variants on real-world data sets**", in Autonomous Robots, 2013.

Navigation
Elevation Mapping – Dense Terrain Mapping

Open Source
github.com/ethz-asl/elevation_mapping



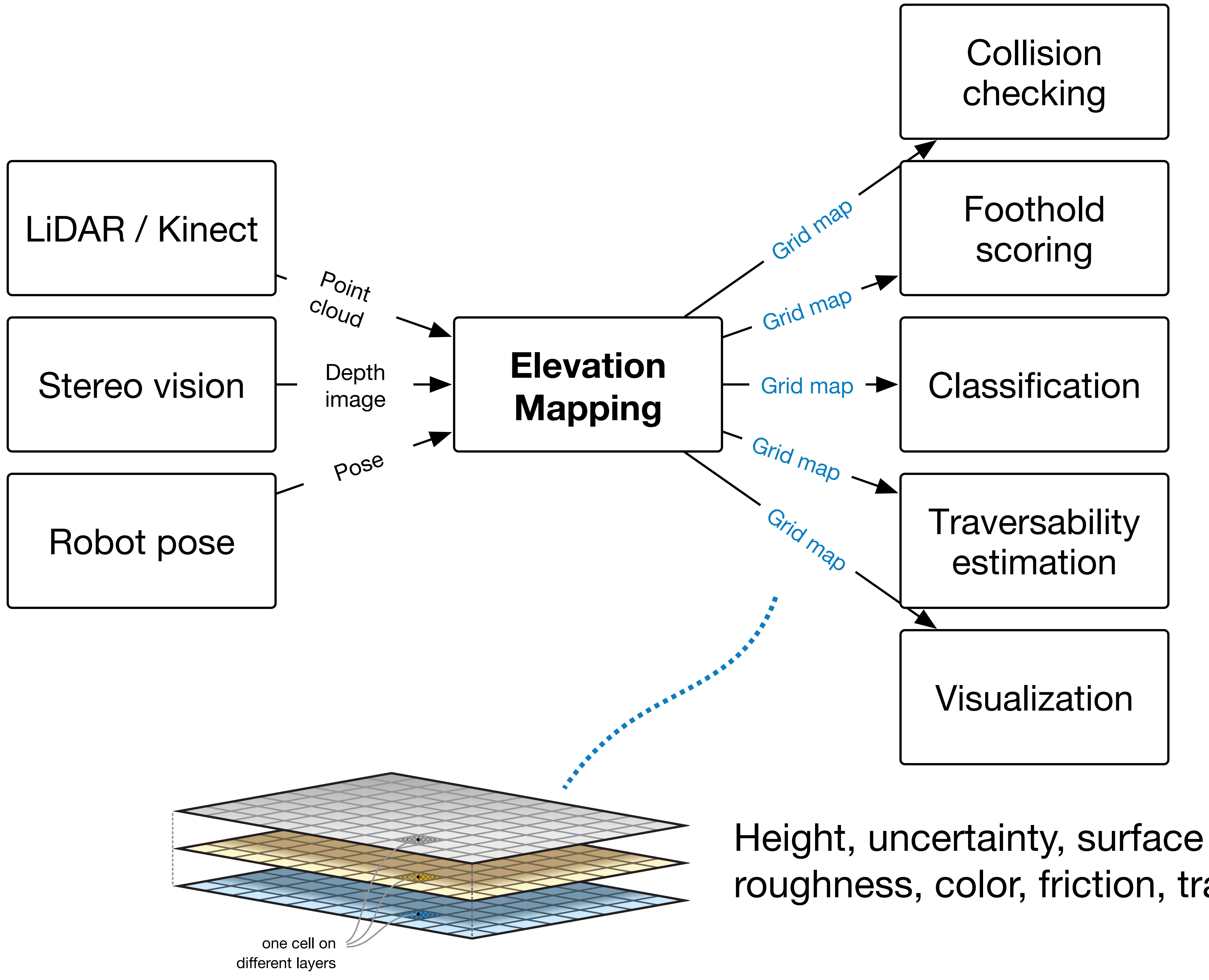
- Probabilistic fusion of range measurements and pose estimation
- Explicitly handles drift of state estimation (robot-centric)
- Input data from laser, Kinect, stereo cameras, Velodyne etc.

P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, R. Siegwart **"Robot-Centric Elevation Mapping with Uncertainty Estimates,"** in International Conference on Climbing and Walking Robots (CLAWAR), 2014.

P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "**Robust Rough-Terrain Locomotion with a Quadrupedal Robot**," in IEEE International Conference on Robotics and Automation (ICRA), 2018.

Navigation
Grid Map – Universal Multi-Layer Grid Map Library

Open Source
github.com/ethz-asl/grid_map



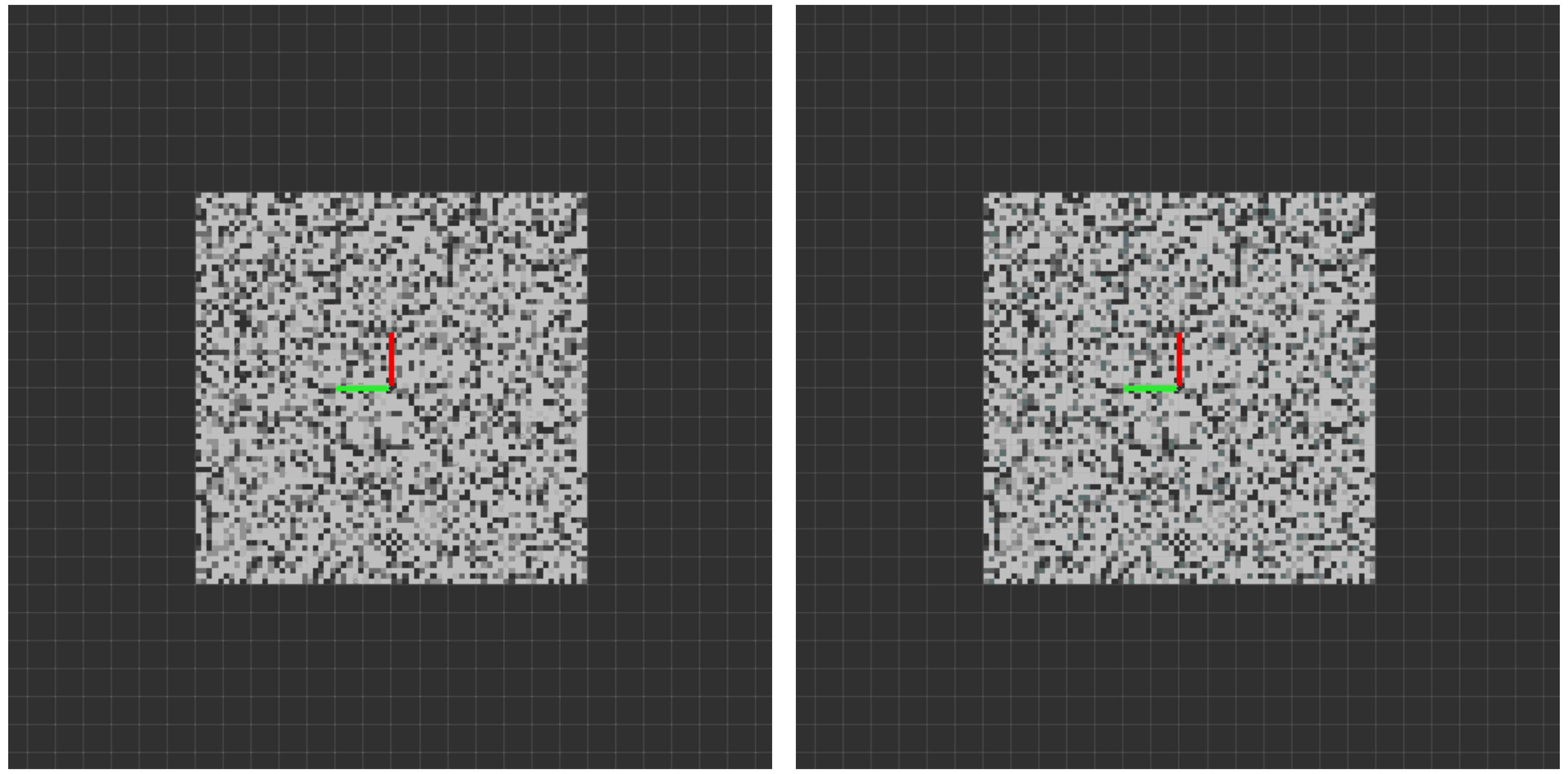
Height, uncertainty, surface normal, roughness, color, friction, traversability …

P. Fankhauser and M. Hutter, **“A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation,”** in Robot Operating System (ROS) - The Complete Reference, Springer, 2015.

Navigation
Grid Map – Universal Multi-Layer Grid Map Library

Open Source
github.com/ethz-asl/grid_map

- 2D circular buffer data structure
  - ➡ Efficient map repositioning



`setPosition(…)`   `move(…)`

P. Fankhauser and M. Hutter, **"A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation,"** in Robot Operating System (ROS) - The Complete Reference, Springer, 2015.

Navigation
Grid Map – Universal Multi-Layer Grid Map Library

Open Source
github.com/ethz-asl/grid_map

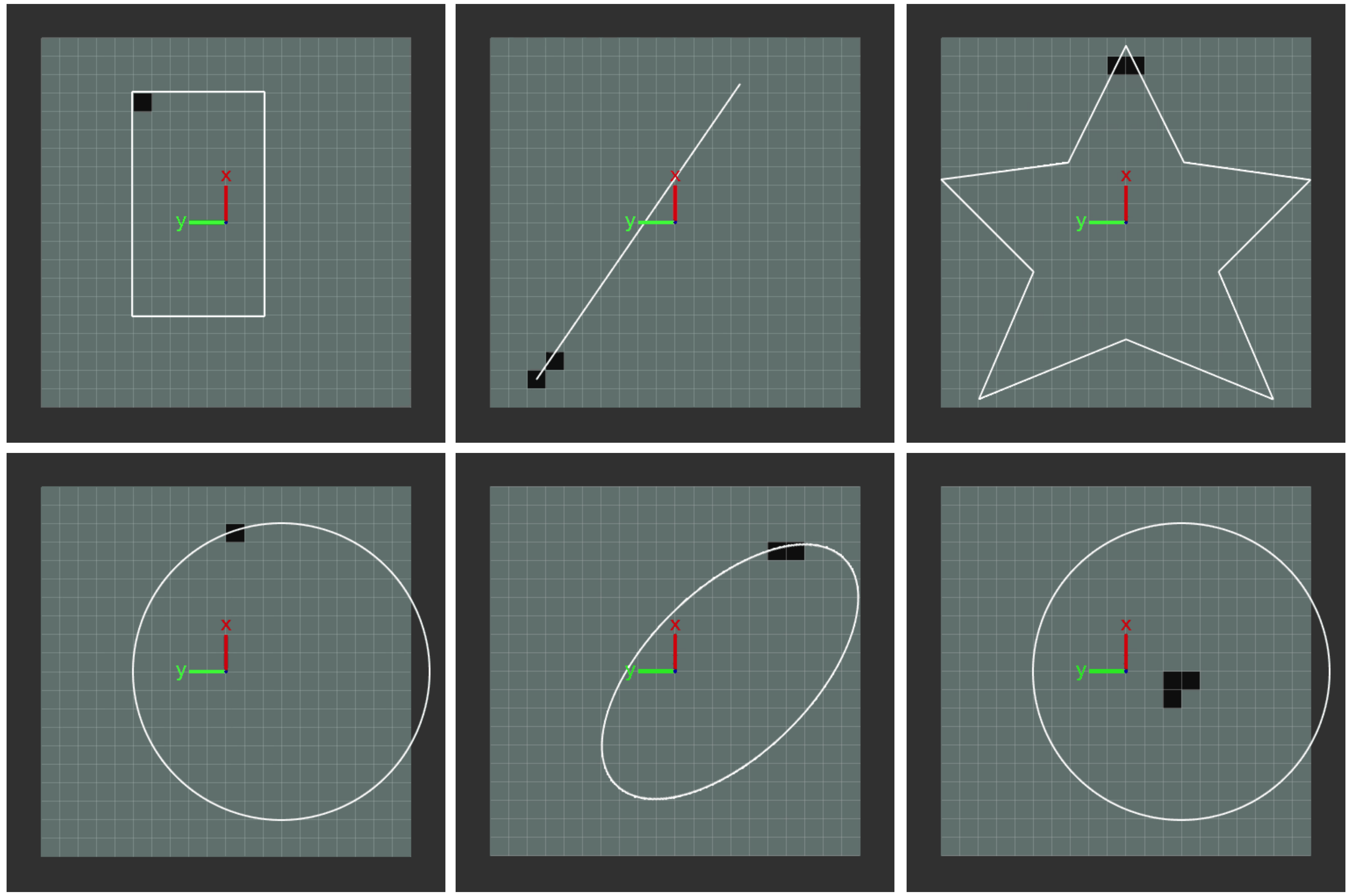
```
double rmse =
       sqrt(map["error"].array().pow(2).sum() / nCells);
```

- 2D circular buffer data structure
  - ➡ Efficient map repositioning
- Based on Eigen (C++)
  - ➡ Versatile and efficient data manipulation

P. Fankhauser and M. Hutter, **"A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation,"** in Robot Operating System (ROS) - The Complete Reference, Springer, 2015.

Navigation
Grid Map – Universal Multi-Layer Grid Map Library

Open Source
github.com/ethz-asl/grid_map

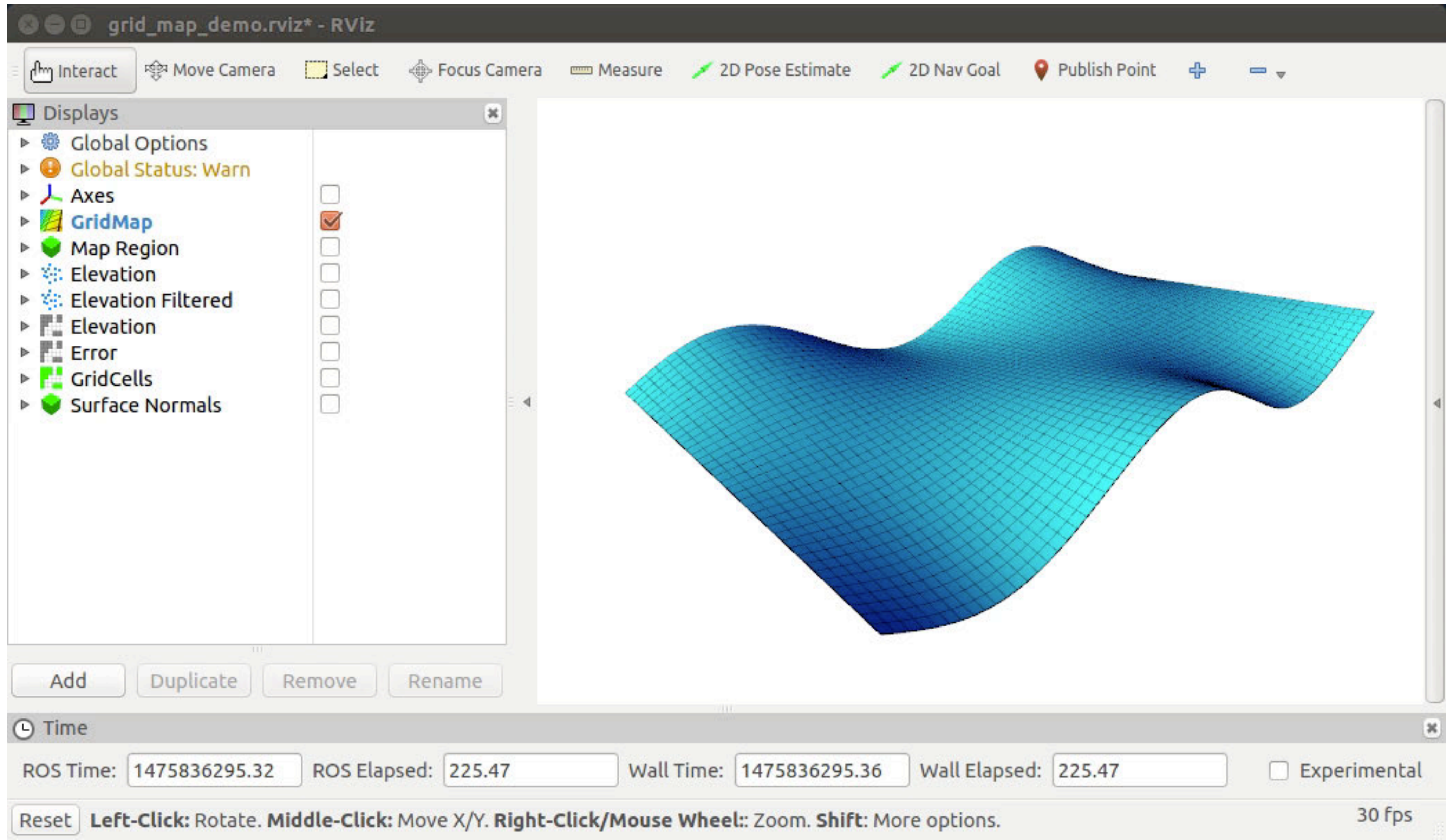


- 2D circular buffer data structure
  - ➡ Efficient map repositioning
- Based on Eigen (C++)
  - ➡ Versatile and efficient data manipulation
- Convenience functions
  - ➡ Iterators, math tools, etc.

P. Fankhauser and M. Hutter, **“A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation,”** in Robot Operating System (ROS) - The Complete Reference, Springer, 2015.

Navigation
Grid Map – Universal Multi-Layer Grid Map Library

Open Source
github.com/ethz-asl/grid_map

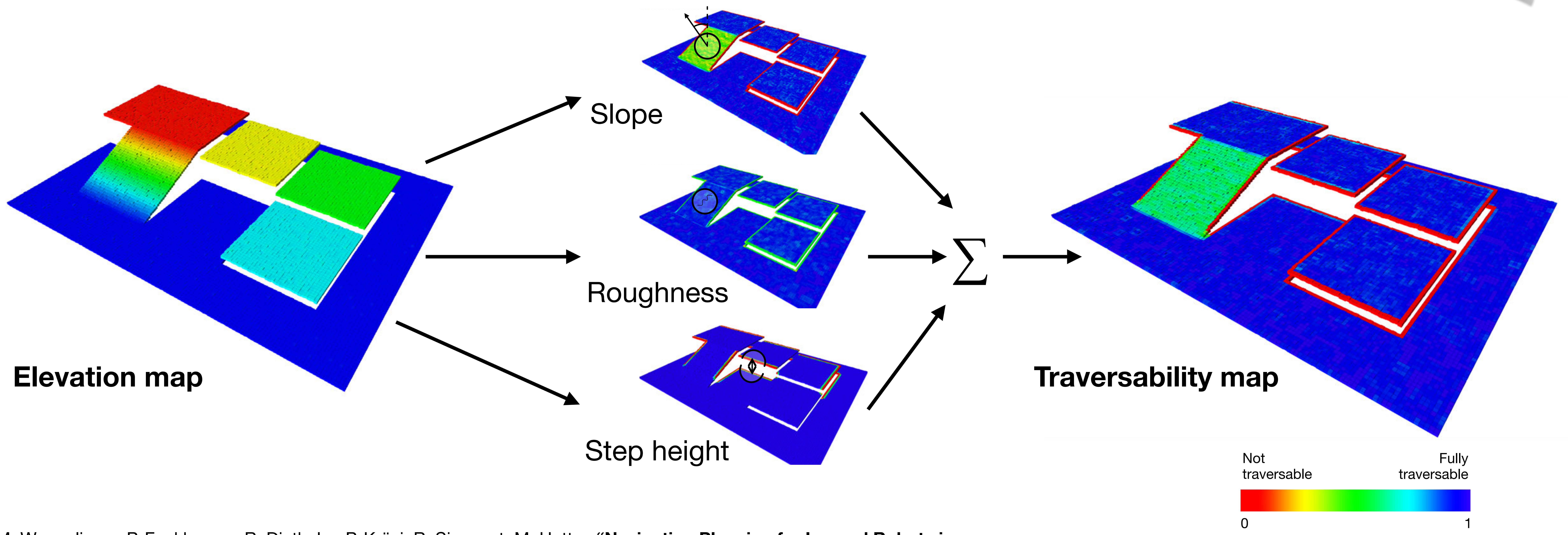


- 2D circular buffer data structure
  - ➡ Efficient map repositioning
- Based on Eigen (C++)
  - ➡ Versatile and efficient data manipulation
- Convenience functions
  - ➡ Iterators, math tools, etc.
- ROS & OpenCV interfaces
  - ➡ Conversion from/to images, point clouds, occupancy grids, grid cells

P. Fankhauser and M. Hutter, **"A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation,"** in Robot Operating System (ROS) - The Complete Reference, Springer, 2015.

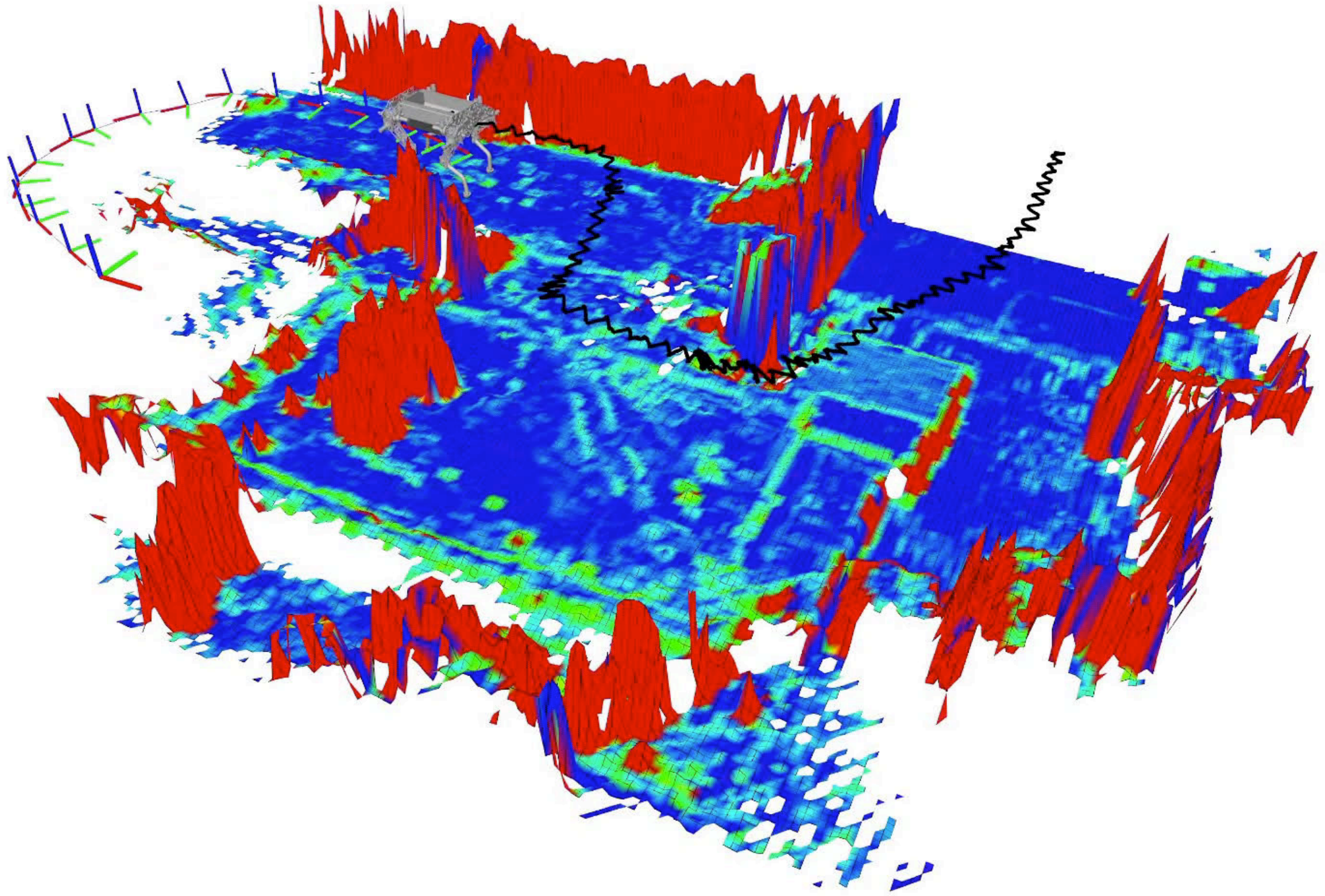
Open Source
github.com/ethz-asl/traversability_estimation

Navigation
Traversability Estimation



M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, M. Hutter, **"Navigation Planning for Legged Robots in Challenging Terrain,"** in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.

Navigation
Navigation Planning

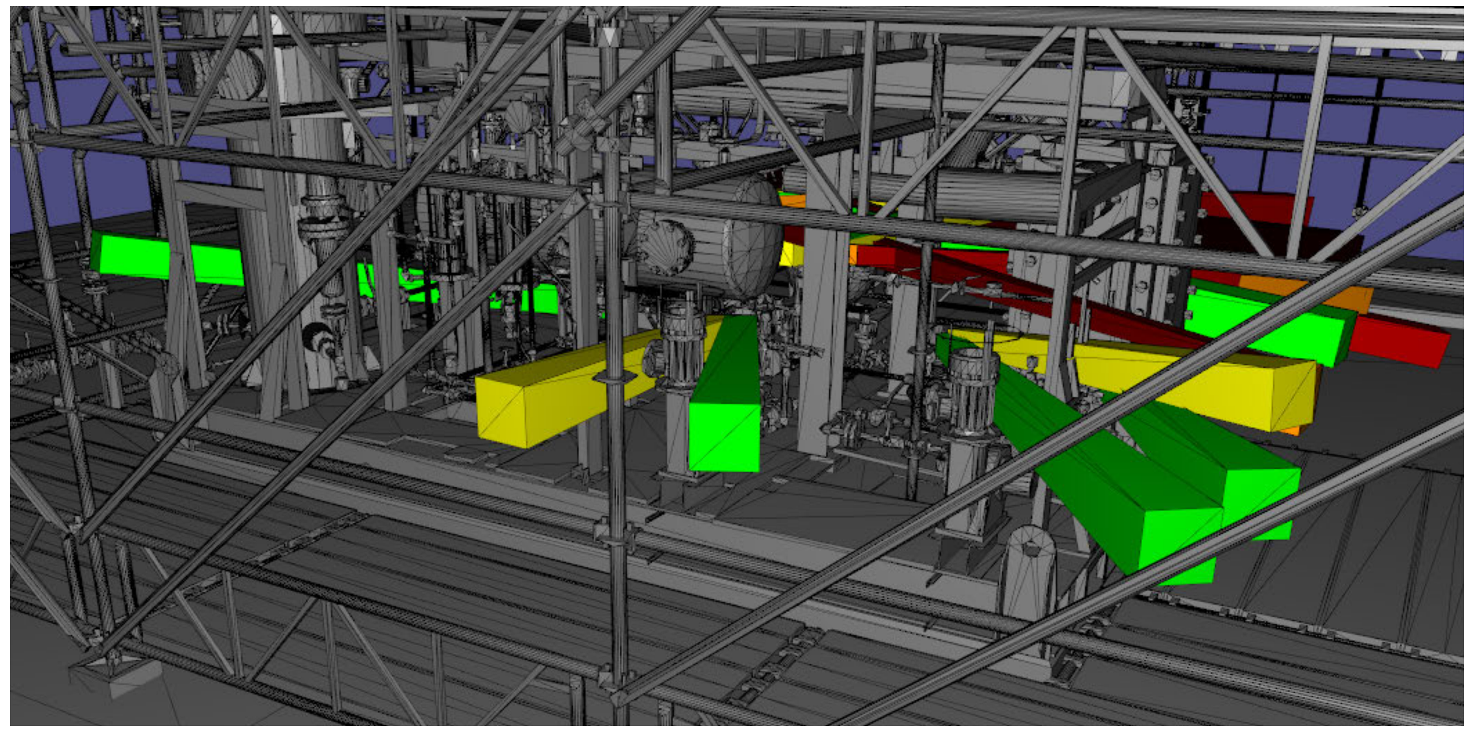


- Online navigation planning based on RRT* (OMPL)
- Works with and without initial map
- Continuous for changing environments

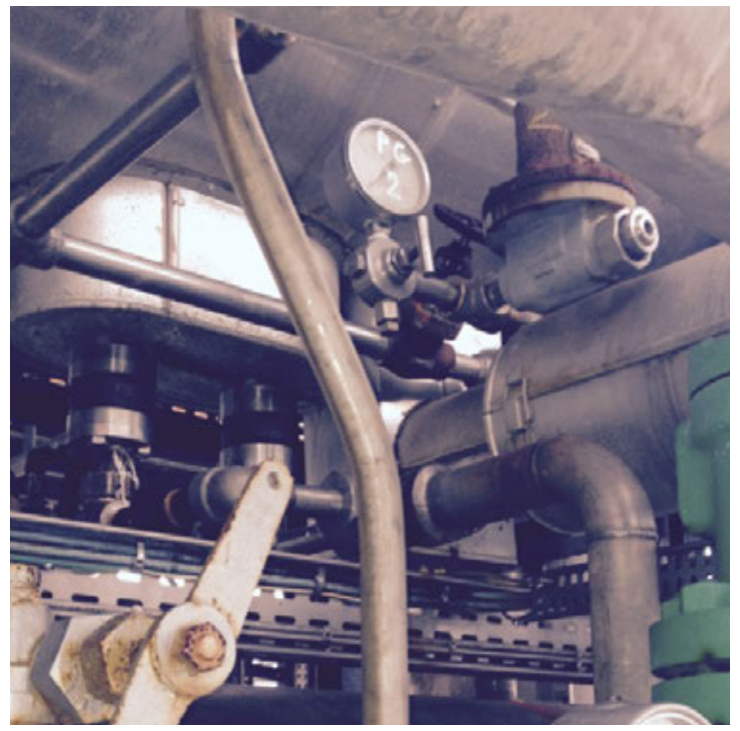
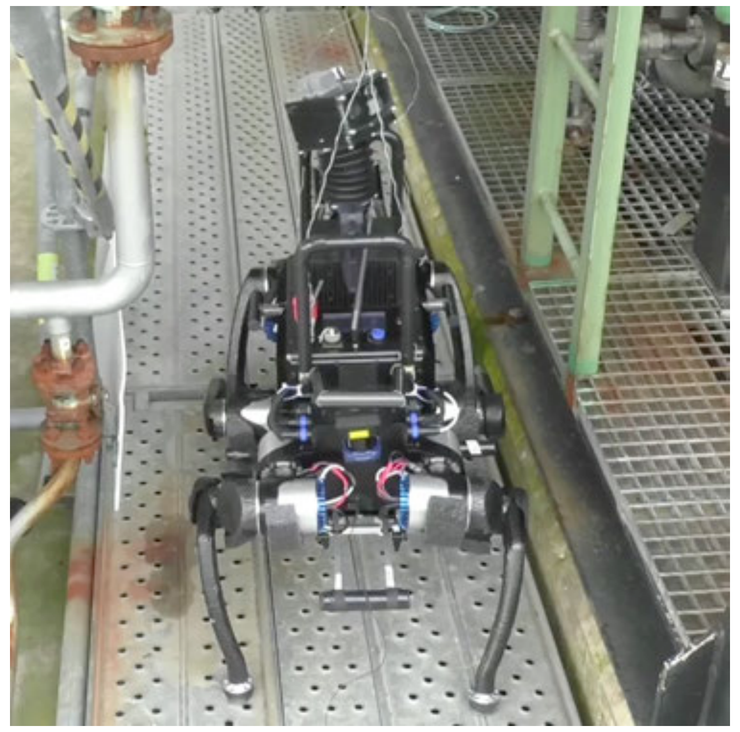
M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, M. Hutter, **"Navigation Planning for Legged Robots in Challenging Terrain,"** in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.

Inspection
Visual Inspection of Pressure Gauges

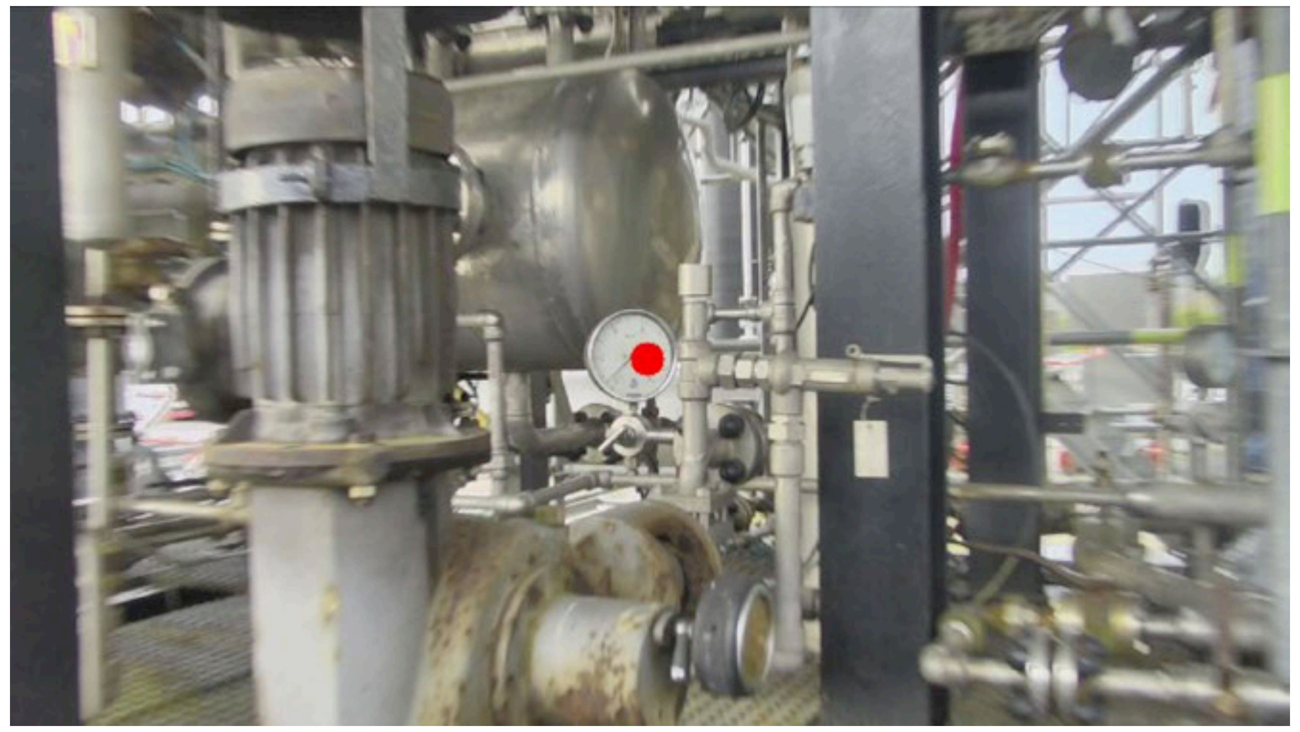
A Automatic view point generation



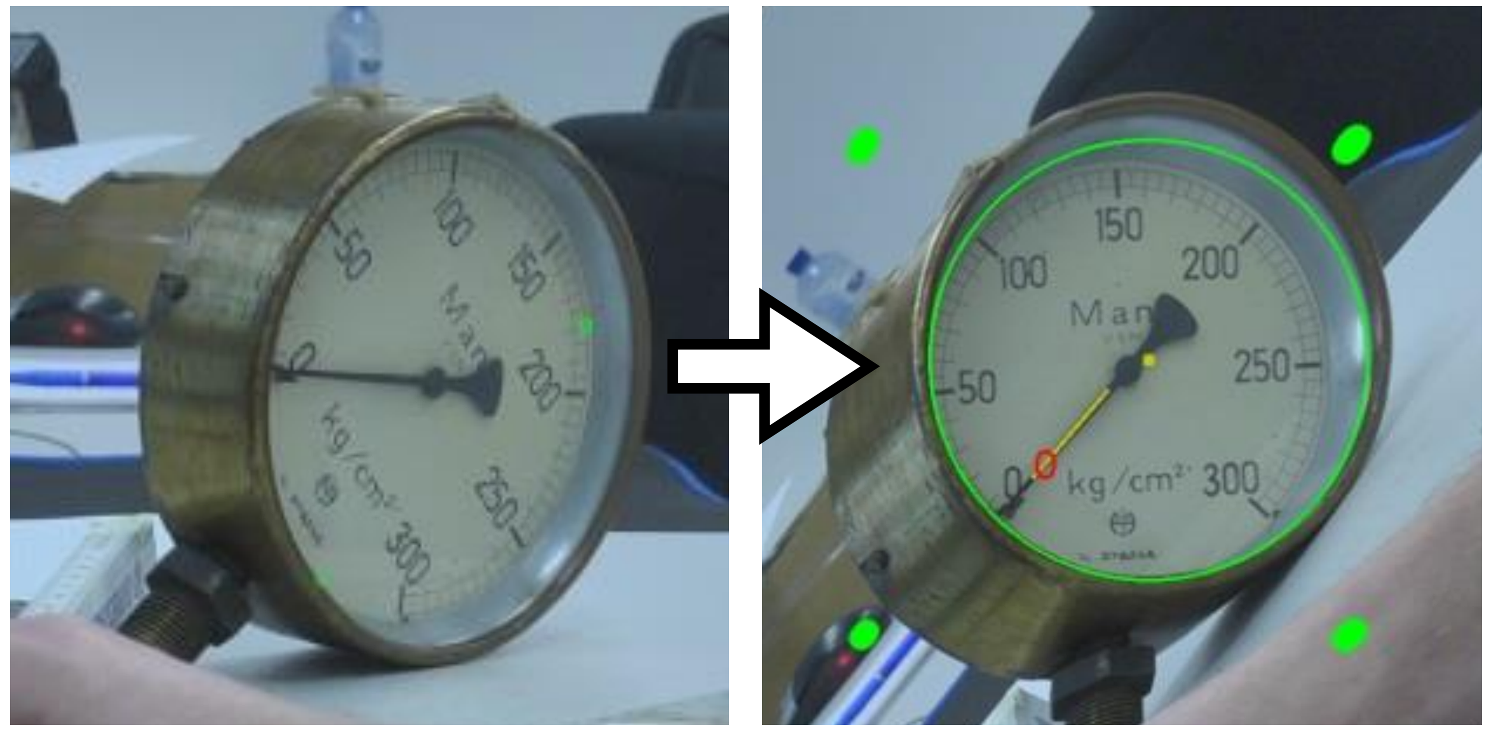
B Whole-body camera positioning

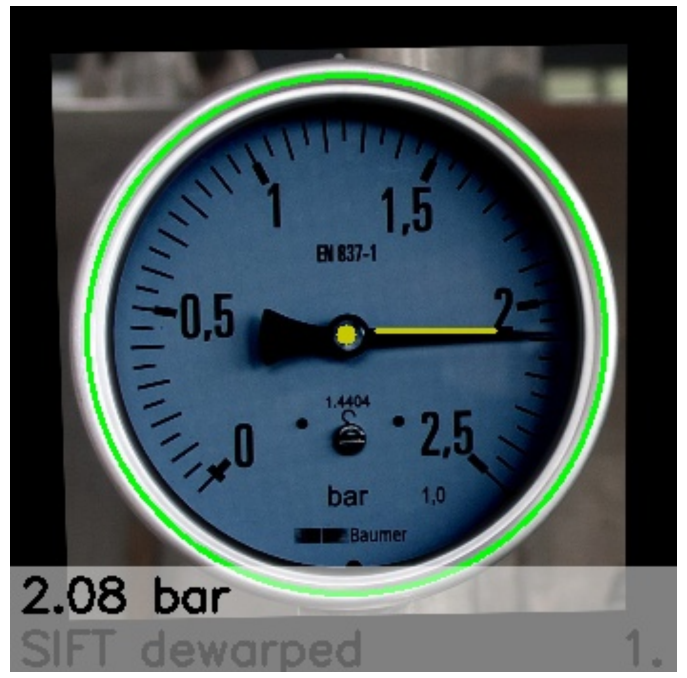
C Visual servoing



D Image de-warping



E Indicator reading



➔ Reading ok


➔ Reading unsuccessful, try alternative position or report as unknown

S. Bachmann, “**Visual Inspection of Manometers and Valve Levers**”, Master’s Thesis, ETH Zurich, 2015.

User Interface

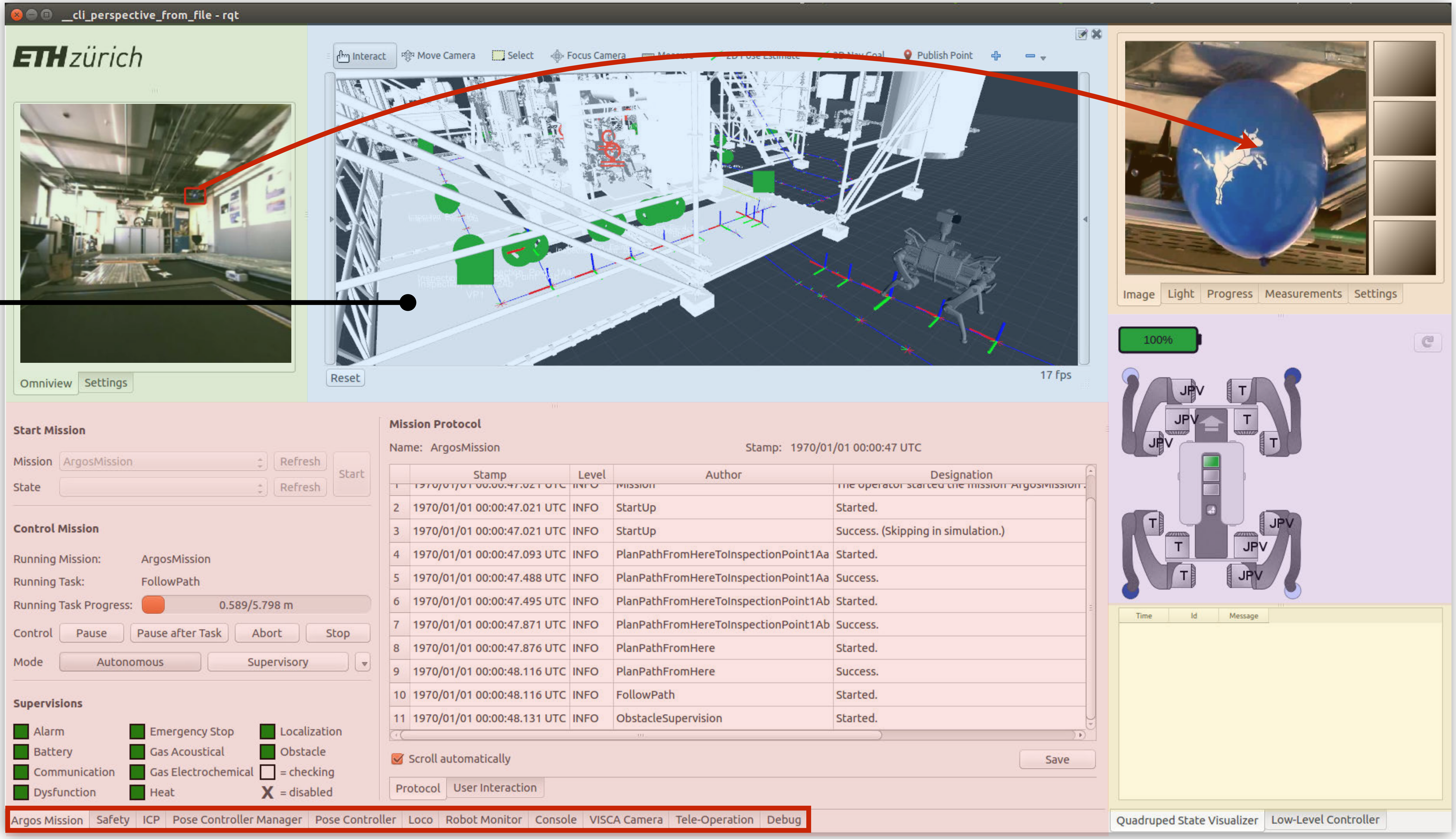
Interface for remote control, semi-, and full autonomous operation.

Situational camera

3D view (RViz)

Mission control & protocol

Other modules

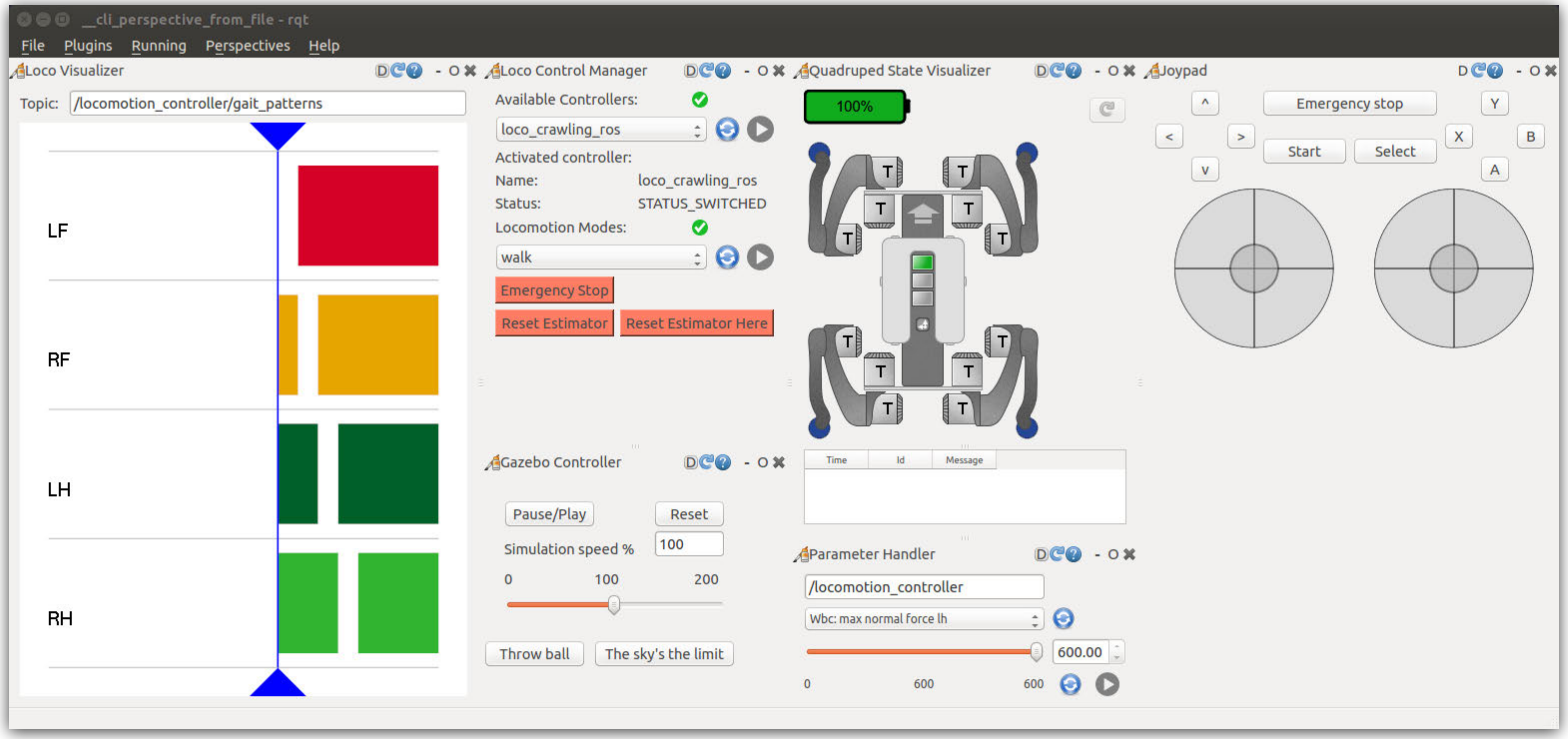


Inspection cameras

Robot actuators & sensors

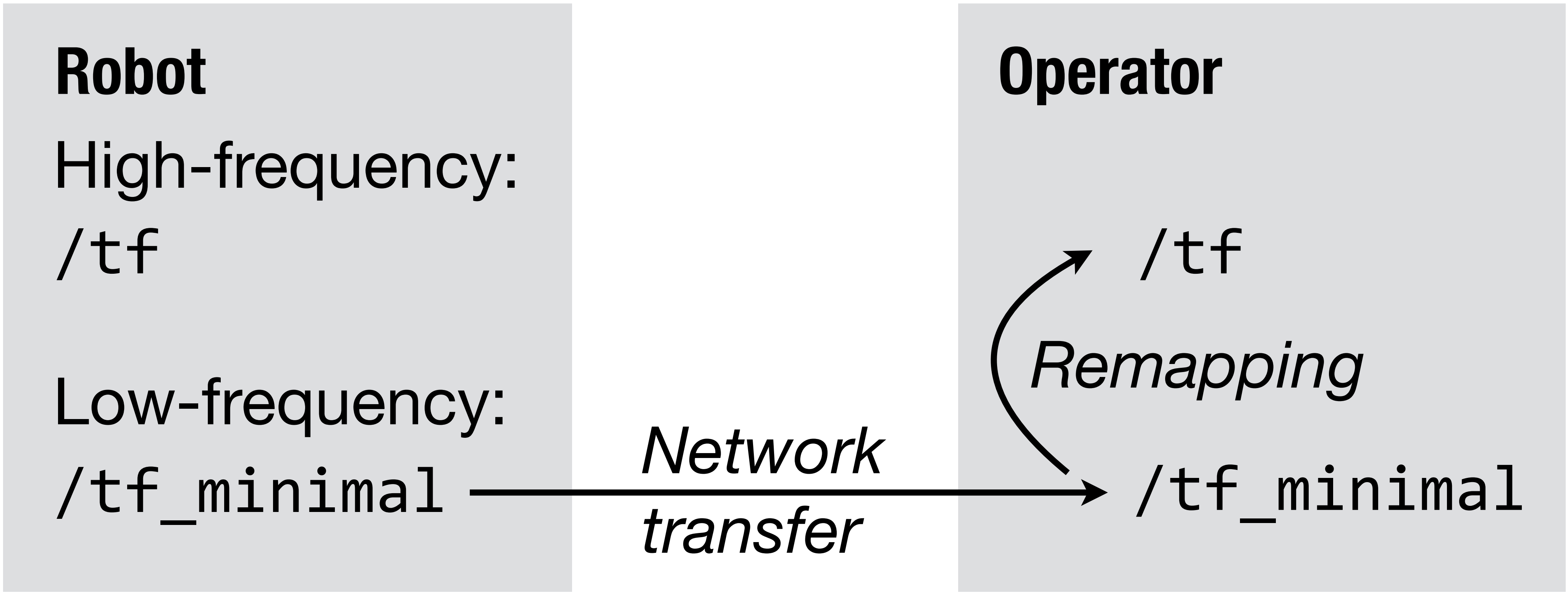
Error protocol

User Interface

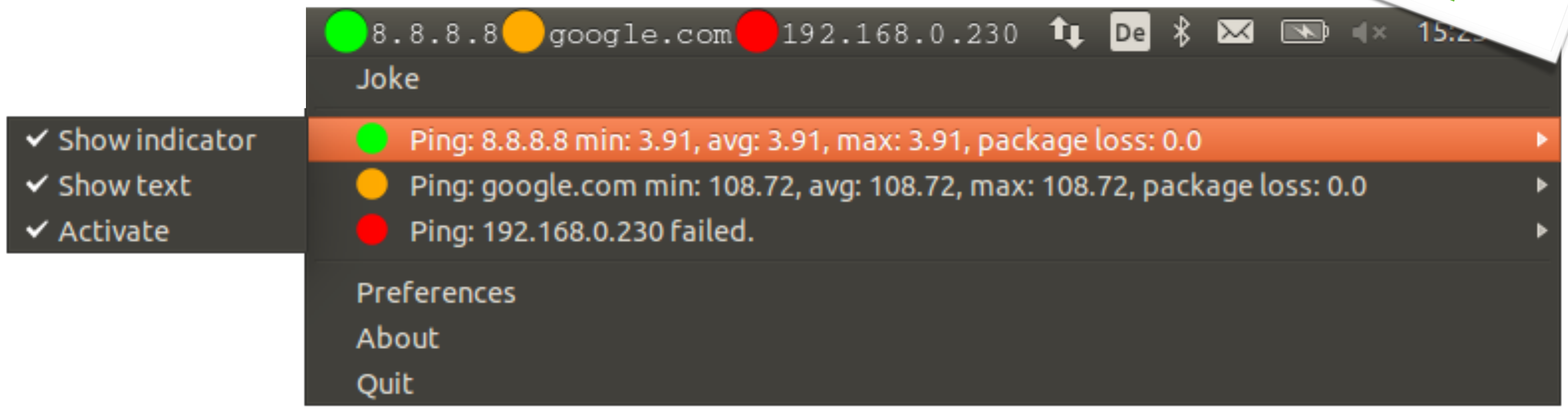


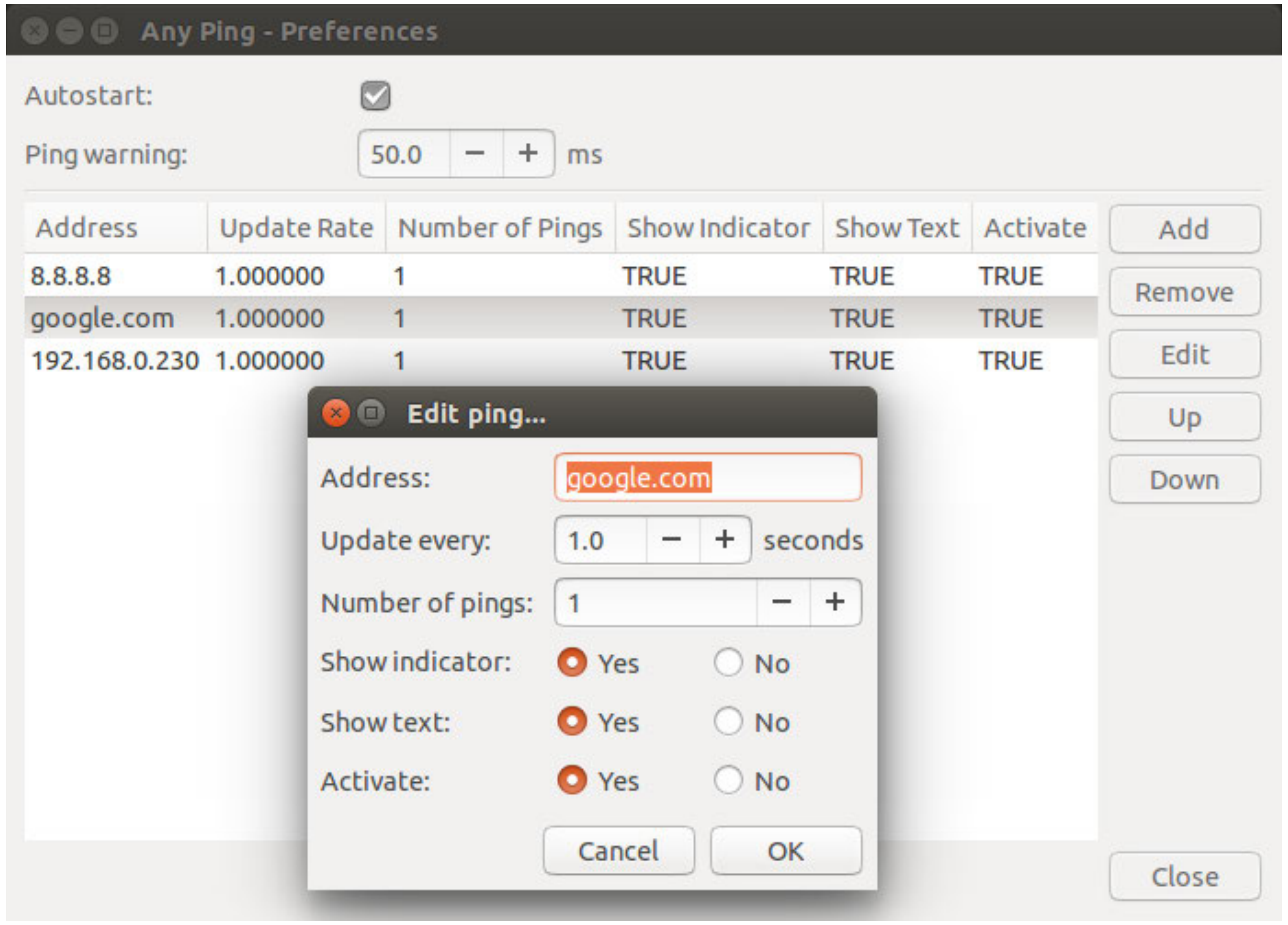
User Interface
Bandwidth Considerations

- Only critical data is transmitted by default (robot state and position)
- Other data is transmitted on demand (video, maps, etc.)
- Separation of onboard TF and operator TF
- Connection status node monitors WiFi status and triggers recovery behavior



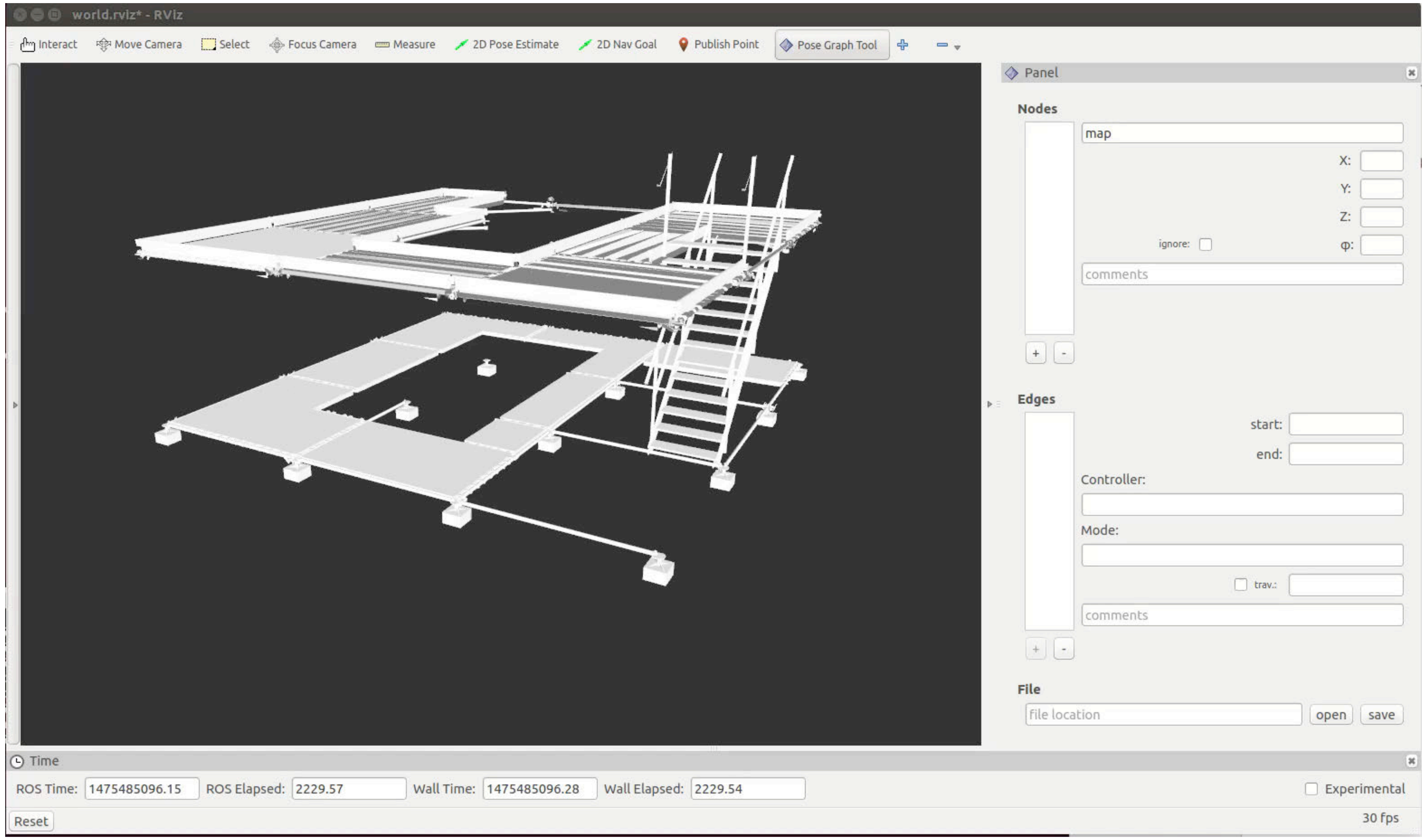
User Interface
ANYping Indicator





- Indicates PC network availability in Ubuntu menu bar

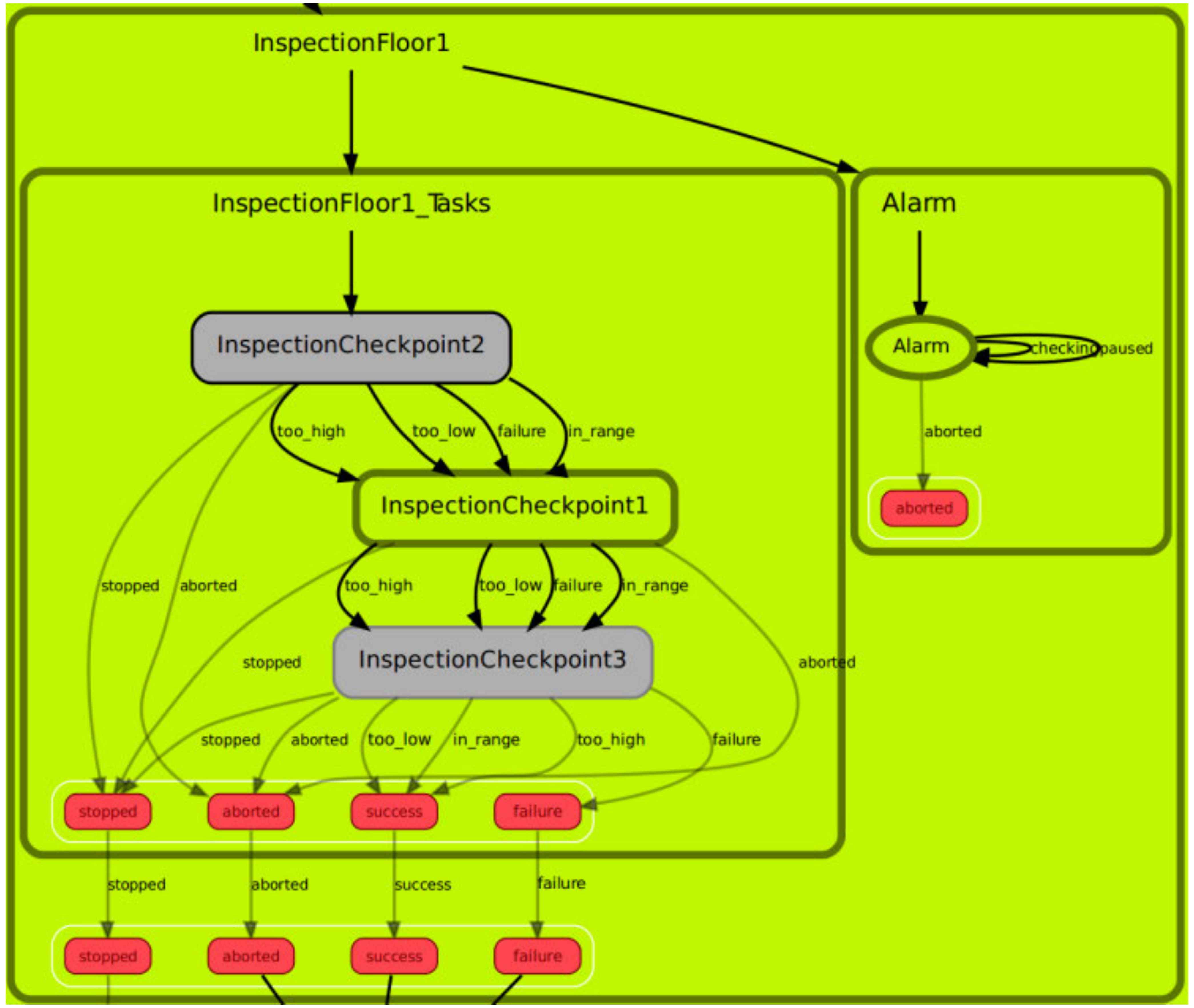
User Interface
Pose Graph



- Pose graph for inspection, special maneuvers (e.g. stairs), docking station etc.
- Visualization and interactive editing of pose graph
- Continuous updating and (re-)planning on pose graph during mission

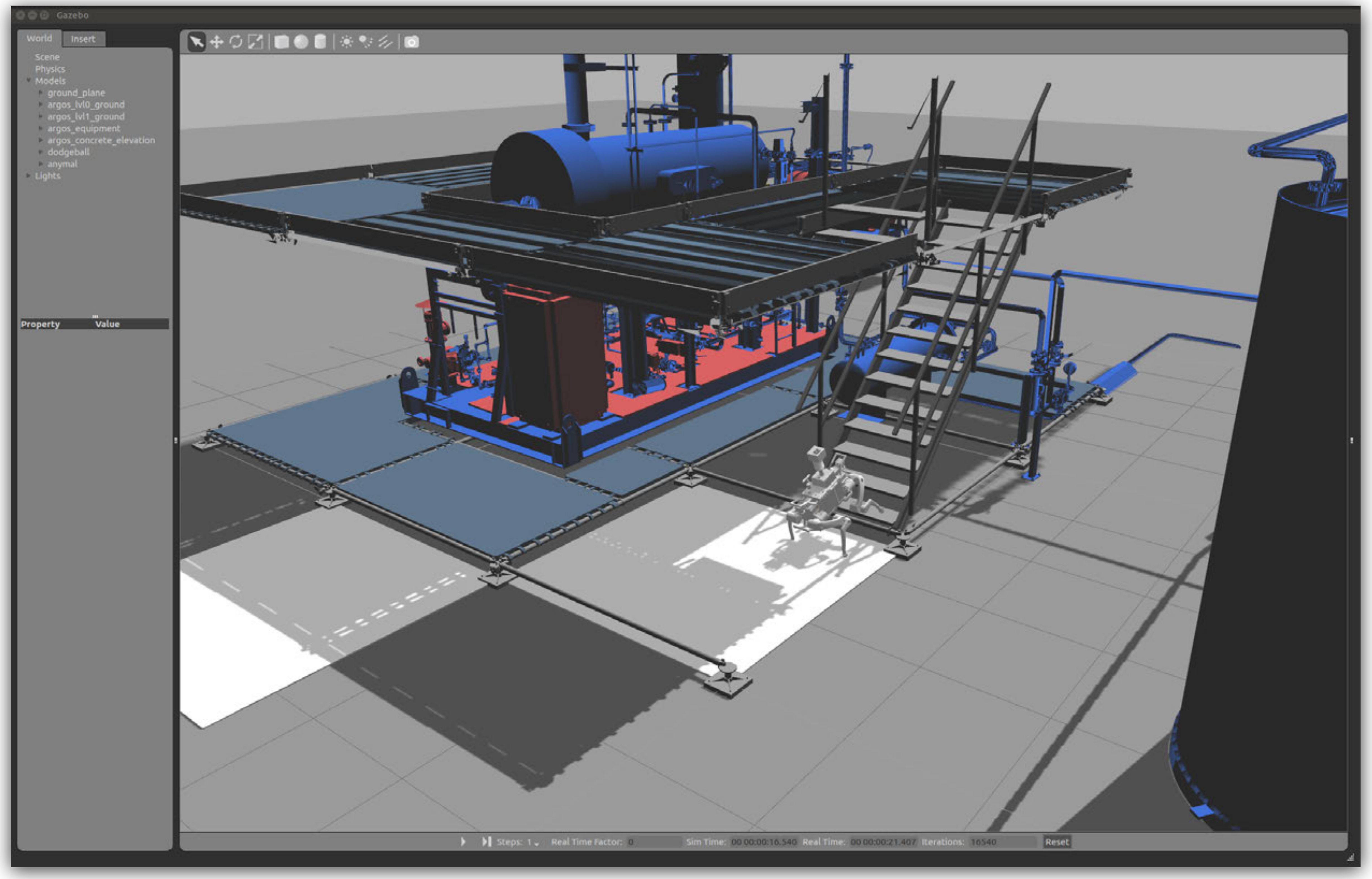
User Interface
Mission Creation

- Task-level state machine (C++ library, similar to SMACH)
- State machine defined in YAML format
- Common building blocks to facilitate construction
- Typical missions programmed in 5–20 minutes



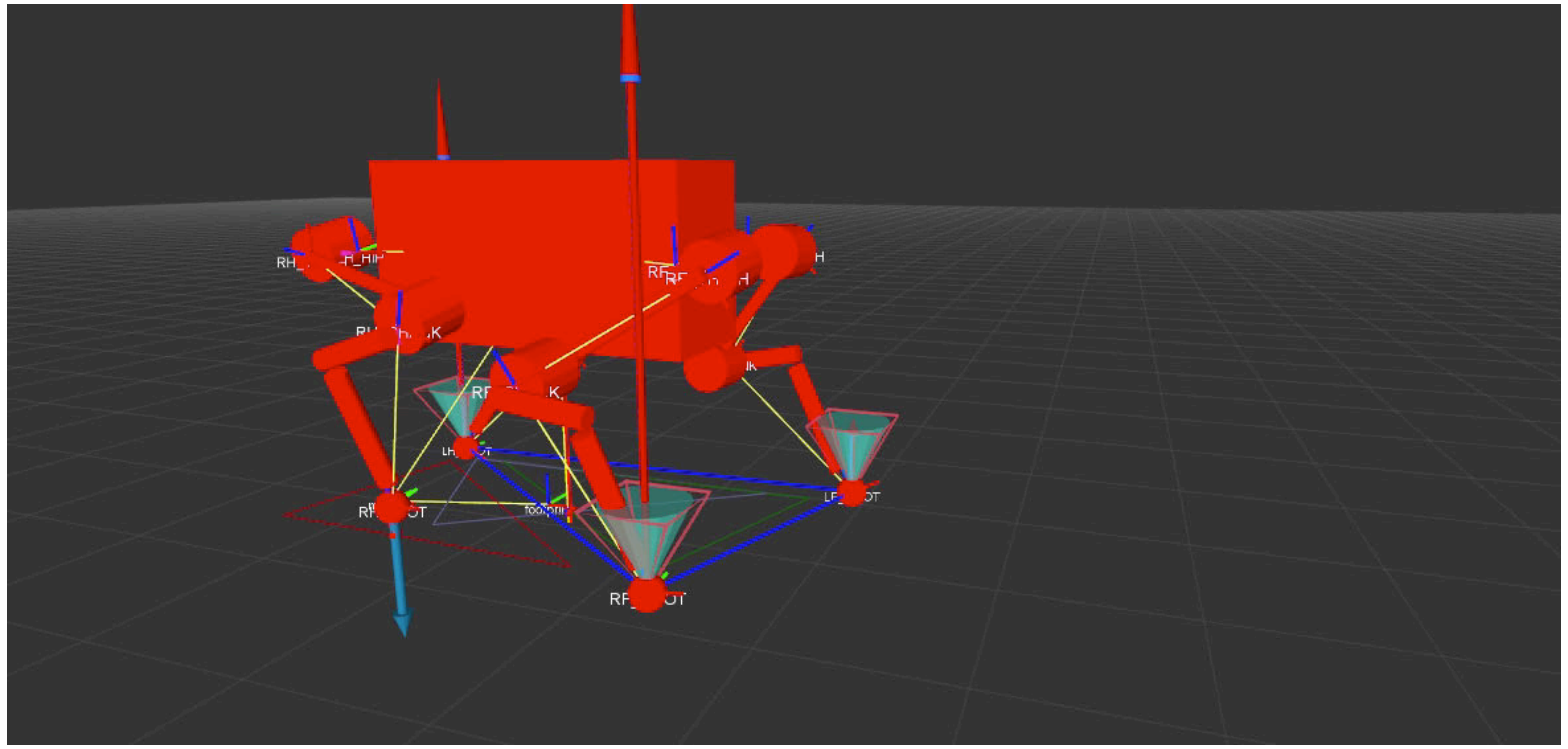
Software Tools – How We (Try) To Keep Things Smooth

- All developers and robots same setup
  - ➡ Ubuntu 16.04 LTS, ROS Kinetic
- Software version control with Git
  - ➡ Bitbucket & GitHub
- Conventions for package structure, format, naming, and code style
  - ➡ github.com/ethz-asl/ros_best_practices/wiki
- Extensive use of simulation
  - ➡ Gazebo



Software Tools – How We (Try) To Keep Things Smooth

- All developers and robots same setup
  - ➡ Ubuntu 16.04 LTS, ROS Kinetic
- Software version control with Git
  - ➡ Bitbucket & GitHub
- Conventions for package structure, format, naming, and code style
  - ➡ github.com/ethz-asl/ros_best_practices/wiki
- Extensive use of simulation
  - ➡ Gazebo
- Visualizing as much as possible

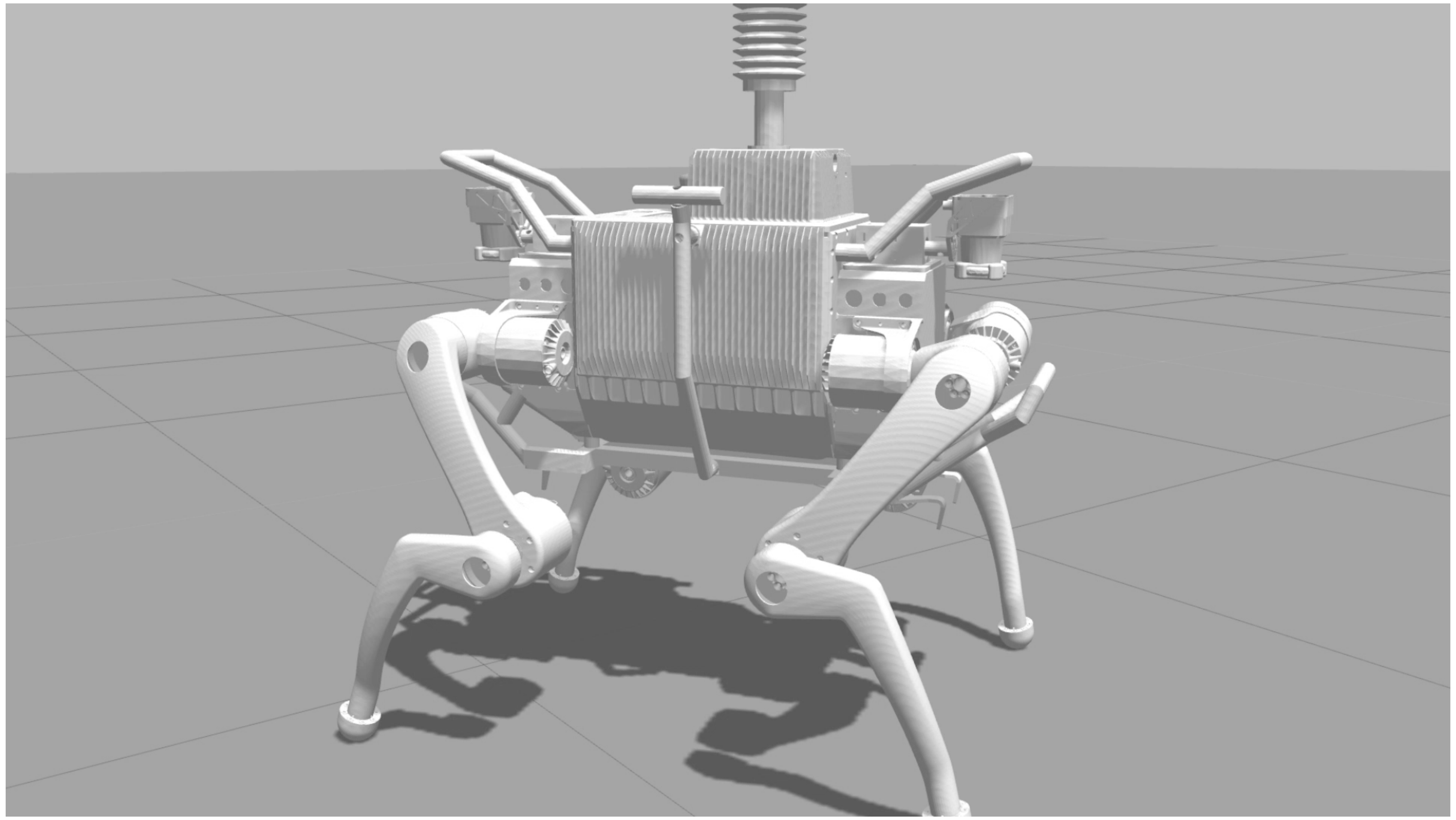


Software Tools – How We (Try) To Keep Things Smooth

- Lots of tests on hardware
  - ➡ Weekly “shakeouts” for defined tasks
  - ➡ Lots of demos
- Continuous Integration
  - ➡ Jenkins
  - ➡ Unit tests (after each change)
  - ➡ ROS integration tests (at night)

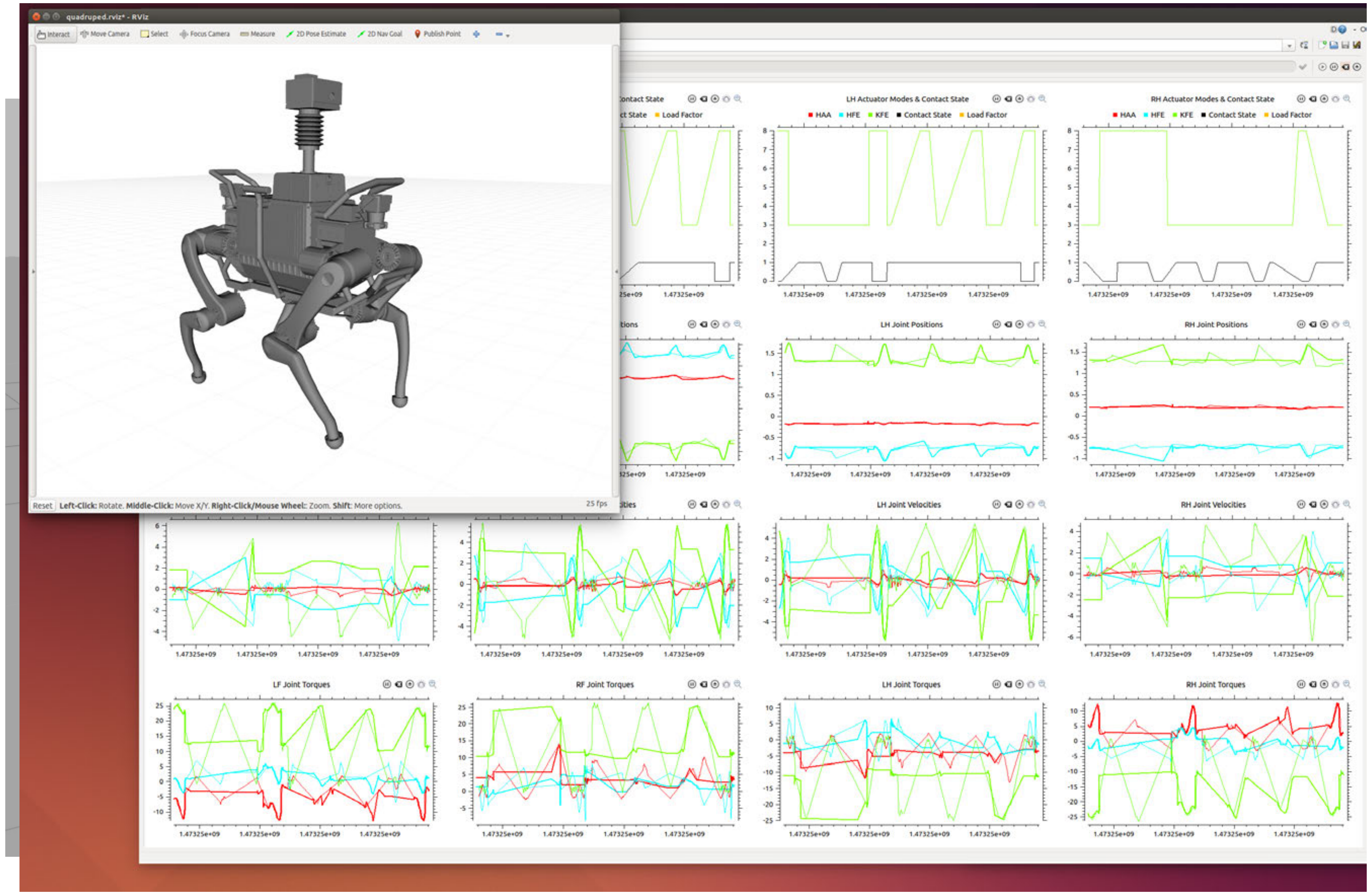
Software Tools – How We (Try) To Keep Things Smooth

- Lots of tests on hardware
  - ➡ Weekly “shakeouts” for defined tasks
  - ➡ Lots of demos
- Continuous Integration
  - ➡ Jenkins
  - ➡ Unit tests (after each change)
  - ➡ ROS integration tests (at night)



Software Tools – How We (Try) To Keep Things Smooth

- Lots of tests on hardware
  - ➡ Weekly “shakeouts” for defined tasks
  - ➡ Lots of demos
- Continuous Integration
  - ➡ Jenkins
  - ➡ Unit tests (after each change)
  - ➡ ROS integration tests (at night)
- Logging (rosbag)
  - ➡ All important information is always logged
  - ➡ Review logs with RViz and RQT Multiplot



Thank You

Open-Source Software

github.com/ethz-asl

github.com/leggedrobotics

Péter Fankhauser
pfankhauser@anybotics.com

RSL Robotic Systems Lab

www.rsl.ethz.ch

ANYbotics

www.anybotics.com