

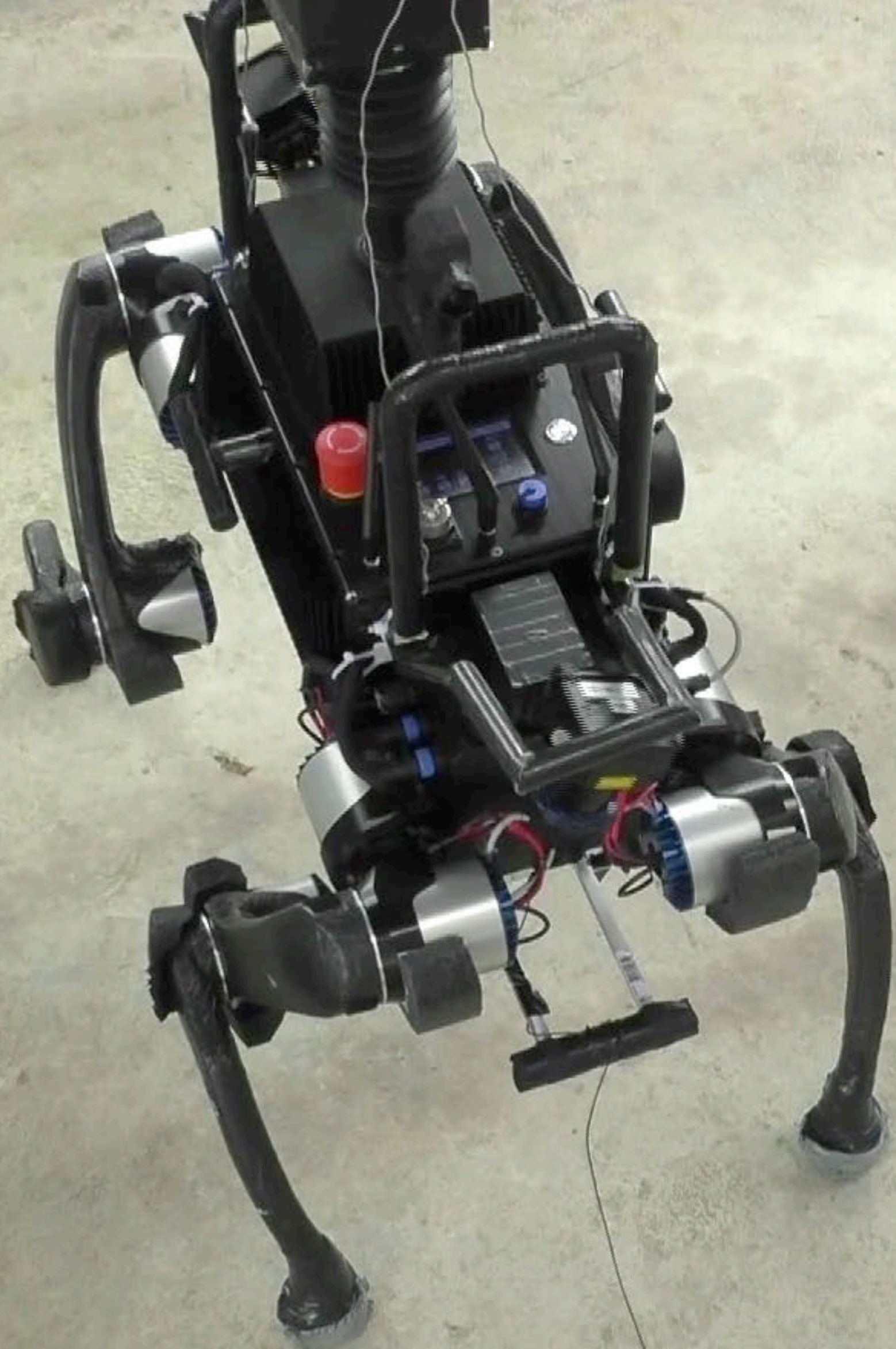
ANYmal at the ARGOS Challenge

Tools and Experiences from the Autonomous Inspection of Oil & Gas Sites with a Legged Robot

Péter Fankhauser

Remo Diethelm, Samuel Bachmann, Christian Gehring, Martin Wermelinger, Dario Bellicoso, Vassilios Tsounis, Andreas Lauber, Michael Bloesch, Philipp Leemann, Gabriel Hottiger, Dominik Jud, Ralf Kaestner, Linus Isler, Mark Hoepflinger, Roland Siegwart, Marco Hutter

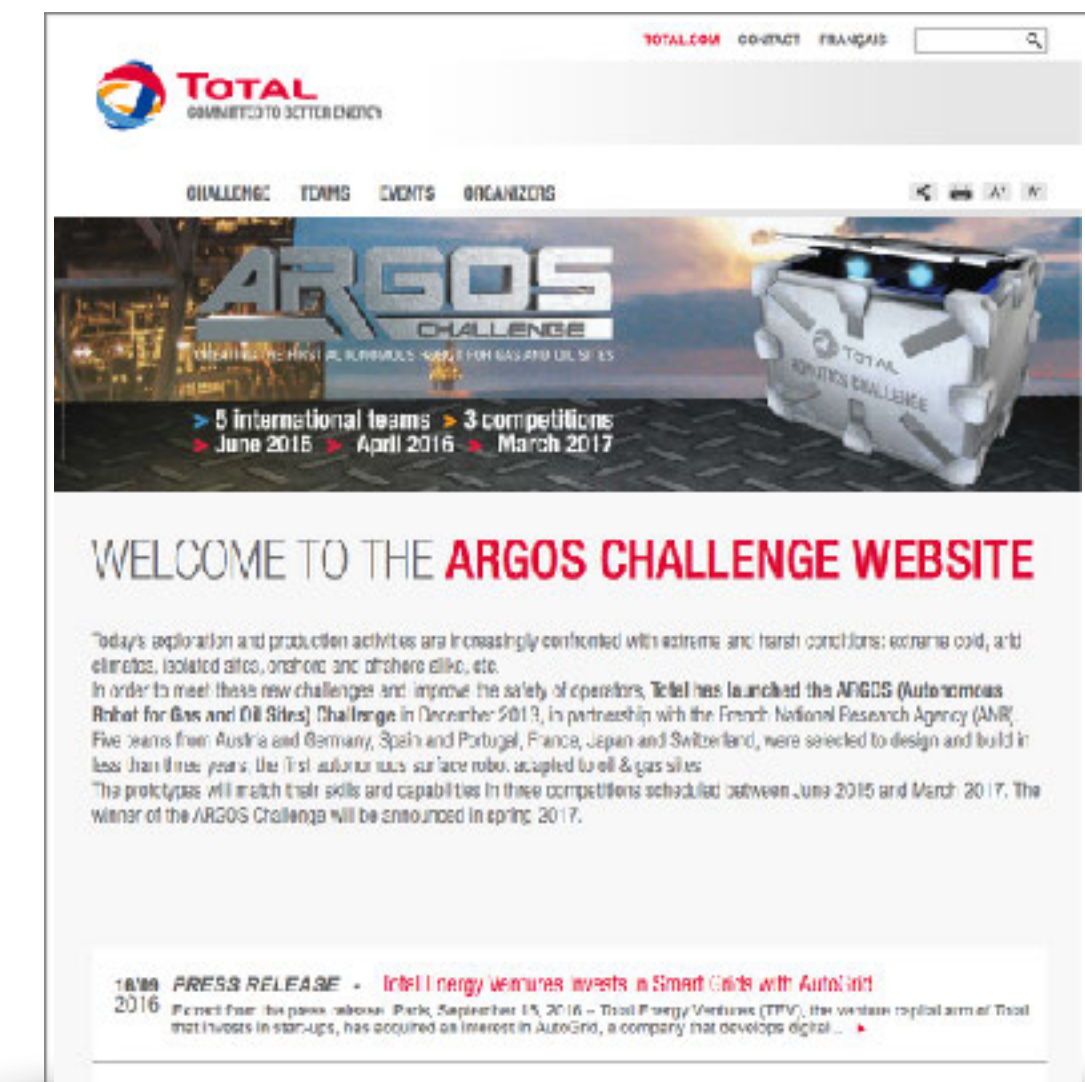
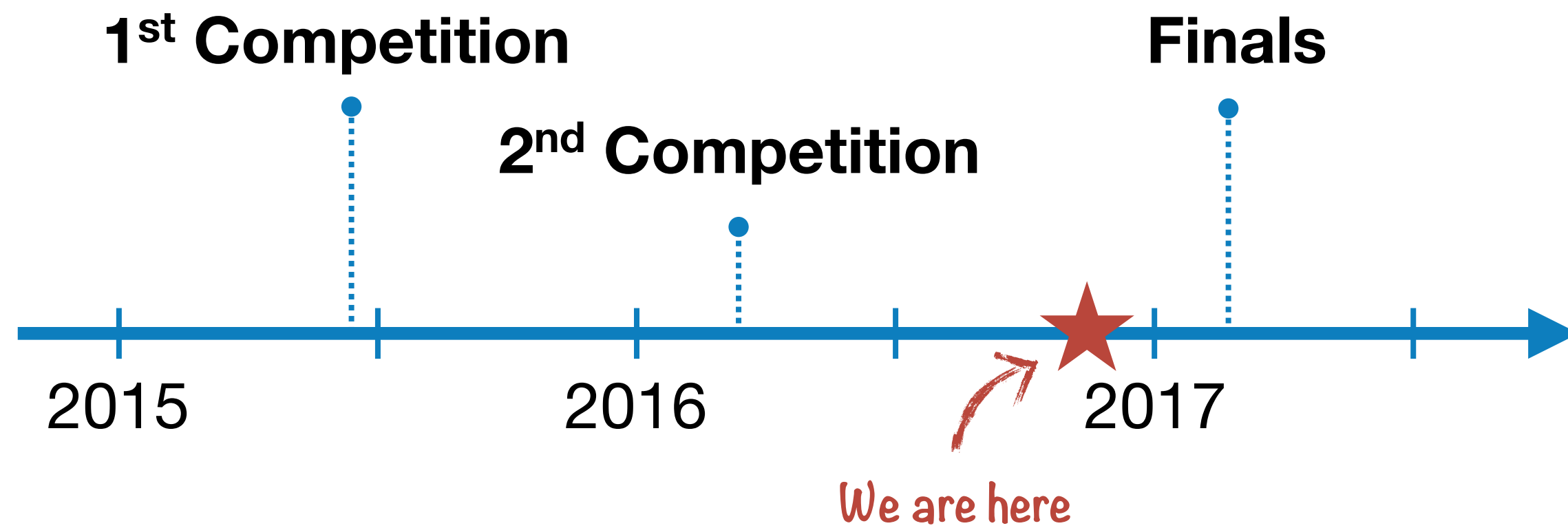




ANYmal for the
Oil & Gas Industry

ARGOS Challenge

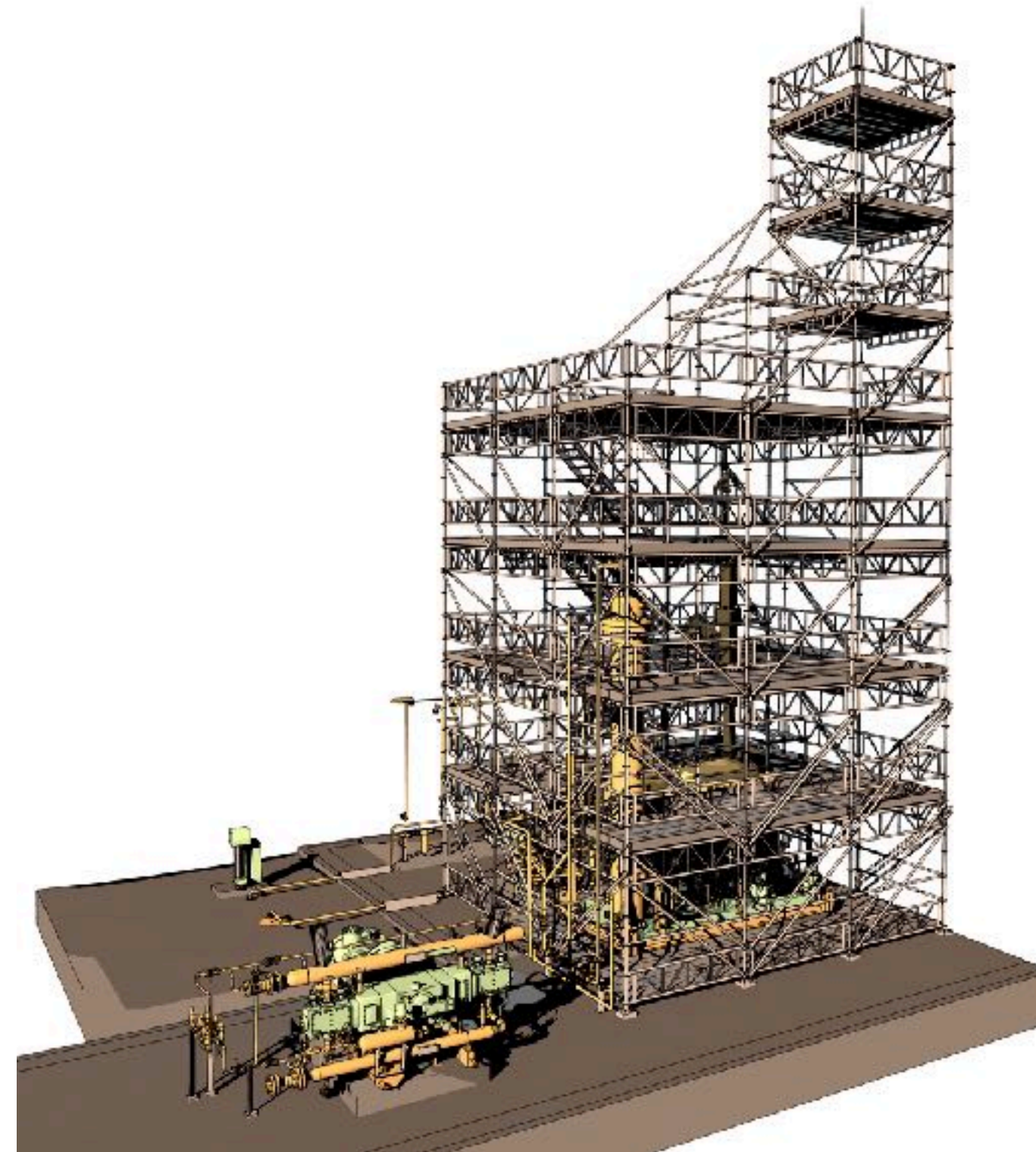
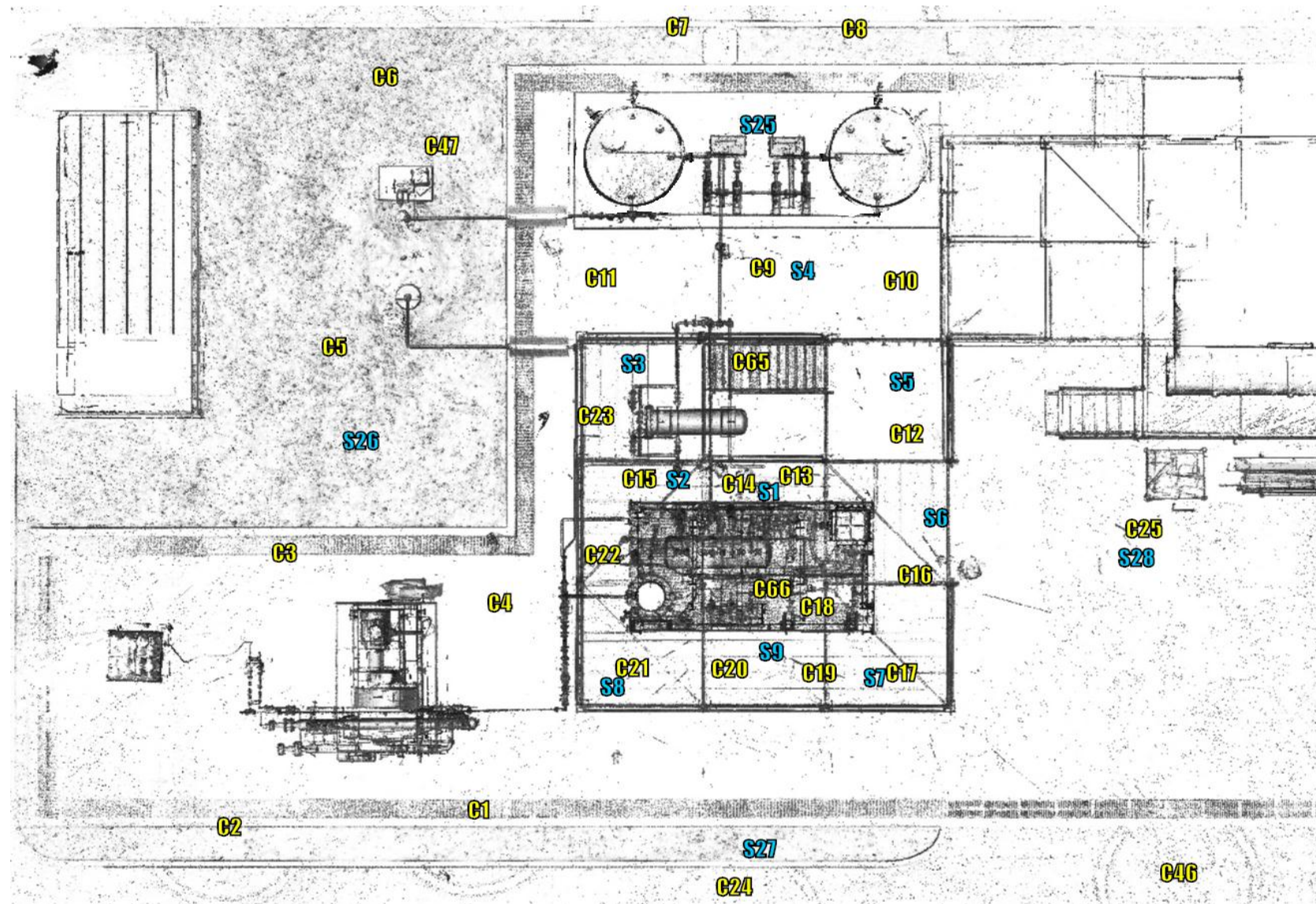
“Creating the first autonomous robot for gas and oil sites”






<http://www.argos-challenge.com>

ARGOS Challenge

Autonomous Inspection of Oil and Gas 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. Reacting to its status sensors. The GPA is not activated by the operator (except in simulation condition, see run-robot).
- Acoustic leak detection (3-2):**
gas leaks exist in the ultrasonic range (25kHz-70kHz, dynamic range 50-100dB SPL).
- Checkpoints:**
checkpoints will have to be detected around their exact position, and their absence reported:
 - Pressure gauge (3-6)
 - Water level (3-7)
 - Yaws (3-8)
 - Thermal measurement points (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 in all cases the expected normal sound.
- Unexpected leak sources (3-11):**
The robot will have to detect unexpected leak sources in the facility (refer to requirement 4 - Sources of high temperature, page 22).
- Obstacles (3-12):**
unexpected obstacles during the 2nd competition can be objects or part of structure on the ground (positive obstacles) or part of the ground (expected elevations), 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:

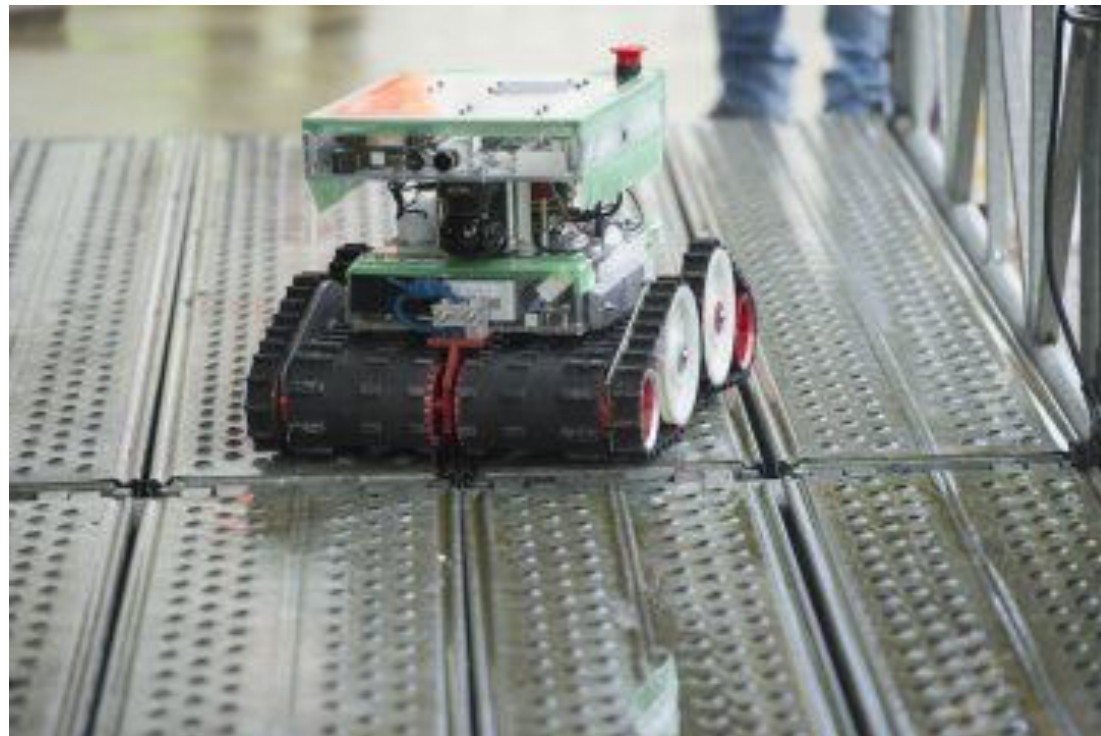
- Gas leak - IR detection (3-3)
- gas leak detection and localization based on the point IR detector.
- Safety equipment - extinguishers (2-4):
The robot has to detect whether the extinguishers are present in their expected locations.
- allow and plug (3-9):
monitoring the environment to identify missing plugs on open pipes.

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

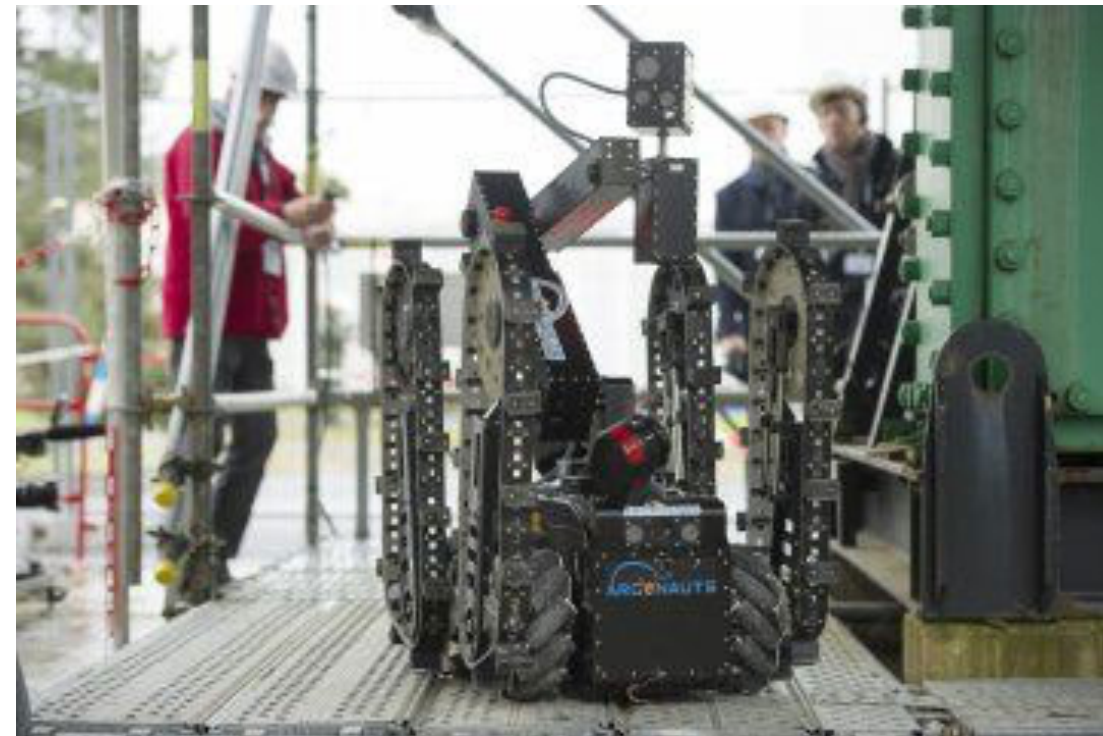
[21]

ARGOS Challenge

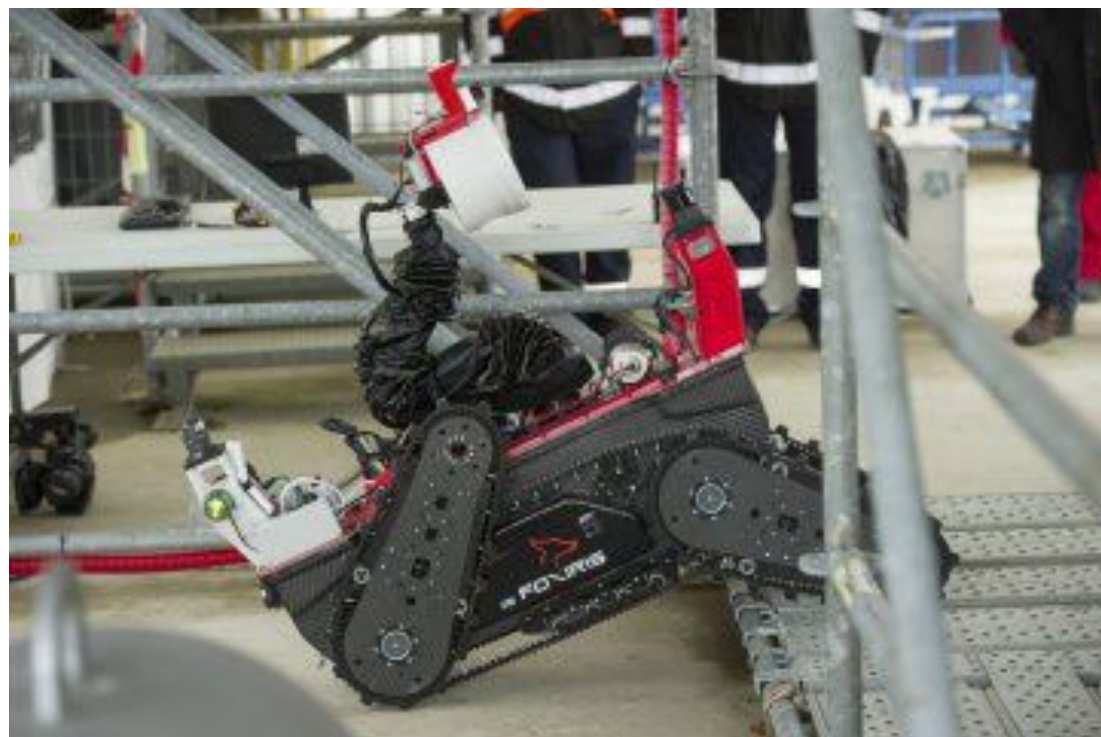
5 International Teams



AIR-K
Japan



ARGONAUTS
Austria & Germany



FOXIRIS
Spain & Portugal



VIKINGS
France

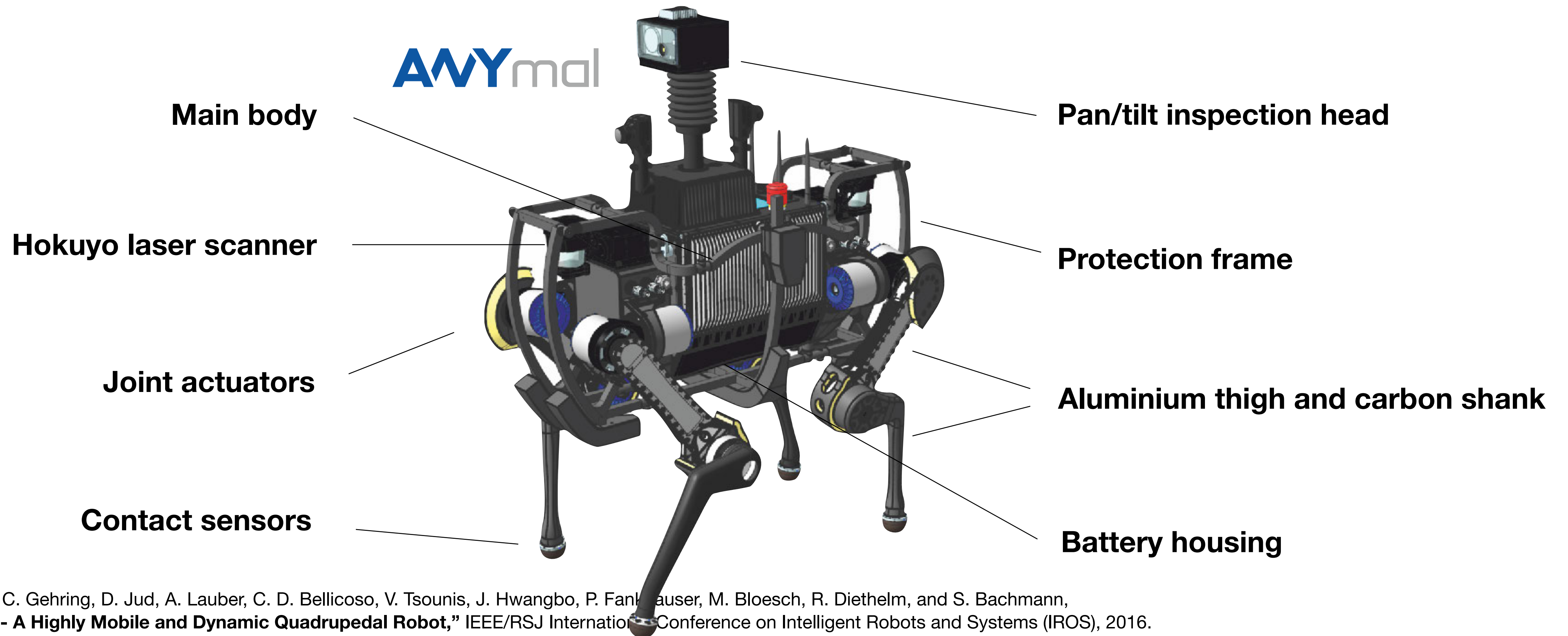


LIO
Switzerland

ARGOS Challenge Team LIO



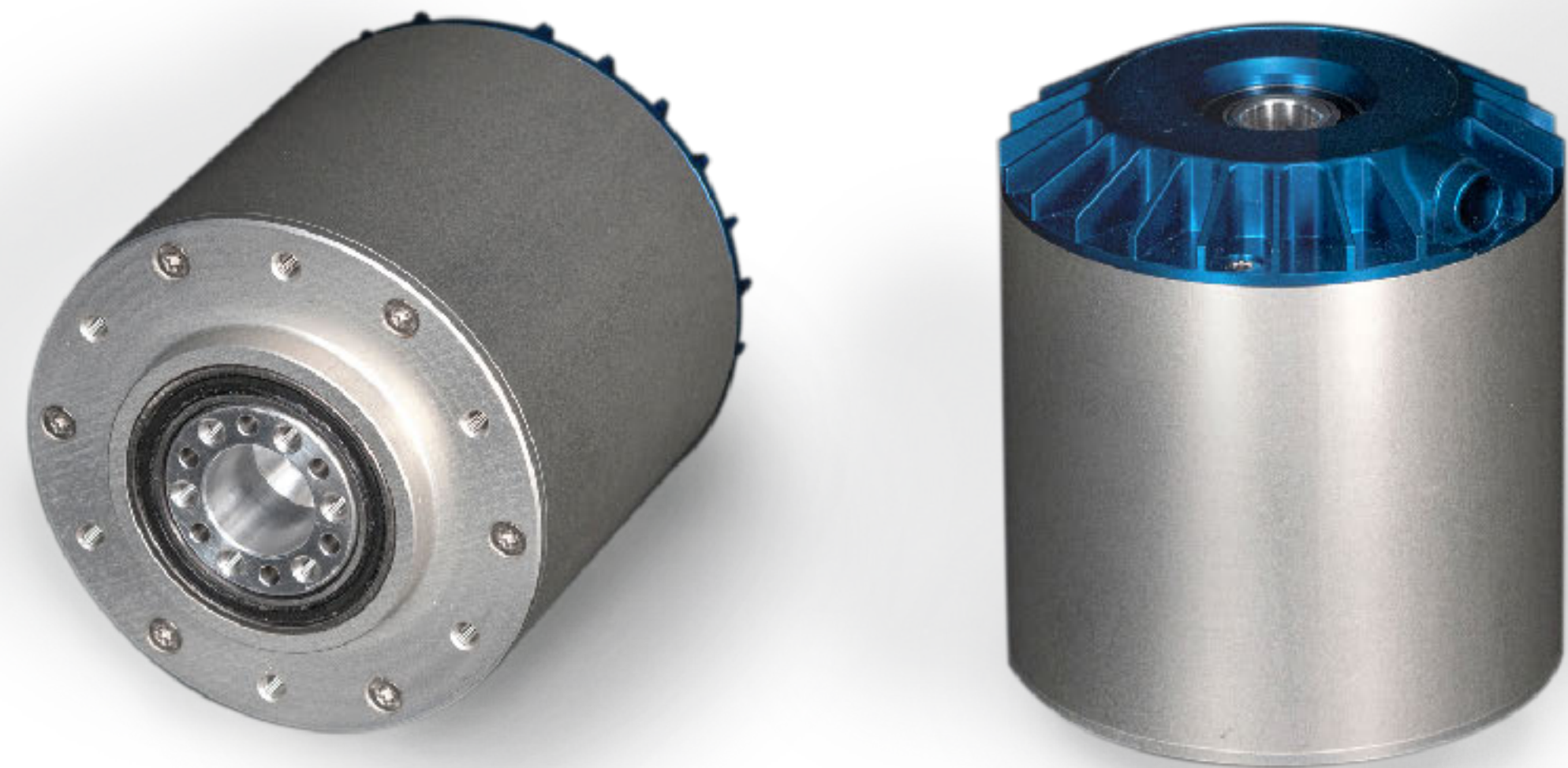
ANYmal – A High-Performance & Versatile Quadrupedal Robot



M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, P. Fankhauser, M. Bloesch, R. Diethelm, and S. Bachmann, "ANYmal - A Highly Mobile and Dynamic Quadrupedal Robot," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016.

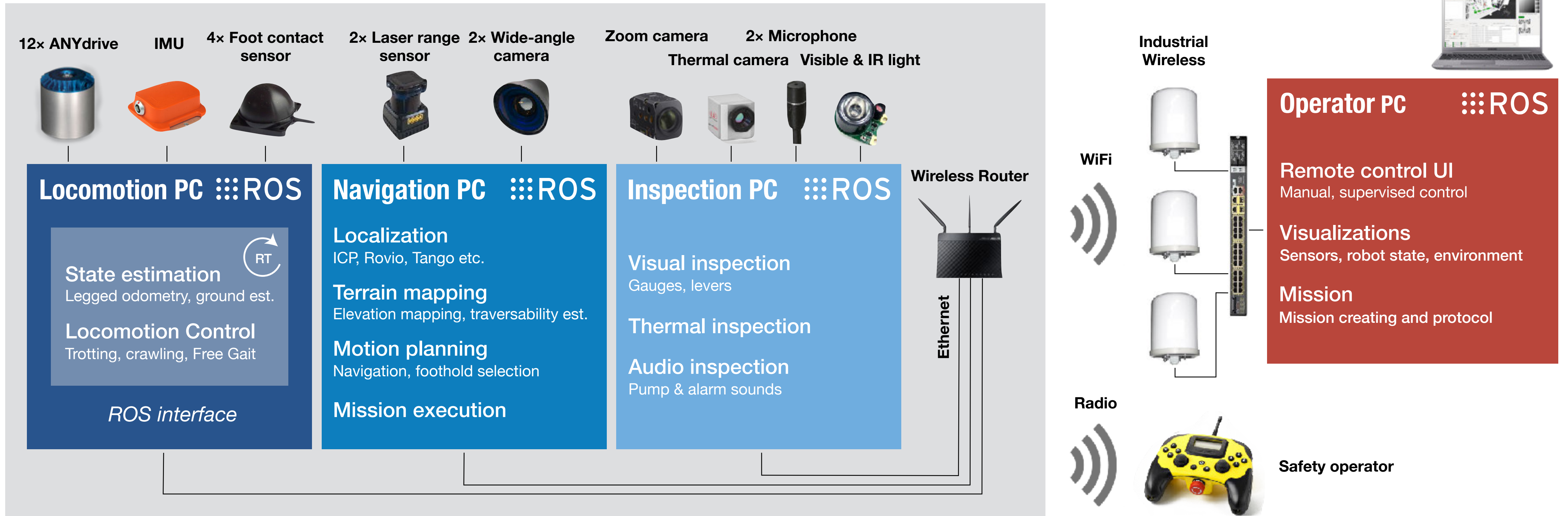
ANYdrive – A Integrated, Robust, Torque-Controllable Robot Joint

- Fully integrated
- Accurate position & torque control
- Absolute position sensing
- Programmable controller
- Impact robust
- Hollow-shaft
- Water-proof

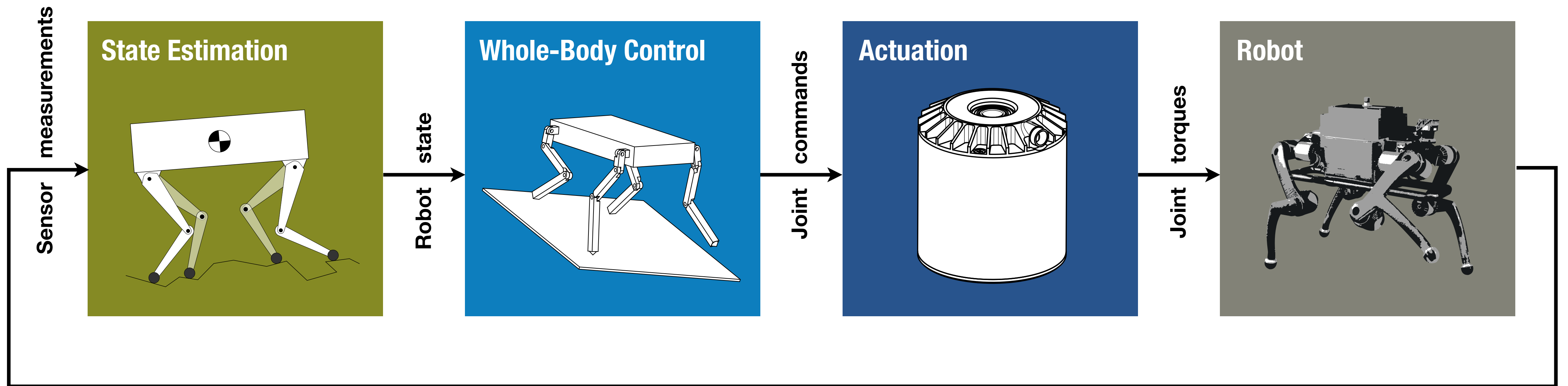


ANYdrive

System Overview



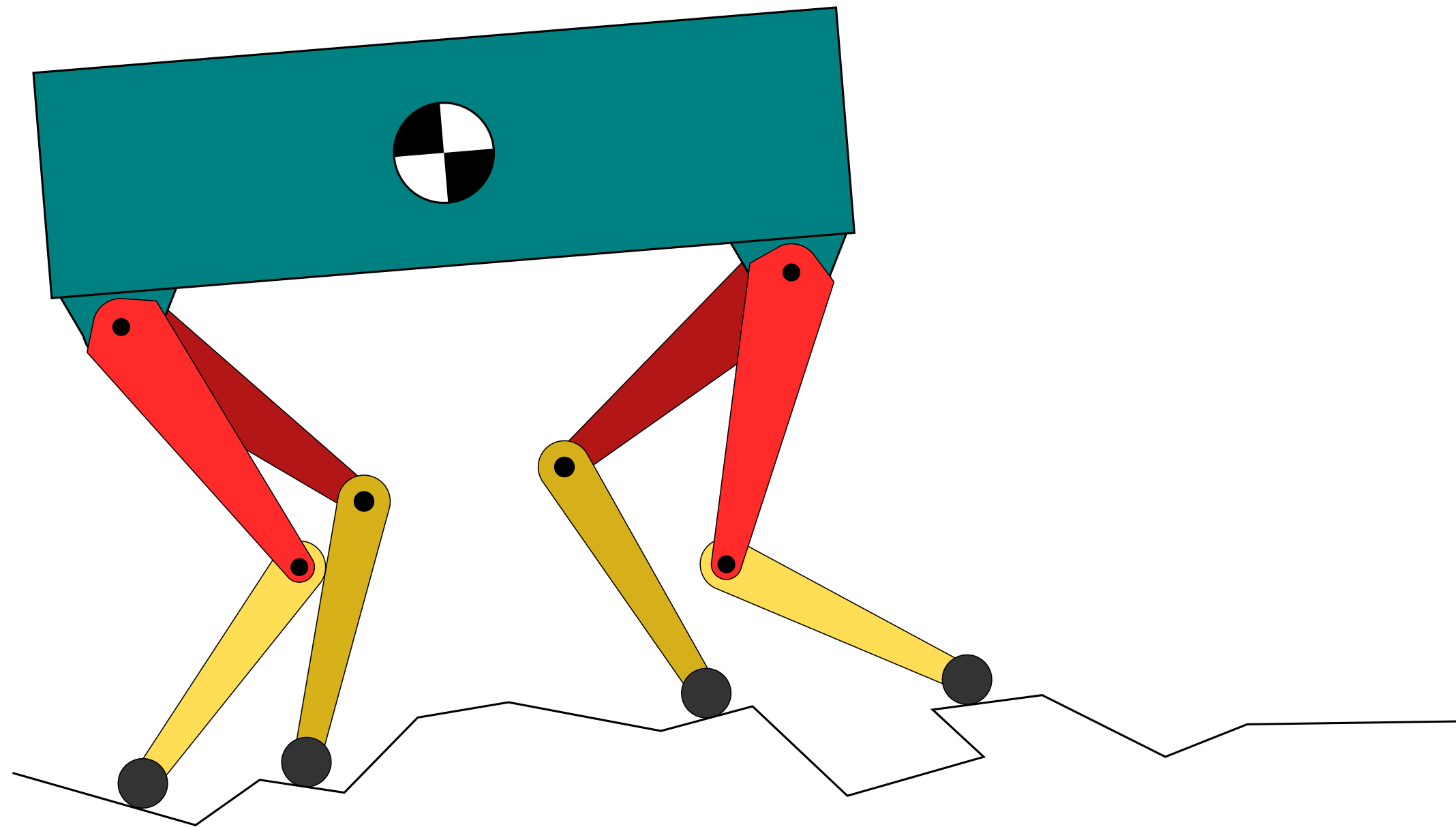
Locomotion



Locomotion State Estimation

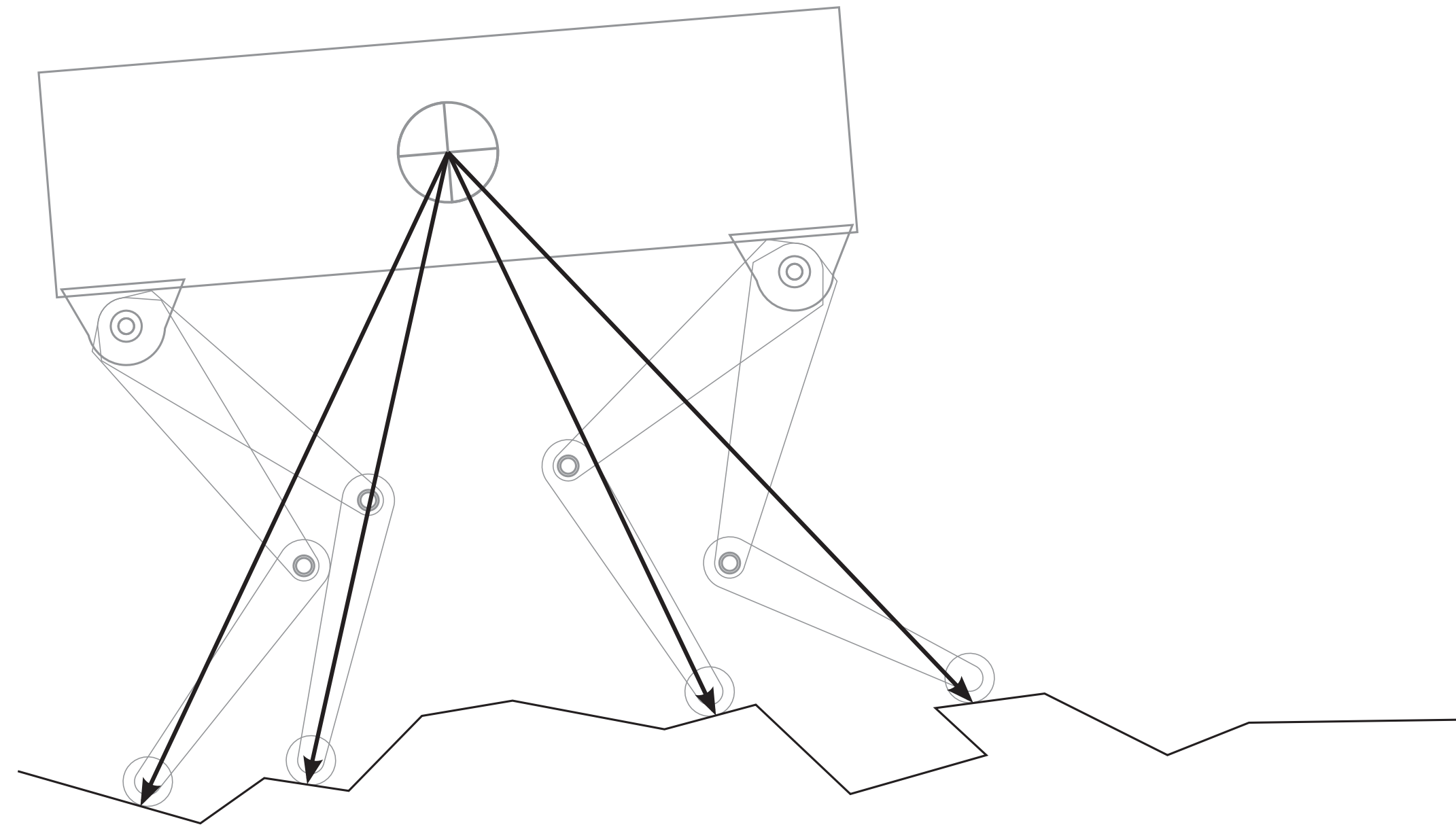
Extended Kalman Filter

- No assumption on terrain



M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger and R. Siegwart, “**State Estimation for Legged Robots on Unstable and Slippery Terrain**”, in International Conference on Intelligent Robots and Systems (IROS), 2013.

Locomotion State Estimation

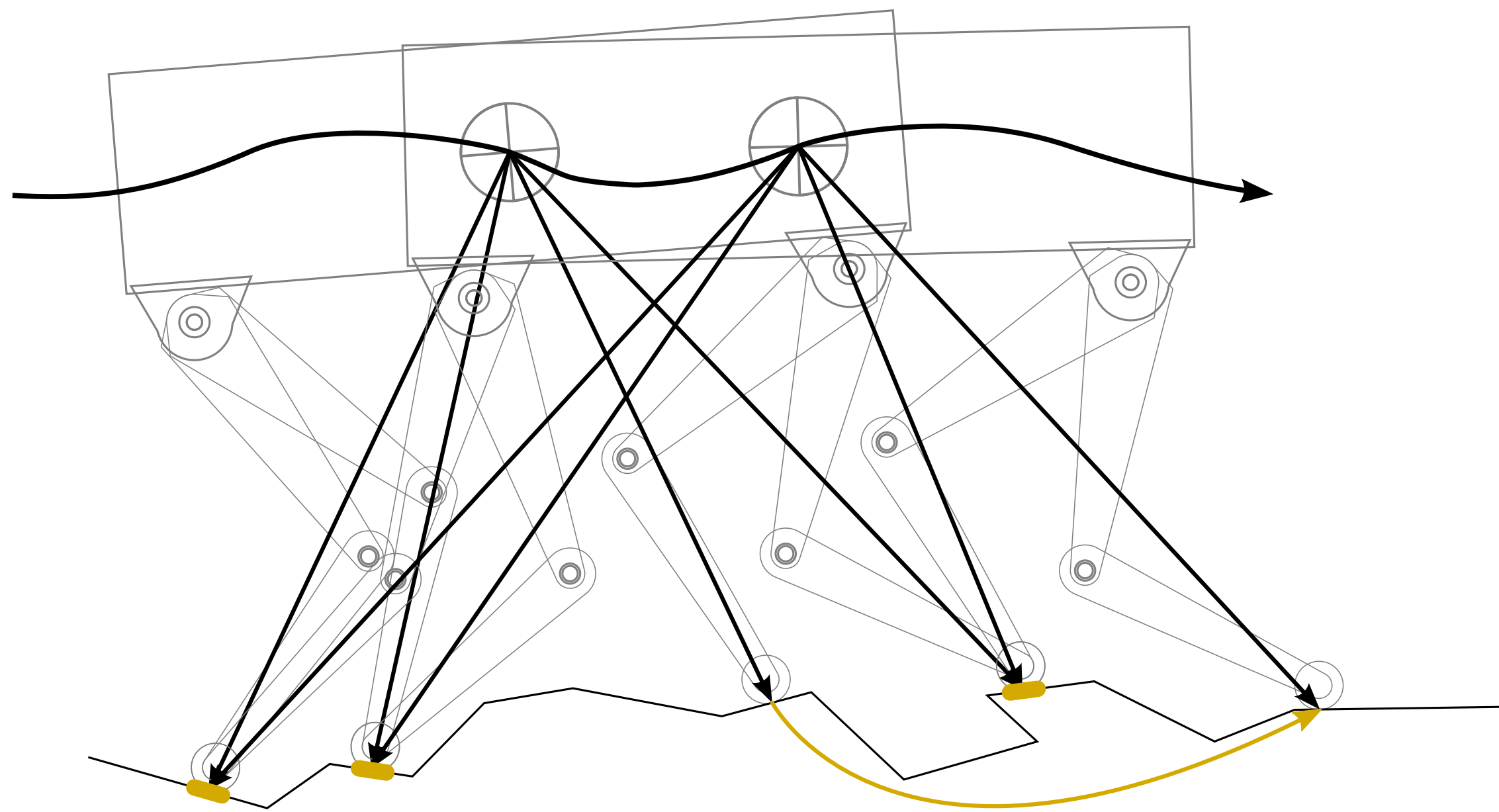


Extended Kalman Filter

- No assumption on terrain
- Kinematic measurements (encoders) for legs in contact

M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger and R. Siegwart, “**State Estimation for Legged Robots on Unstable and Slippery Terrain**”, in International Conference on Intelligent Robots and Systems (IROS), 2013.

Locomotion State Estimation



Kinematic measurements

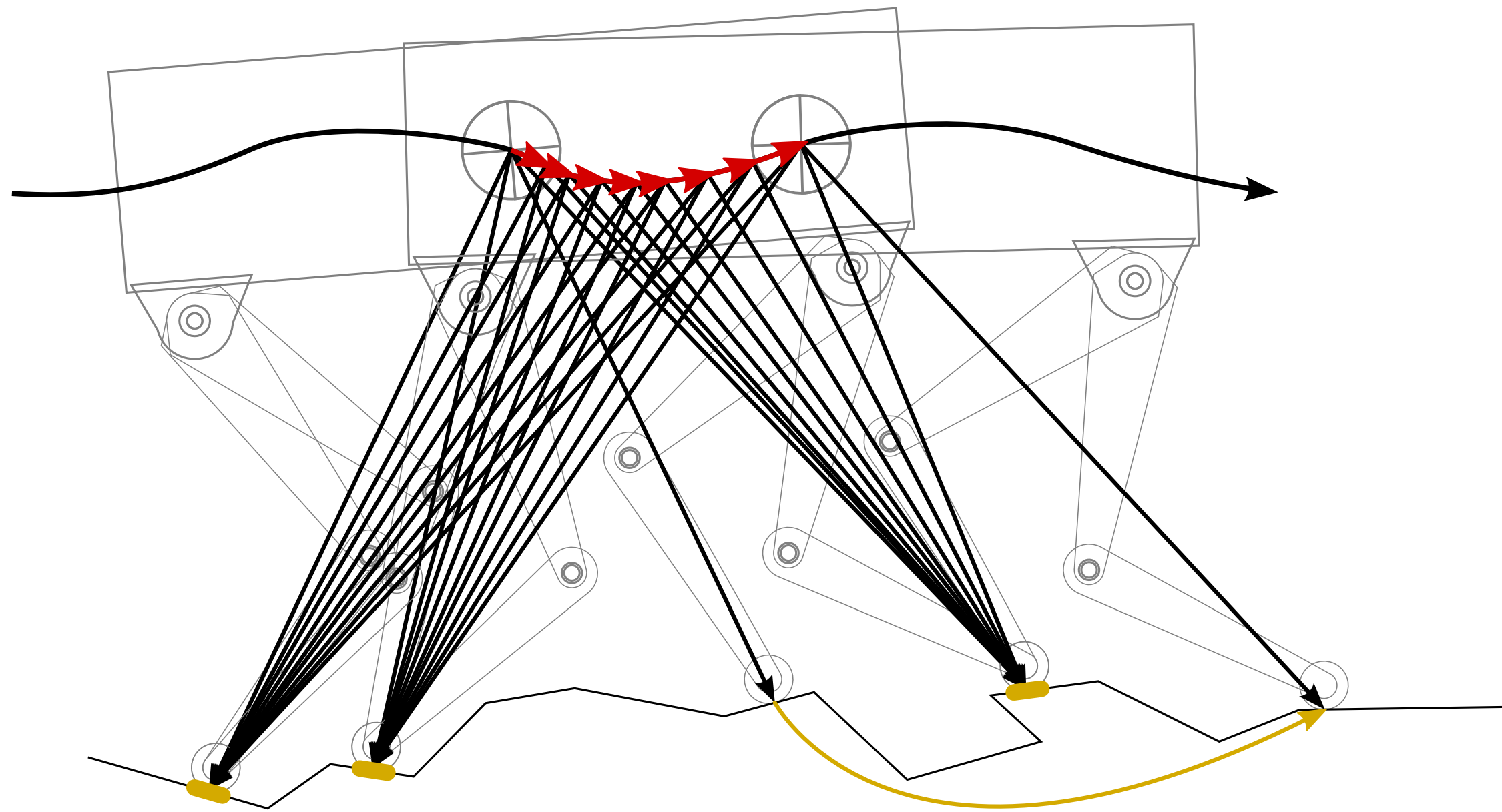
Extended Kalman Filter

- No assumption on terrain
- Kinematic measurements (encoders) for legs in contact

M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger and R. Siegwart, “**State Estimation for Legged Robots on Unstable and Slippery Terrain**”, in International Conference on Intelligent Robots and Systems (IROS), 2013.

Locomotion State Estimation

Inertial measurements



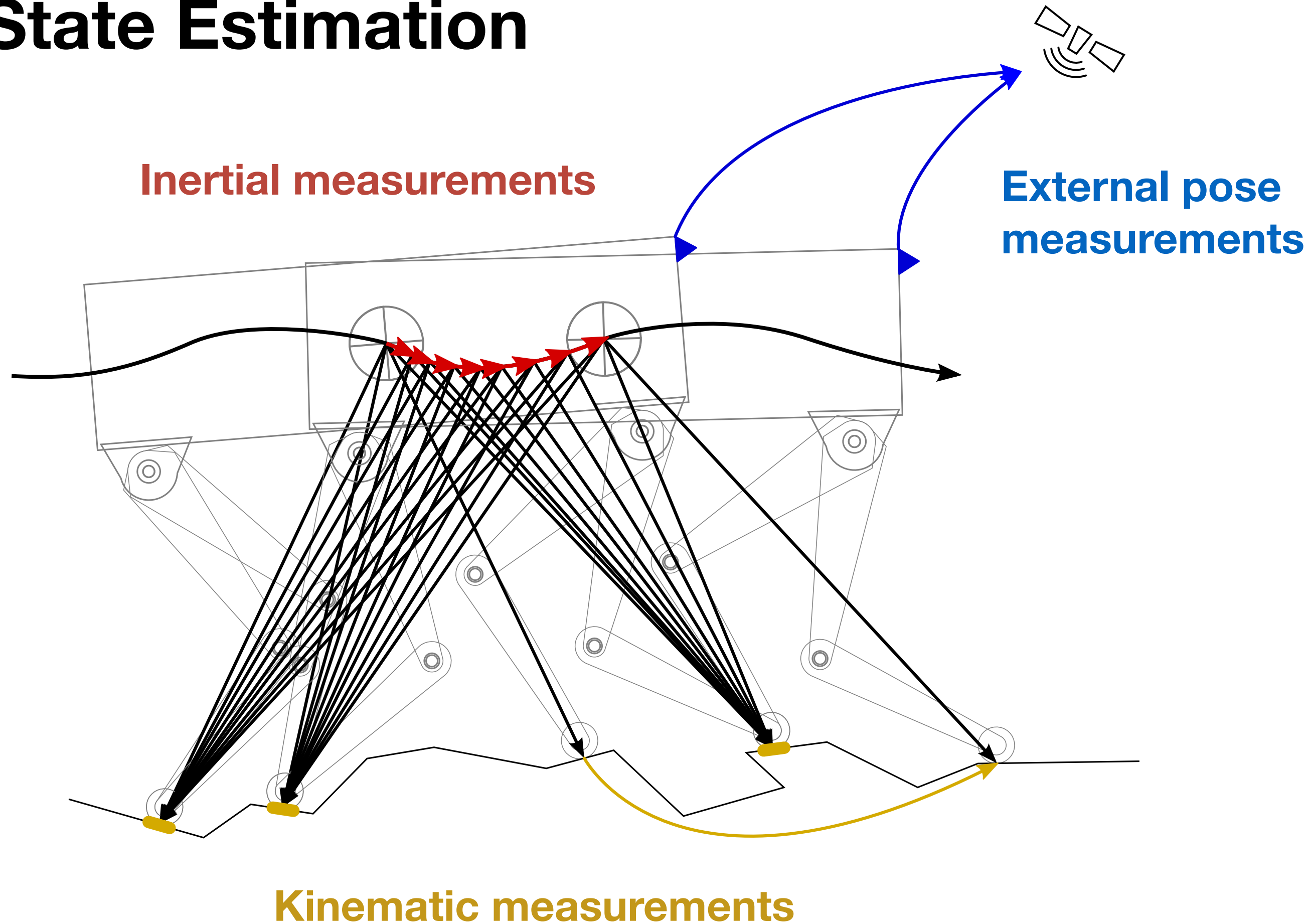
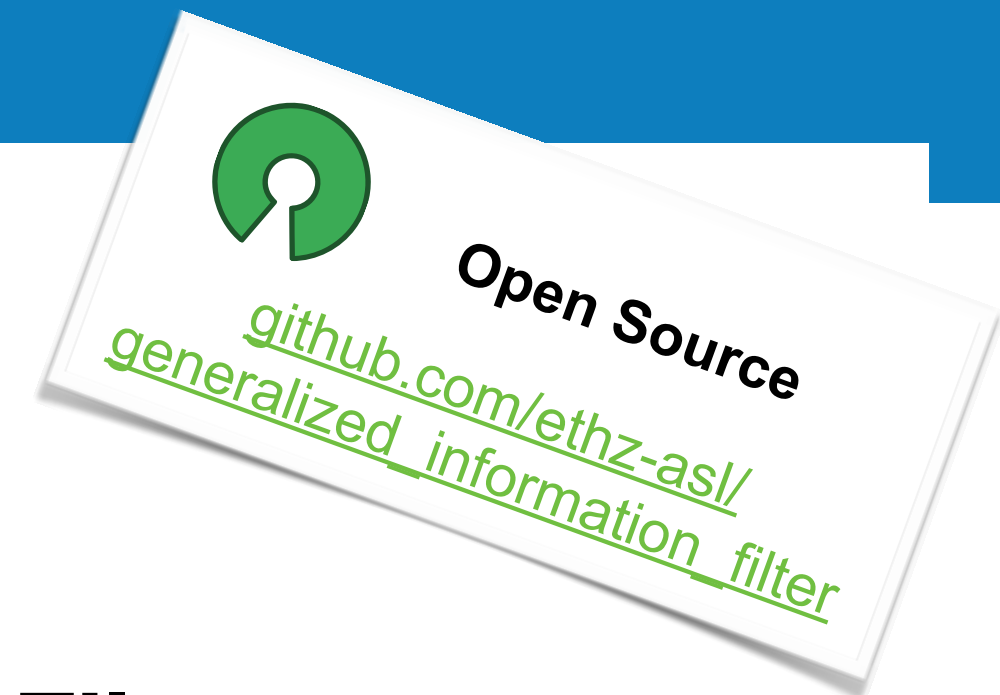
Kinematic measurements

Extended Kalman Filter

- No assumption on terrain
- Kinematic measurements (encoders) for legs in contact
- Fused with inertial measurements (IMU)
- Error < 5% over distance

M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger and R. Siegwart, "State Estimation for Legged Robots on Unstable and Slippery Terrain", in International Conference on Intelligent Robots and Systems (IROS), 2013.

Locomotion State Estimation



Extended Kalman Filter

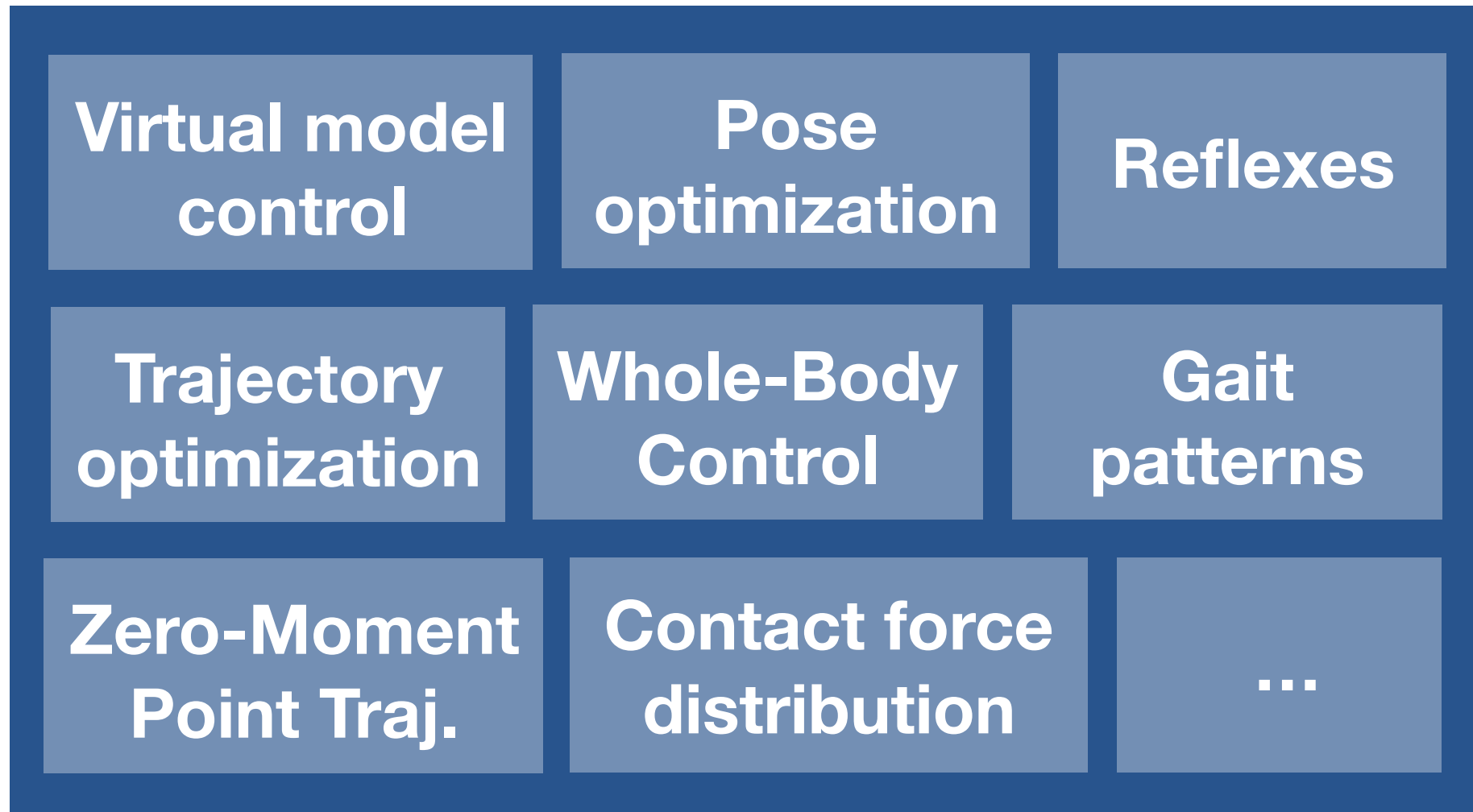
- No assumption on terrain
- Kinematic measurements (encoders) for legs in contact
- Fused with inertial measurements (IMU)
- Error < 5% over distance
- Optionally combined with external pose (GPS, laser, vision, etc.)

M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger and R. Siegwart, "State Estimation for Legged Robots on Unstable and Slippery Terrain", in International Conference on Intelligent Robots and Systems (IROS), 2013.

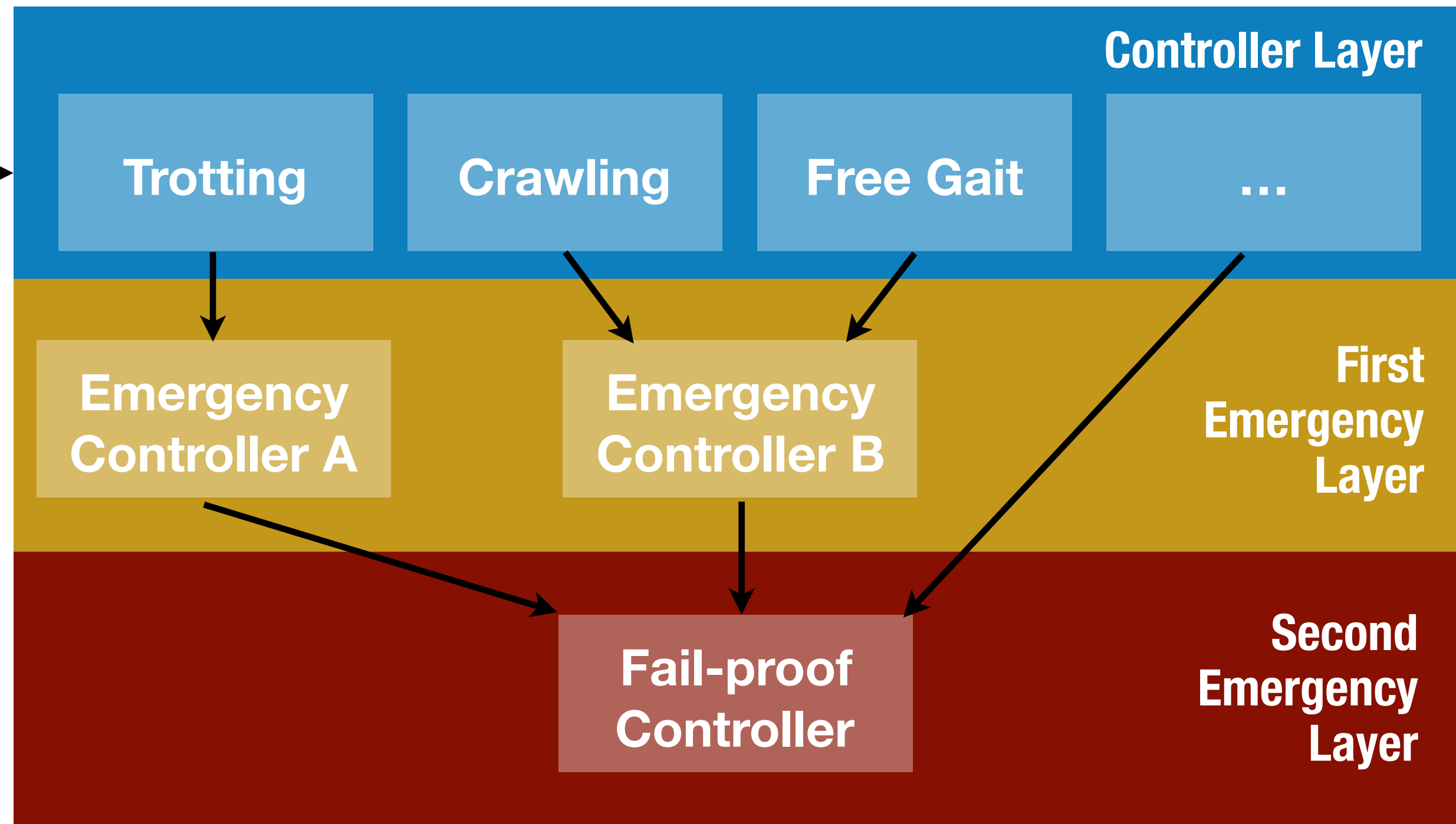
Locomotion Whole-Body Control



Locomotion Controller Modules (Loco)



Robot Controller Manager (Rocoma)



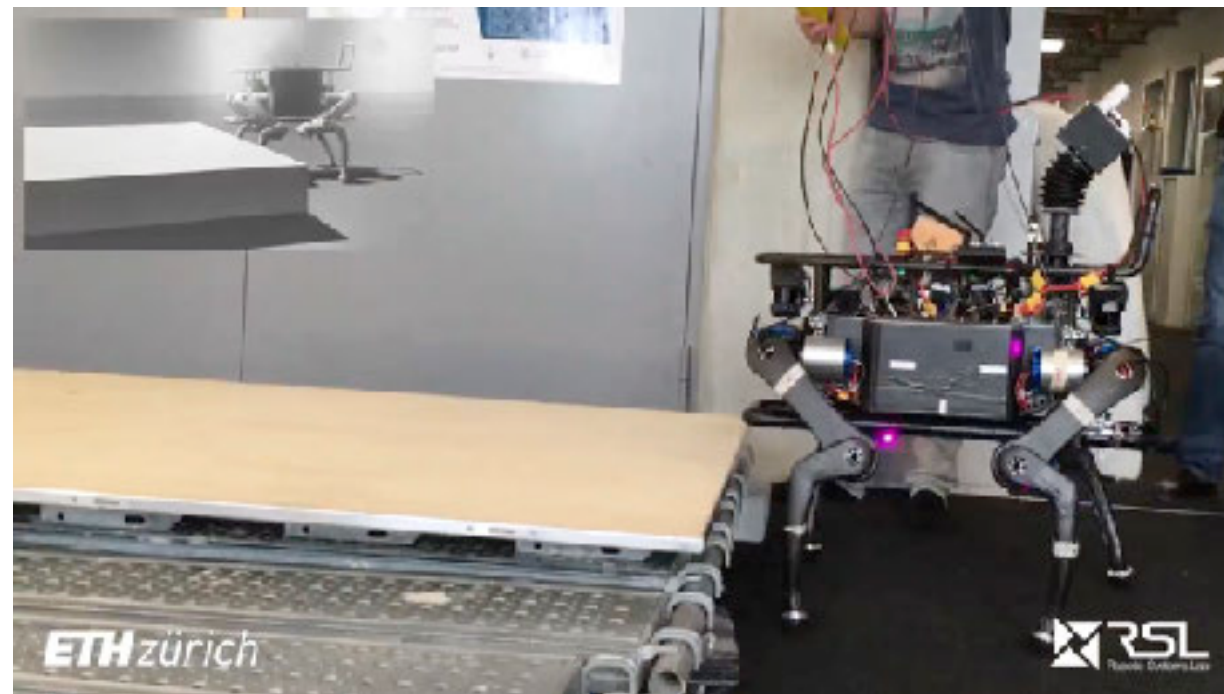
C. Gehring, S. Coros, M. Hutter, D. Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. A. Hoepflinger, and R. Siegwart, "Practice Makes Perfect: An Optimization-Based Approach to Controlling Agile Motions for a Quadruped Robot.", in IEEE Robotics & Automation Magazine, 2016.

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.

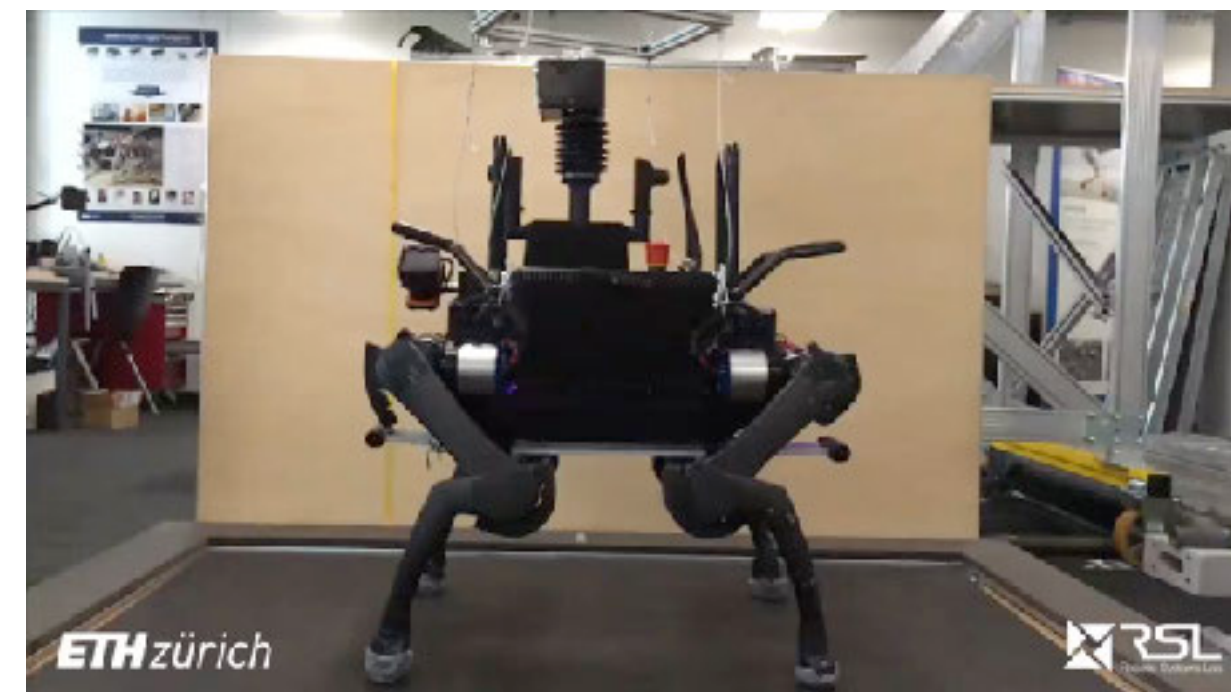
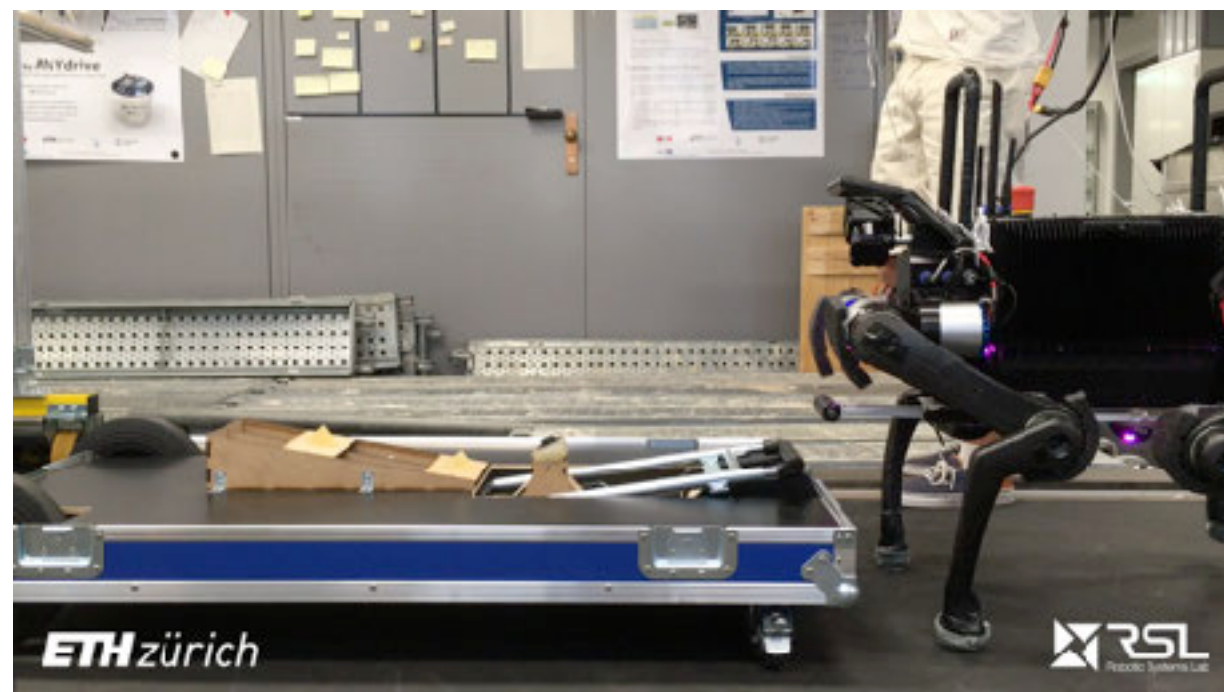


Locomotion

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



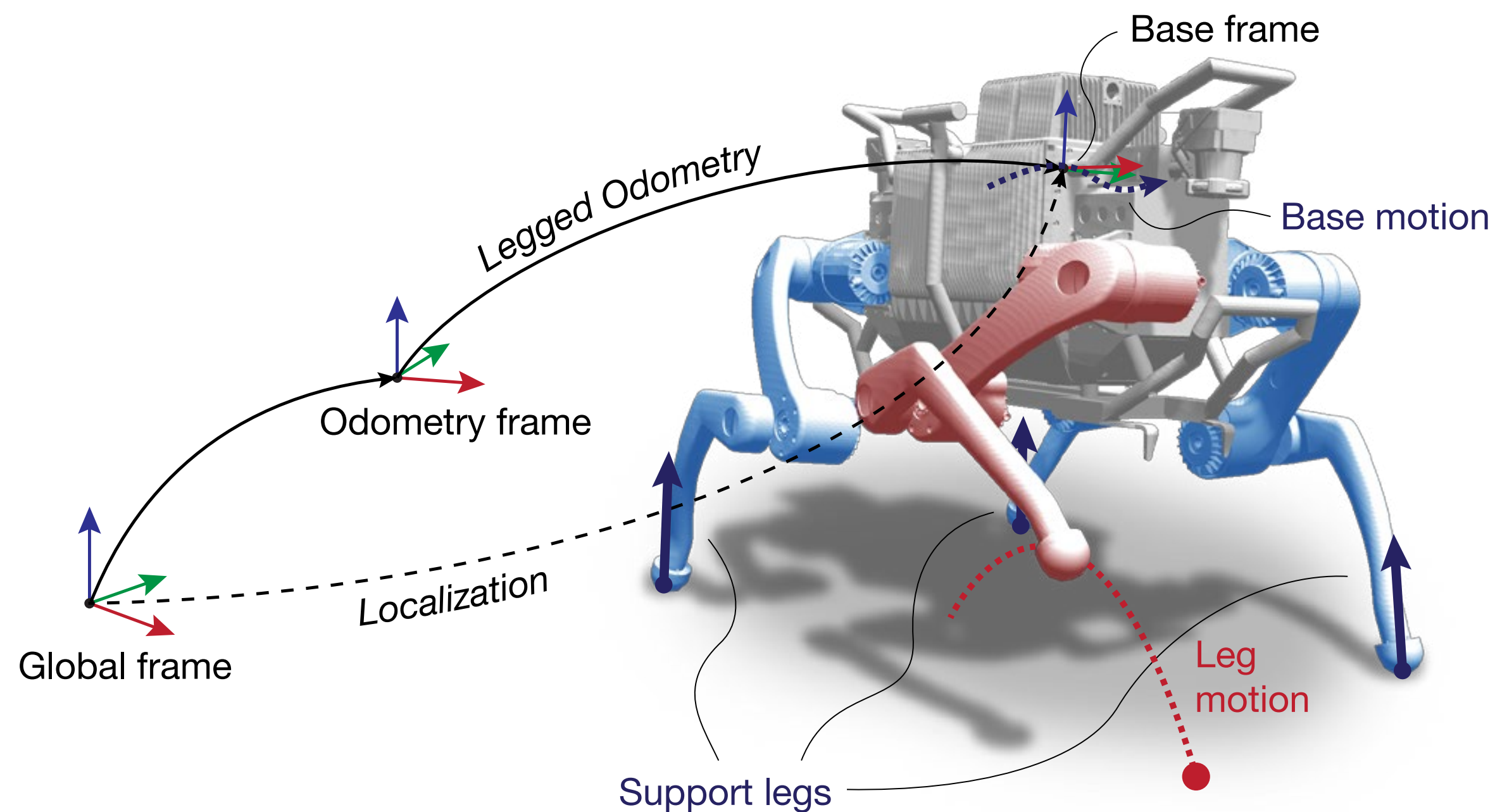
- Abstraction Layer for Whole-Body Motions (Free Gait API)



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.



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.



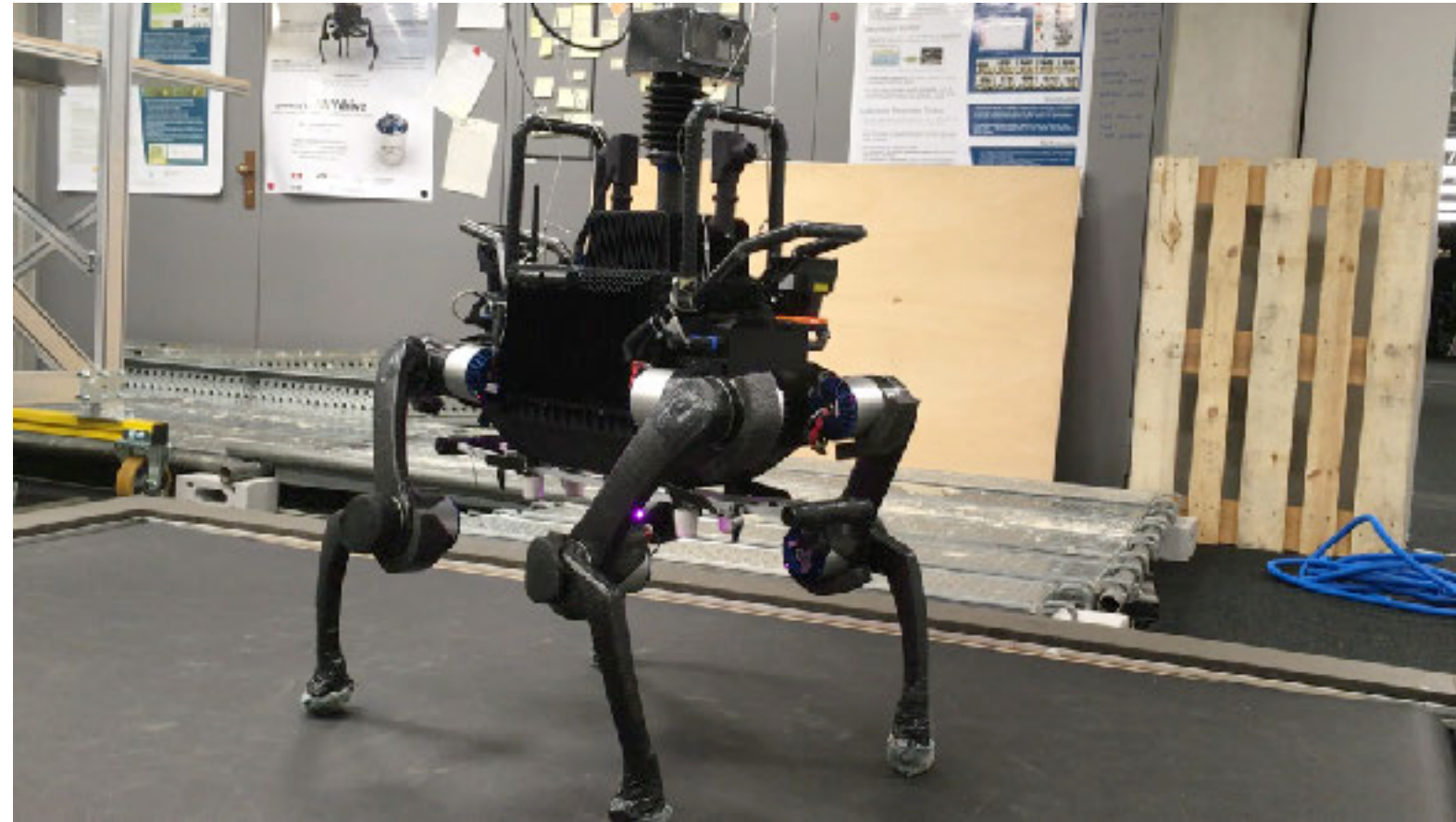
Locomotion

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

```

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.

Locomotion

Kindr – Kinematics and Dynamics for Robotics



- C++ library for the consistent handling of 3d position and rotations
- Support for *rotation matrices, quaternions, angle-axis, rotation vectors, Euler angles, etc.*
- Support for all common operations and includes time-derivates
- ROS interface available
- Based on Eigen, 1000+ unit tests

M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nuetzi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter, R. Siegwart, "A Primer on the Differential Calculus of 3D Orientations," in arXiv:1606.05285, 2016.

Kindr Library – Kinematics and Dynamics for Robotics
 Christian Gehring, C. Dario Bellicoso, Michael Bloesch, Hannes Sommer, Peter Fankhauser, Marco Hutter, Roland Siegwart

Nomenclature		
(Hyper-)complex number	Q	normal capital letter
Column vector	\mathbf{a}	bold small letter
Matrix	\mathbf{M}	bold capital letter
Identity matrix	$\mathbf{I}_{n \times m}$	$n \times m$ -matrix
Coordinate system (CS)	$\mathbf{e}_x^A, \mathbf{e}_y^A, \mathbf{e}_z^A$	Cartesian right-hand system A with basis (unit) vectors \mathbf{e}
Inertial frame	$\mathbf{e}_x^I, \mathbf{e}_y^I, \mathbf{e}_z^I$	global / inertial / world coordinate system (never moves)
Body-fixed frame	$\mathbf{e}_x^B, \mathbf{e}_y^B, \mathbf{e}_z^B$	local / body-fixed coordinate system (moves with body)
Rotation	$\Phi \in \text{SO}(3)$	generic rotation (for all parameterizations)
Machine precision	ϵ	

Operators		
Cross product/skew/unskew	$\mathbf{a} \times \mathbf{b} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = (\mathbf{a})^\wedge \mathbf{b} = \mathbf{a}\mathbf{b} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$	
Euclidean norm	$\ \mathbf{a}\ = \sqrt{\mathbf{a}^T \mathbf{a}} = \sqrt{a_1^2 + \dots + a_n^2}$	
Exponential map for matrix	$\exp: \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^{3 \times 3}, \mathbf{A} \mapsto e^{\mathbf{A}}, \mathbf{A} \in \mathbb{R}^{3 \times 3}$	
Logarithmic map for matrix	$\log: \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^{3 \times 3}, \mathbf{A} \mapsto \log \mathbf{A}, \mathbf{A} \in \mathbb{R}^{3 \times 3}$	

Position & Orientation		
Position		
Vector	$\mathbf{r}_{OP} \in \mathbb{R}^3$	from point O to point P
Position vector	${}^B \mathbf{r}_{OP} \in \mathbb{R}^3$	from point O to point P expr. in frame B
Homogeneous pos. vector	${}^B \mathbf{p}_{OP} = [{}^B \mathbf{r}_{OP}^T \ 1]^T$	from point O to point P expr. in frame B
Orientation/Rotation		
1) Active Rotation:	$\Phi^A: {}^I \mathbf{r}_{OQ} \mapsto {}^I \mathbf{r}_{OQ}$	(rotates the vector \mathbf{r}_{OQ})
2) Passive Rotation:	$\Phi^P: {}^I \mathbf{r}_{OP} \mapsto {}^B \mathbf{r}_{OP}$	(rotates the frame $(\mathbf{e}_x^I, \mathbf{e}_y^I, \mathbf{e}_z^I)$)
3) Elementary Rotations	${}^I \mathbf{r}_{OP} = \mathbf{C}_{IB} {}^B \mathbf{r}_{OP}$	
around z-axis:	$\mathbf{C}_{IB} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
around y-axis:	$\mathbf{C}_{IB} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$	
around x-axis:	$\mathbf{C}_{IB} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$	
4) Inversion:	$\Phi^{A^{-1}}(\mathbf{r}) = \Phi^P(\mathbf{r})$	
5) Concatenation:	$\Phi_2^P(\Phi_1^P(\mathbf{r})) = (\Phi_2^P \otimes \Phi_1^P)(\mathbf{r}) = (\Phi_1^{A^{-1}} \otimes \Phi_2^{A^{-1}})^{-1}(\mathbf{r})$	
6) Exponential map:	$\exp: \mathbb{R}^3 \rightarrow \text{SO}(3), \mathbf{v} \mapsto \exp(\Phi^V), \mathbf{v} \in \mathbb{R}^3$	
7) Logarithmic map:	$\log: \text{SO}(3) \rightarrow \mathbb{R}^3, \Phi \mapsto \log(\Phi^V), \Phi \in \text{SO}(3)$	
8) Box plus:	$\Phi_2 \boxplus \mathbf{v} = \exp(\mathbf{v}) \otimes \Phi_1, \Phi_1, \Phi_2 \in \text{SO}(3), \mathbf{v} \in \mathbb{R}^3$	
9) Box minus:	$\mathbf{v} = \Phi_1 \boxminus \Phi_2 = \log(\Phi_1 \otimes \Phi_2^{-1}), \Phi_1, \Phi_2 \in \text{SO}(3), \mathbf{v} \in \mathbb{R}^3$	
10) Discrete integration:	$\Phi_{IB}^{k+1} = \Phi_{IB}^k \boxplus ({}^I \omega_{IB}^k \Delta t), \Phi_{IB}^{k+1} = \Phi_{IB}^k \boxminus (-{}^B \omega_{IB}^k \Delta t)$	
11) Discrete differential:	${}^I \omega_{IB}^k = \Phi_{IB}^{k+1} \boxminus \Phi_{IB}^k / \Delta t, {}^B \omega_{IB}^k = -(\Phi_{IB}^{k+1} \boxminus \Phi_{IB}^k) / \Delta t$	
12) (Spherical) linear interpolation $t \in [0, 1]$:	$\Phi_t = \Phi_0 \boxplus ((\Phi_1 \boxminus \Phi_0)t), \Phi_t = \Phi(t), \Phi_0 = \Phi(0), \Phi_1 = \Phi(1)$	

Rotation Parameterizations		
Rotation Matrix	$\mathbf{C}_{IB} \in \text{SO}(3)$	The rotation matrix (Direction Cosine Matrix) is a coordinate transformation matrix, which transforms vectors from frame B to frame I .
Quaternion	$\mathbf{q}_{IB} = [q_0 \ q_1 \ q_2 \ q_3]^T$	Hamiltonian unit quaternion (hypercomplex number) $Q = q_0 + q_1 i + q_2 j + q_3 k \in \mathbb{H}, q_i \in \mathbb{R}, \ Q\ = 1$
Angle-axis	$(\theta, \mathbf{n})_{IB}$	Rotation with unit rotation axis \mathbf{n} and angle $\theta \in [0, \pi]$.
Rotation Vector	ϕ_{IB}	Rotation with rotation axis $\mathbf{n} = \frac{\phi}{\ \phi\ }$ and angle $\theta = \ \phi\ $.
Euler Angles ZYX	$[z, y, x]_{IB}$	Tait-Bryan angles (Flight conv.): $z - y' - x''$, i.e. yaw-pitch-roll. Singularities are at $y = \pm \frac{\pi}{2}$.
Euler Angles YPR		$z \in [-\pi, \pi], y \in [-\frac{\pi}{2}, \frac{\pi}{2}], x \in [-\pi, \pi]$
Euler Angles XYZ	$[x, y, z]_{IB}$	Cardan angles: $x - y' - z''$, i.e. roll-pitch-yaw. Singularities are at $y = \pm \frac{\pi}{2}$.
Euler Angles RPY		$x \in [-\pi, \pi], y \in [-\frac{\pi}{2}, \frac{\pi}{2}], z \in [-\pi, \pi]$

Rotation Quaternion
 A rotation quaternion is a Hamiltonian unit quaternion:
 $Q = q_0 + q_1 i + q_2 j + q_3 k \in \mathbb{H}, q_i \in \mathbb{R}, i^2 = j^2 = k^2 = ijk = -1, \|Q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$
 Tuple: $Q = (q_0, q_1, q_2, q_3) = (q_0, \mathbf{q})$ with $\mathbf{q} := (q_1, q_2, q_3)^T$
 4 x 1-vector: $\mathbf{q} = [q_0 \ q_1 \ q_2 \ q_3]^T$
 Conjugate: $Q^* = (q_0, -\mathbf{q})$
 Inverse: $Q^{-1} = Q^* = (q_0, -\mathbf{q})$
 Quaternion multiplication:
 $Q \cdot P = (q_0, \mathbf{q}) \cdot (p_0, \mathbf{p}) = (q_0 p_0 - \mathbf{q}^T \mathbf{p}, q_0 \mathbf{p} + p_0 \mathbf{q} + \mathbf{q} \times \mathbf{p})$

$\mathbf{q} \otimes \mathbf{p} = \mathbf{Q}(\mathbf{q}) \mathbf{p} = \begin{pmatrix} q_0 & -\mathbf{q}^T \\ \mathbf{q} & q_0 \mathbf{I}_{3 \times 3} + \mathbf{q} \mathbf{q}^T \end{pmatrix} \begin{pmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} q_0 p_0 - q_1 p_1 - q_2 p_2 - q_3 p_3 \\ q_1 p_0 + q_0 p_1 + q_2 p_3 - q_3 p_2 \\ q_2 p_0 - q_1 p_2 + q_0 p_3 + q_3 p_1 \\ q_3 p_0 + q_1 p_2 - q_2 p_1 + q_0 p_3 \end{pmatrix}$

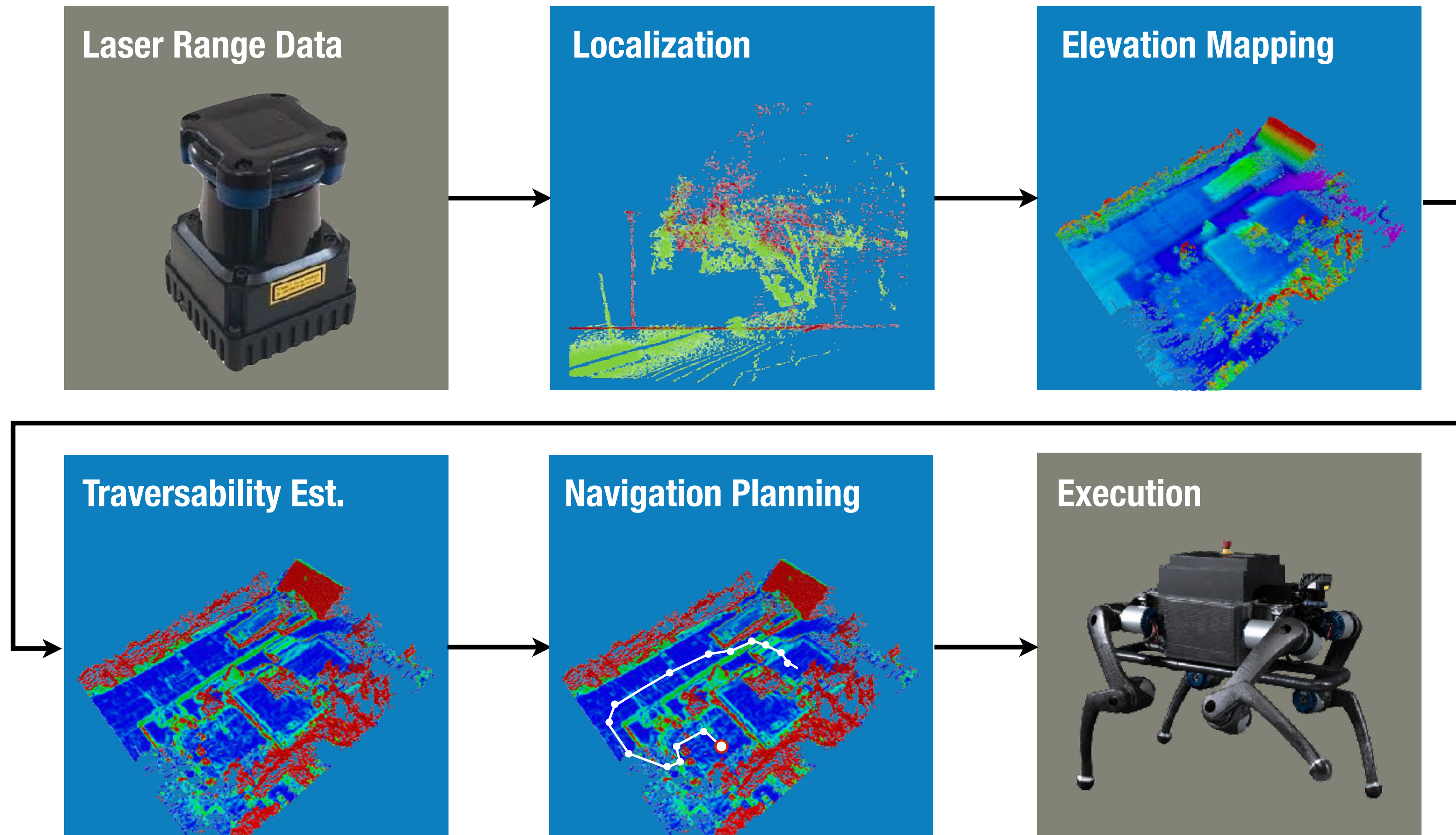
Note that \mathbf{Q}_{IB} and $-\mathbf{Q}_{IB}$ represent the same rotation, but not the same unit quaternion.

Rotation Quaternion \Leftrightarrow Rotation Vector
 $\mathbf{q}_{IB} = \begin{cases} \left[\cos(\frac{1}{2}\|\phi\|), \frac{\phi^T}{\|\phi\|} \sin(\frac{1}{2}\|\phi\|) \right]^T & \text{if } \|\phi\| \geq \epsilon \\ \left[1, \frac{1}{2}\phi^T \right]^T & \text{otherwise} \end{cases} \Leftrightarrow \mathbf{q}_{IB} = \begin{cases} \left[2 \operatorname{atan2}(\|\mathbf{q}\|, q_0), \frac{\mathbf{q}}{\|\mathbf{q}\|} \right]^T & \text{if } \|\mathbf{q}\| \geq \epsilon \\ \left[1, \frac{1}{2}\phi^T \right]^T & \text{otherwise} \end{cases}$

Rotation Quaternion \Leftrightarrow Angle-Axis
 $\mathbf{q}_{IB} = \begin{bmatrix} \cos \frac{\theta}{2} \\ \mathbf{n} \sin \frac{\theta}{2} \end{bmatrix} \Leftrightarrow (\theta, \mathbf{n})_{IB} = \begin{cases} (2 \arccos(q_0), \frac{\mathbf{q}}{\|\mathbf{q}\|}) & \text{if } \|\mathbf{q}\| \geq \epsilon \\ (0, [1 \ 0 \ 0]^T) & \text{otherwise} \end{cases}$

Rotation Quaternion \Leftrightarrow Rotation Matrix
 $\mathbf{C}_{IB} = \mathbf{I}_{3 \times 3} + 2q_0 \hat{\mathbf{q}} + 2\hat{\mathbf{q}}^2 = (2q_0^2 - 1)\mathbf{I}_{3 \times 3} + 2q_0 \hat{\mathbf{q}} + 2\hat{\mathbf{q}}\hat{\mathbf{q}}^T$
 $= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1 q_2 - 2q_0 q_3 & 2q_0 q_2 + 2q_1 q_3 \\ 2q_0 q_3 + 2q_1 q_2 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_2 q_3 - 2q_0 q_1 \\ 2q_1 q_3 - 2q_0 q_2 & 2q_0 q_1 + 2q_2 q_3 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$
 $\mathbf{C}_{IB}^{-1} = \mathbf{C}_{BI} = \mathbf{I}_{3 \times 3} - 2q_0 \hat{\mathbf{q}} + 2\hat{\mathbf{q}}^2$
 $= \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_0 q_3 + 2q_1 q_2 & 2q_1 q_3 - 2q_0 q_2 \\ 2q_1 q_2 - 2q_0 q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_0 q_1 + 2q_2 q_3 \\ 2q_0 q_2 + 2q_1 q_3 & 2q_2 q_3 - 2q_0 q_1 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}$

Navigation

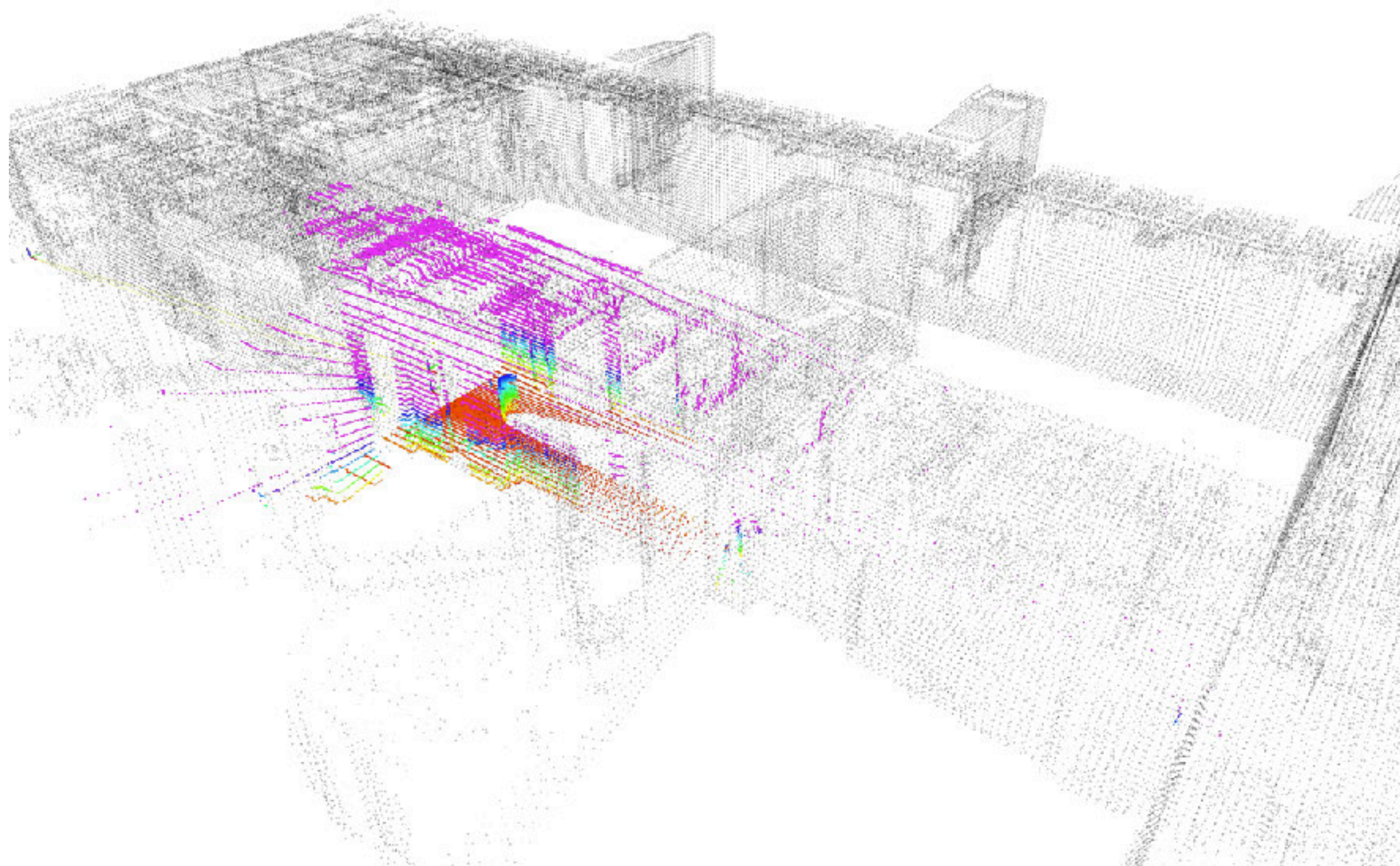


Navigation

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



Open Source

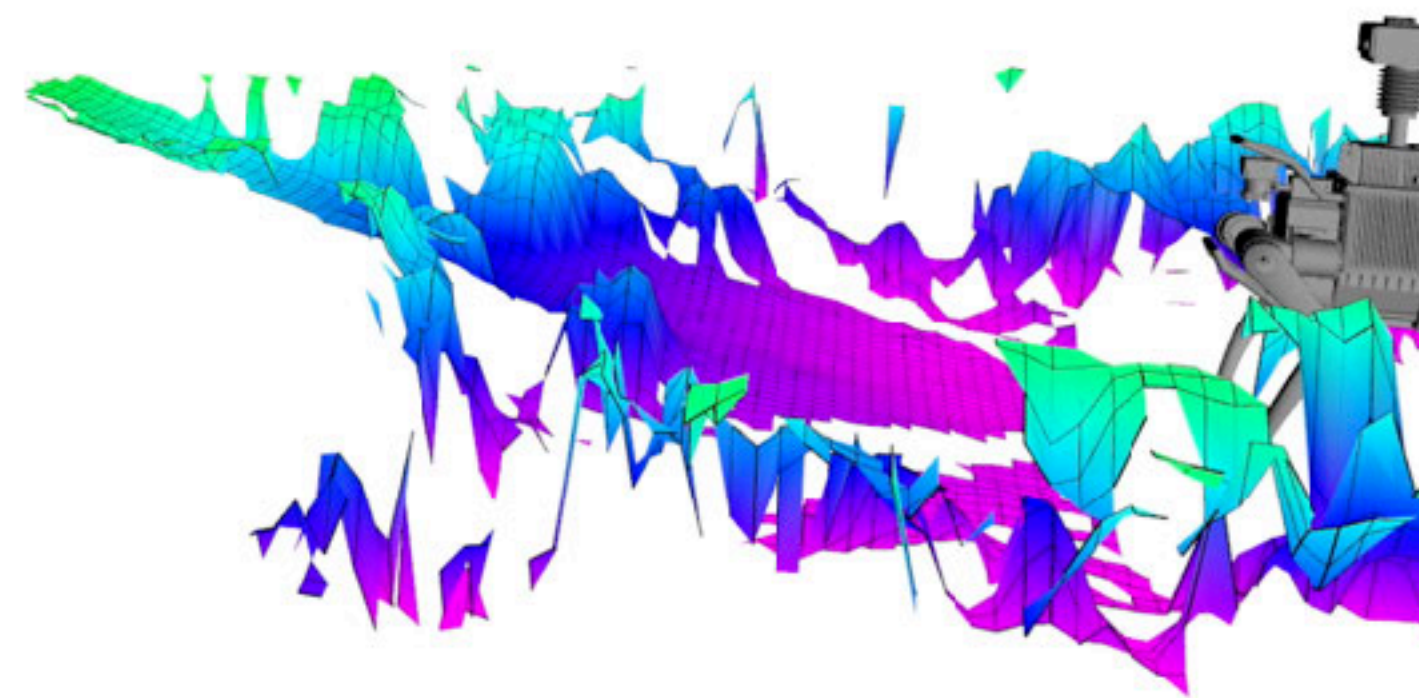
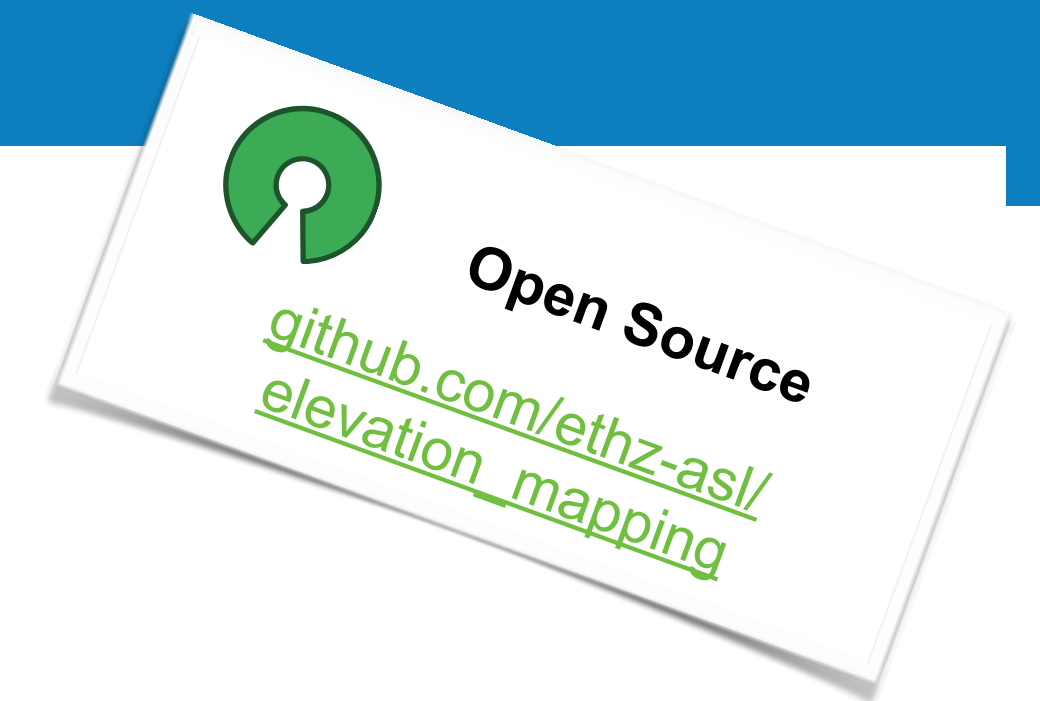
[github.com/ethz-asl/
ethzasl_icp_mapping](https://github.com/ethz-asl/ethzasl_icp_mapping)

- 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

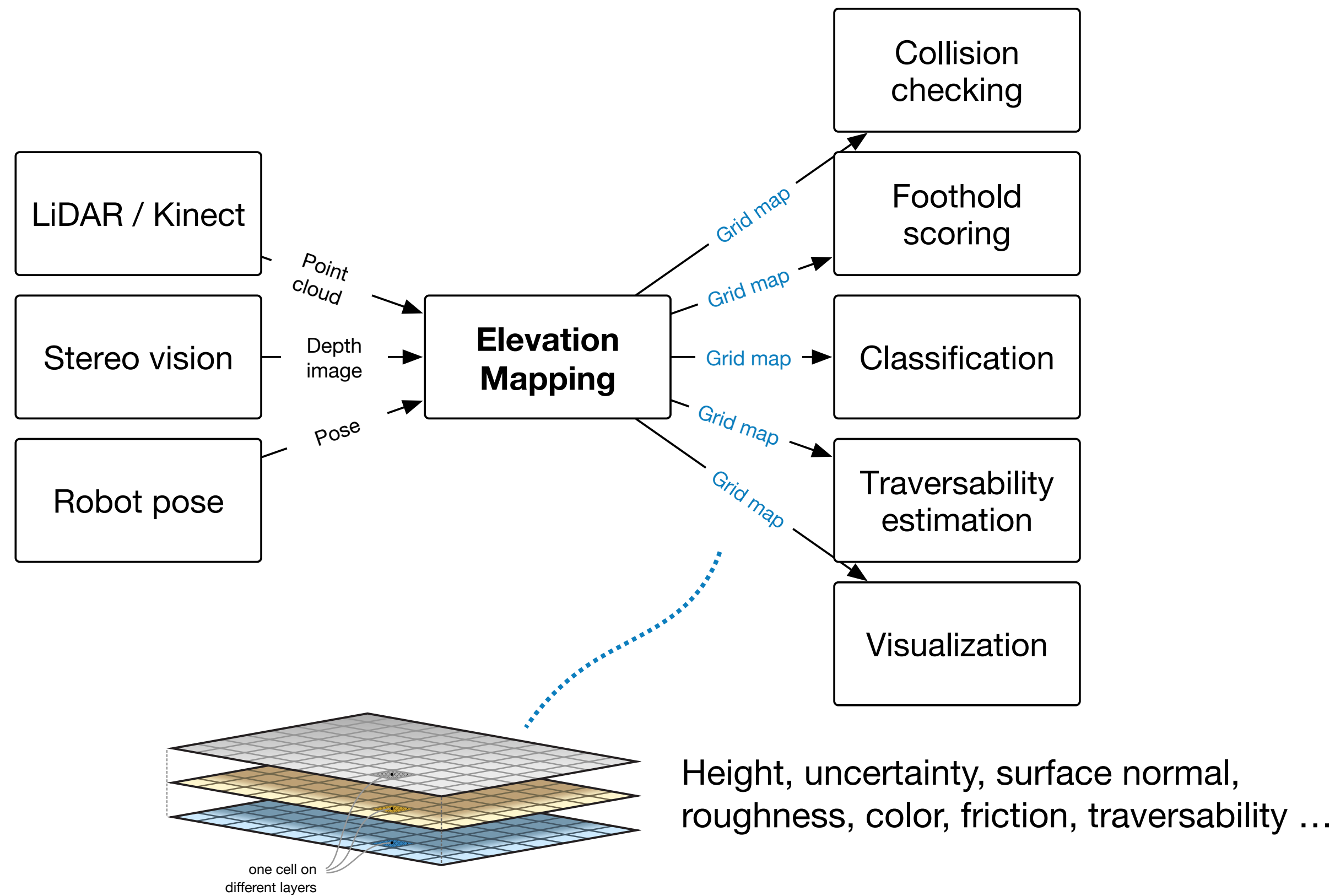
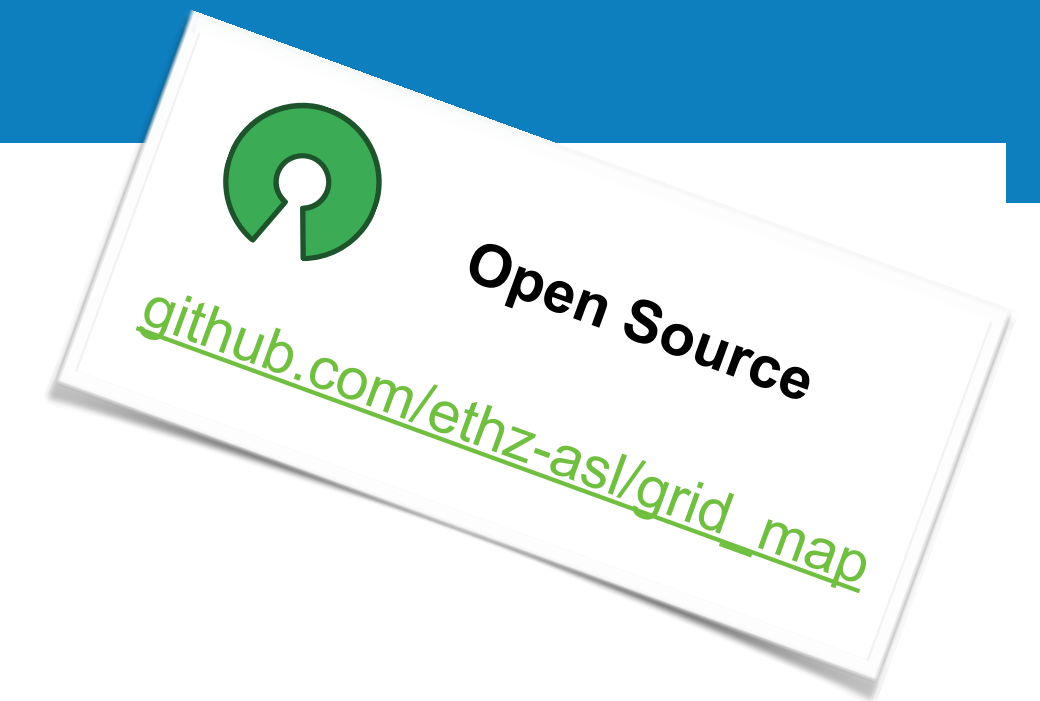
Elevation Mapping – Dense Terrain Mapping



- 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.

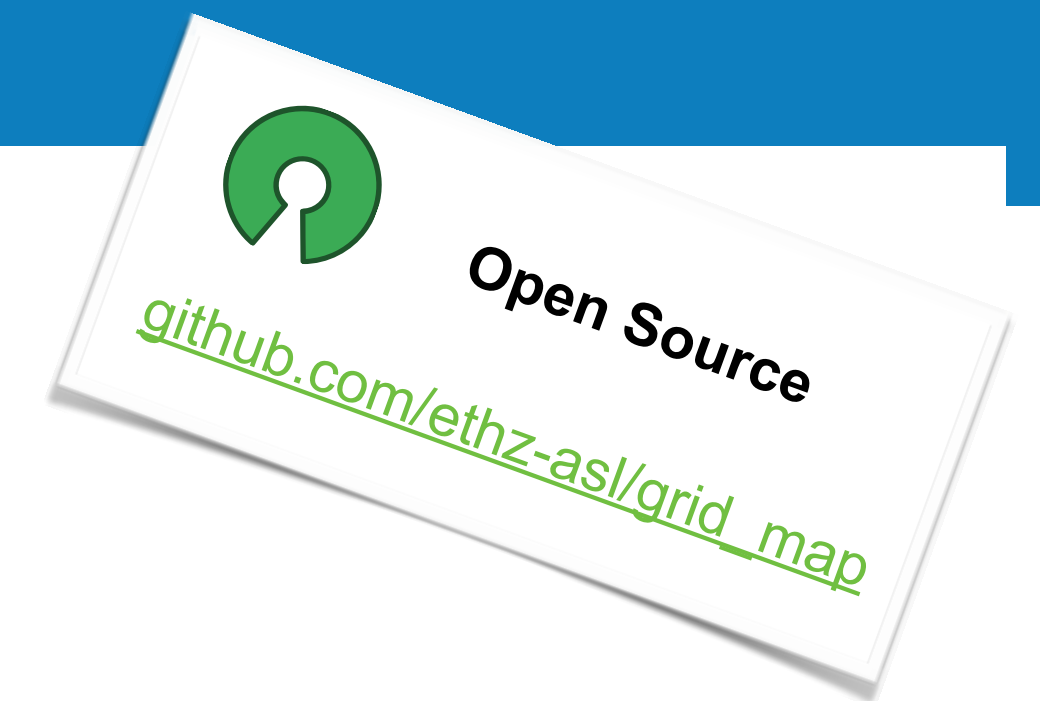
Navigation Grid Map – Universal Multi-Layer Grid Map Library



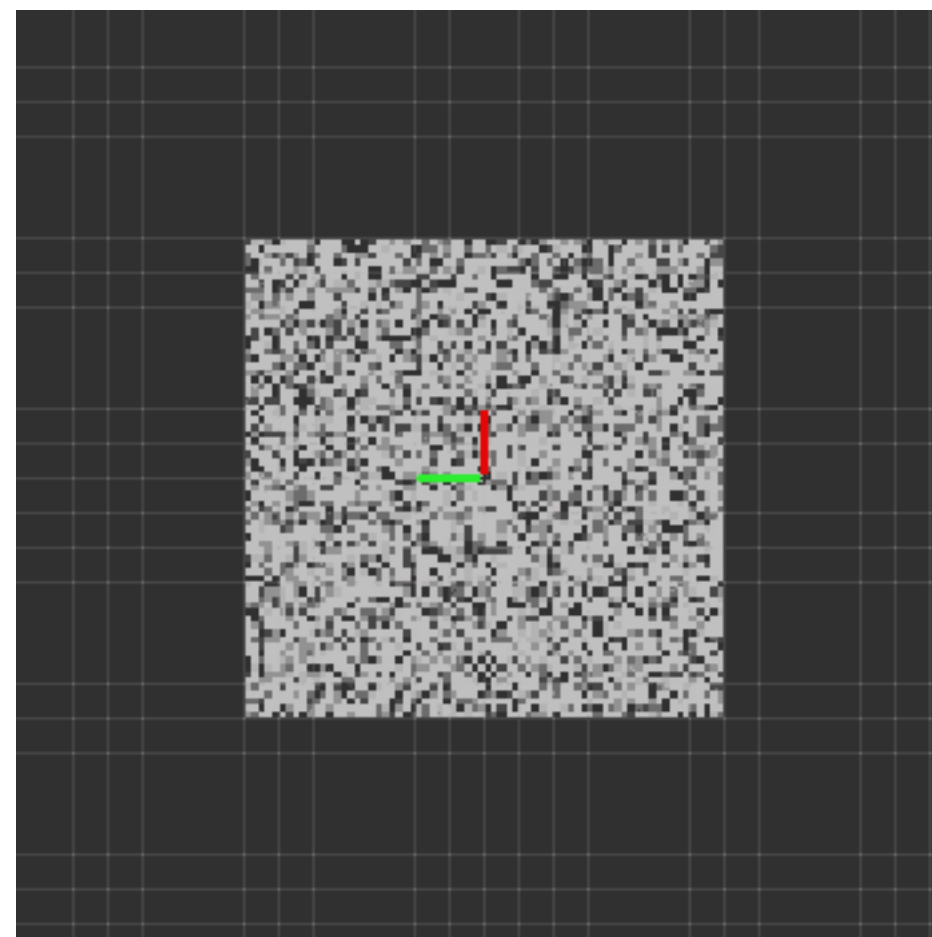
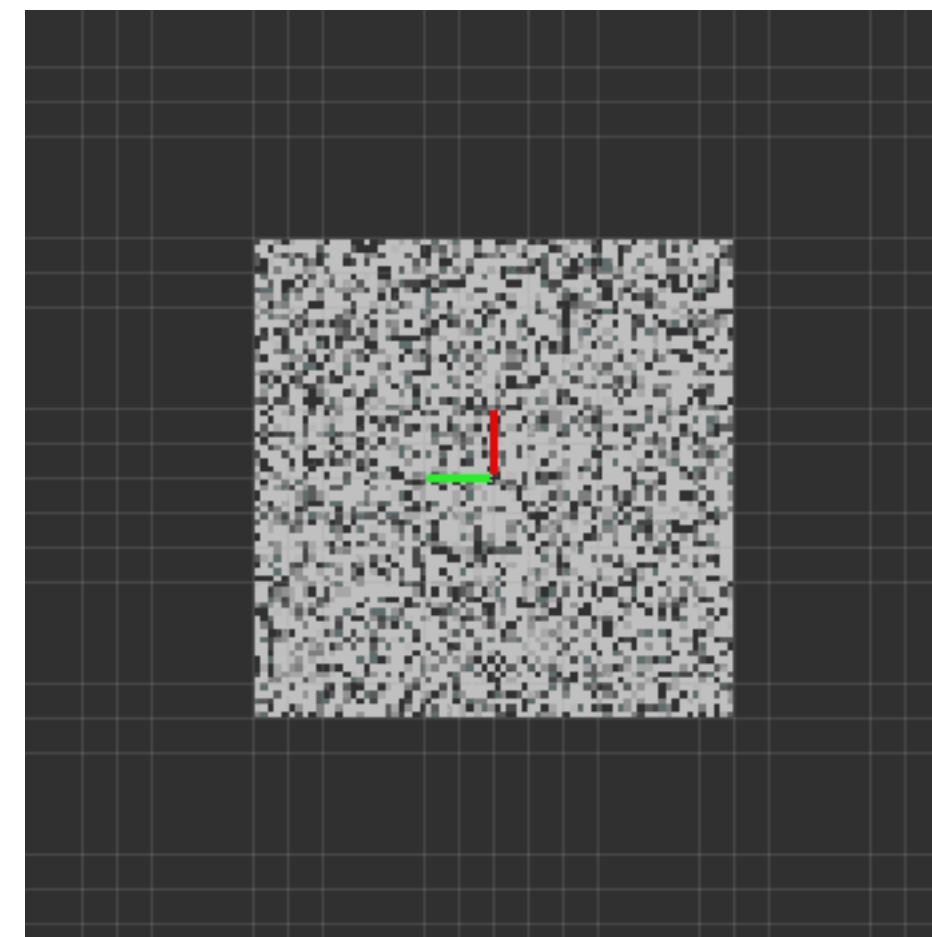
P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in Robot Operating System (ROS) - The Complete Reference, Springer, 2015.

Navigation

Grid Map – Universal Multi-Layer Grid Map Library



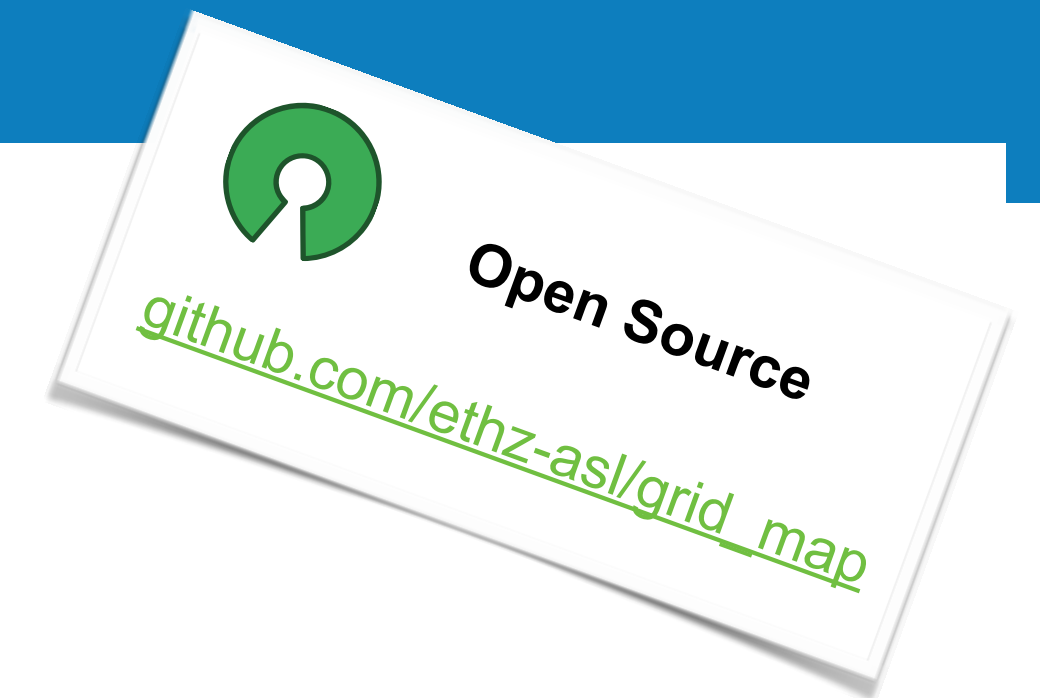
- 2D circular buffer data structure
 - ➔ Efficient map repositioning

`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



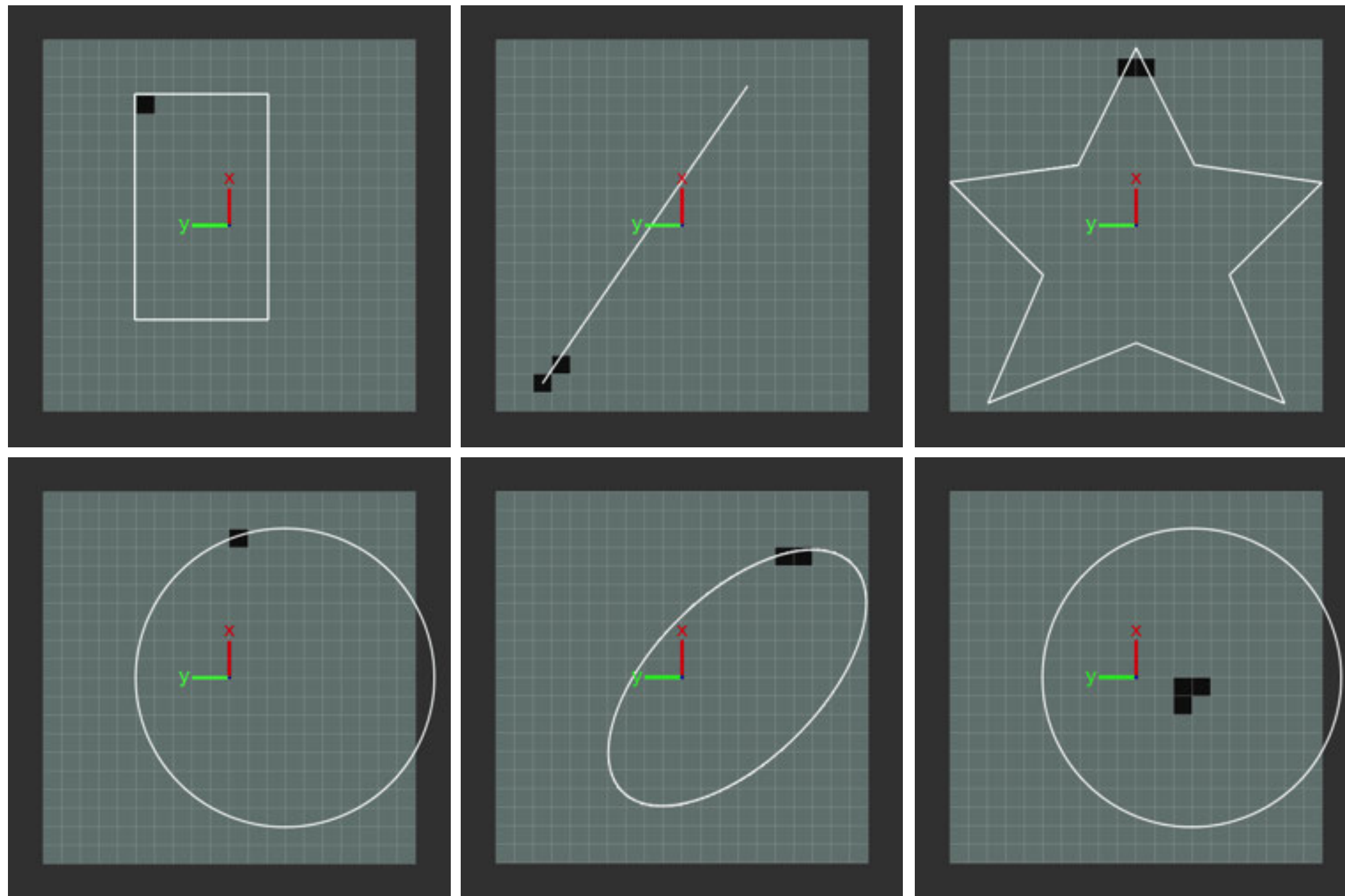
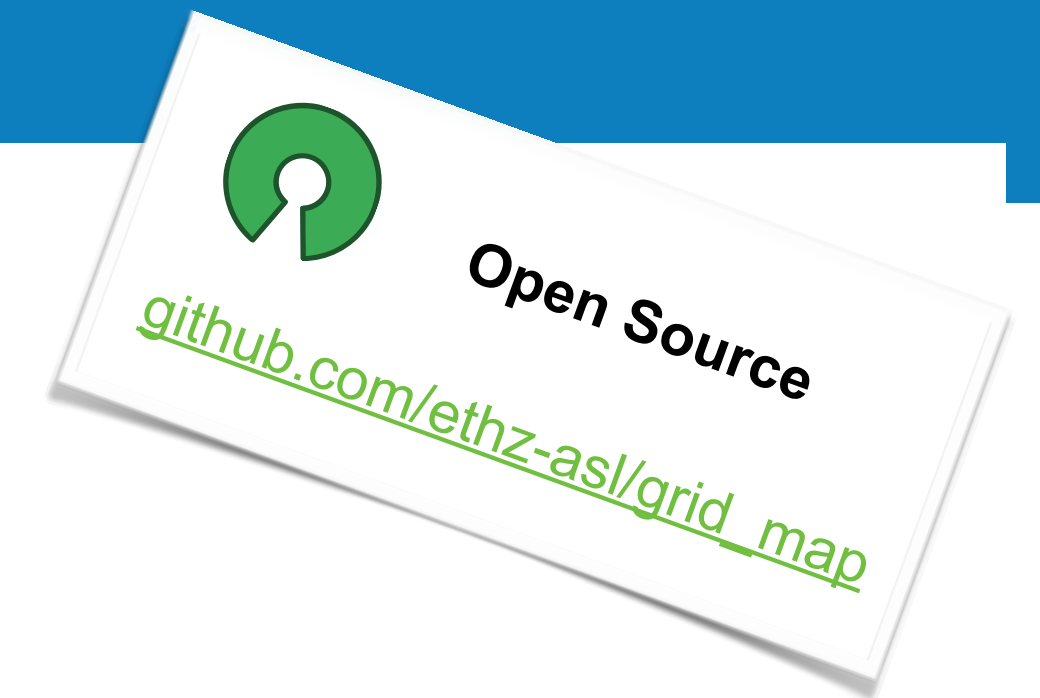
- 2D circular buffer data structure
 - ➔ Efficient map repositioning
- Based on Eigen (C++)
 - ➔ Versatile and efficient data manipulation

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

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

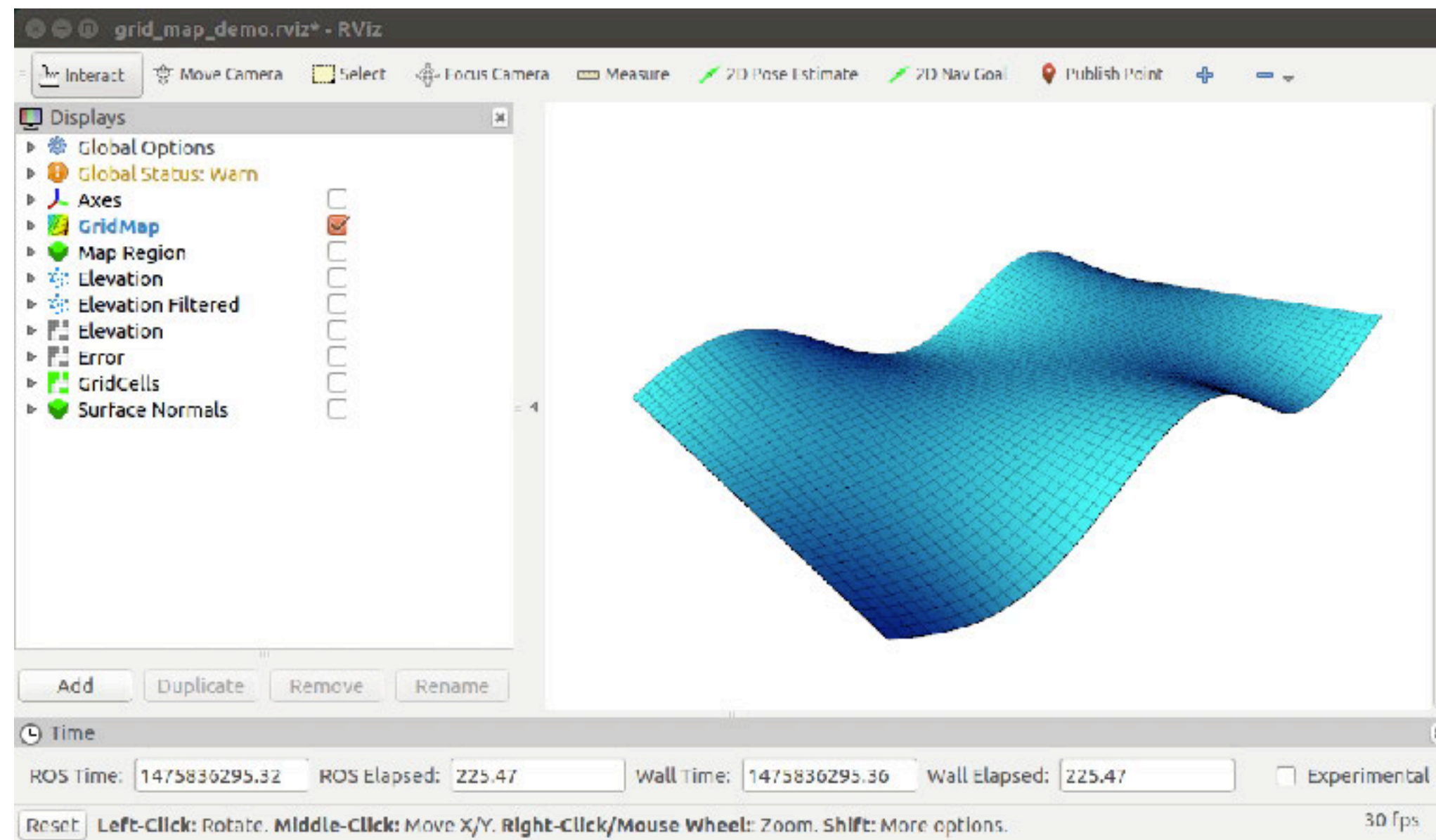
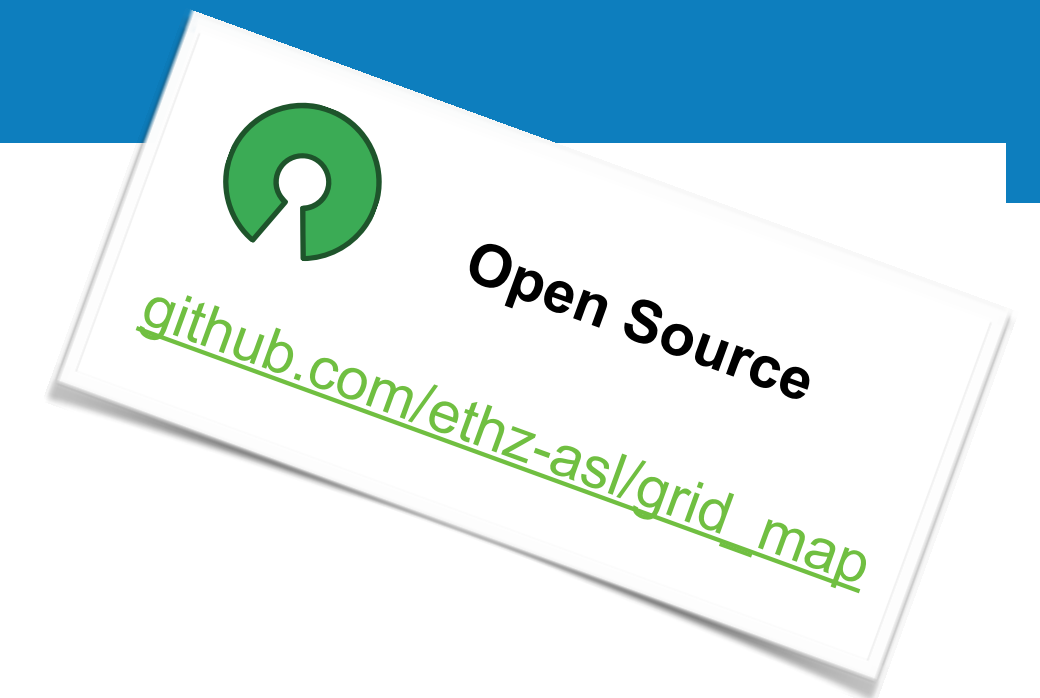


- 2D circular buffer data structure
 - ➔ Efficient map repositioning
- Based on Eigen (C++)
 - ➔ Versatile and efficient data manipulation
- Convenience functions
 - ➔ Iterators, math tools, etc.

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



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

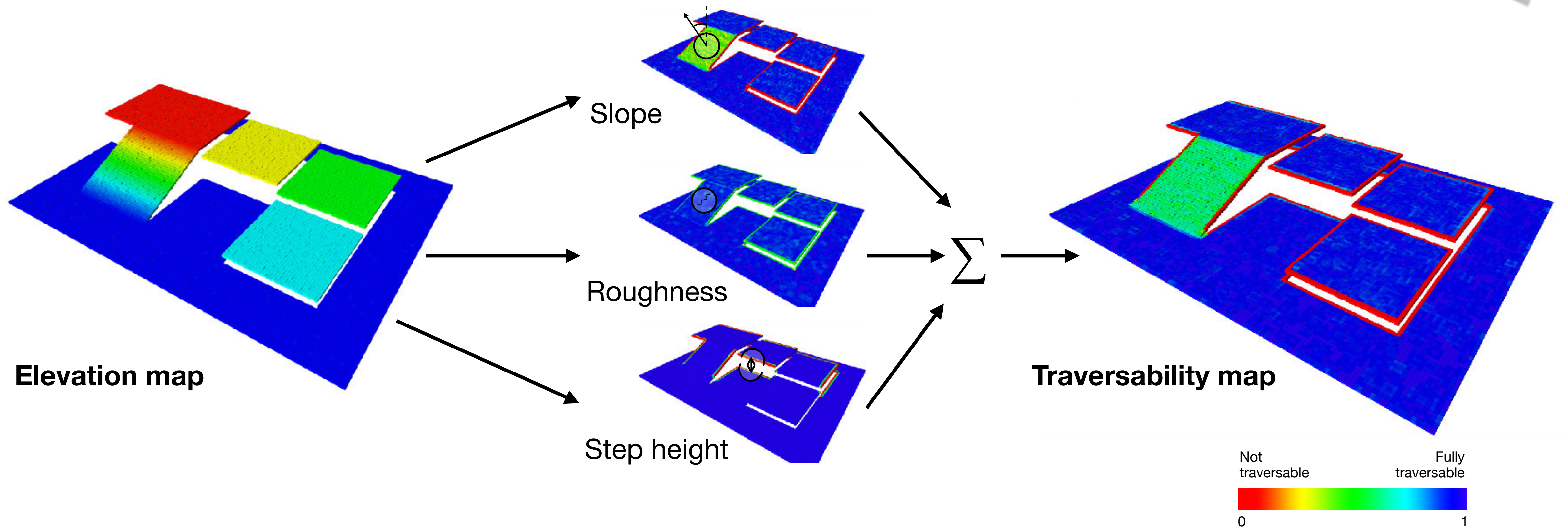
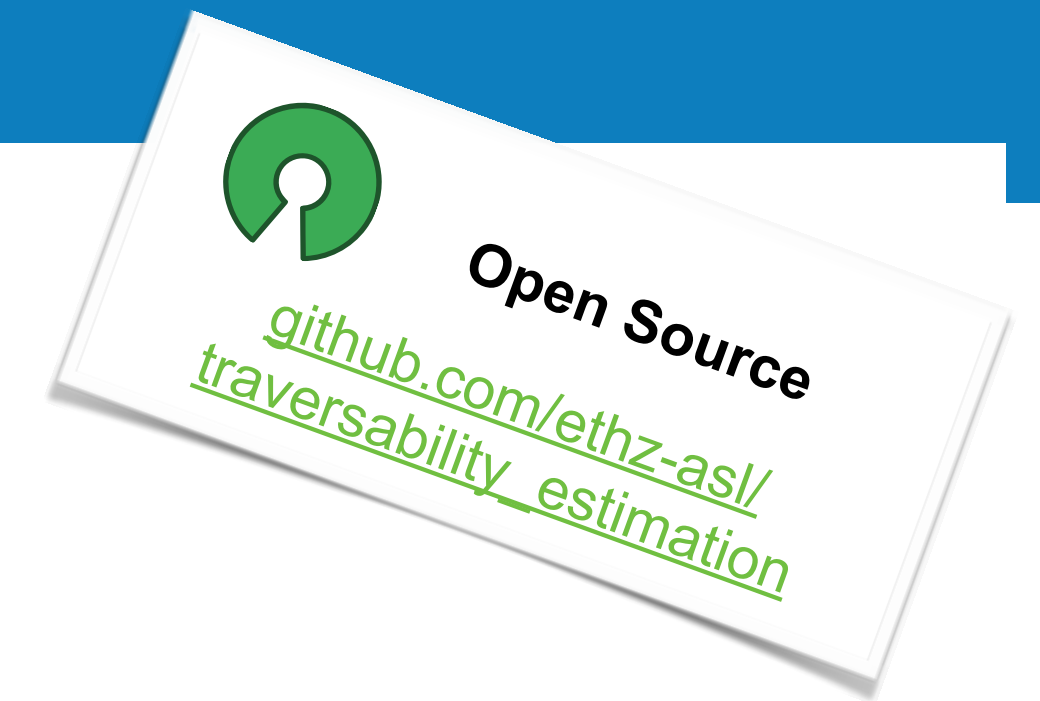
- Based on Eigen (C++)
 - ➔ Versatile and efficient data manipulation

- 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.

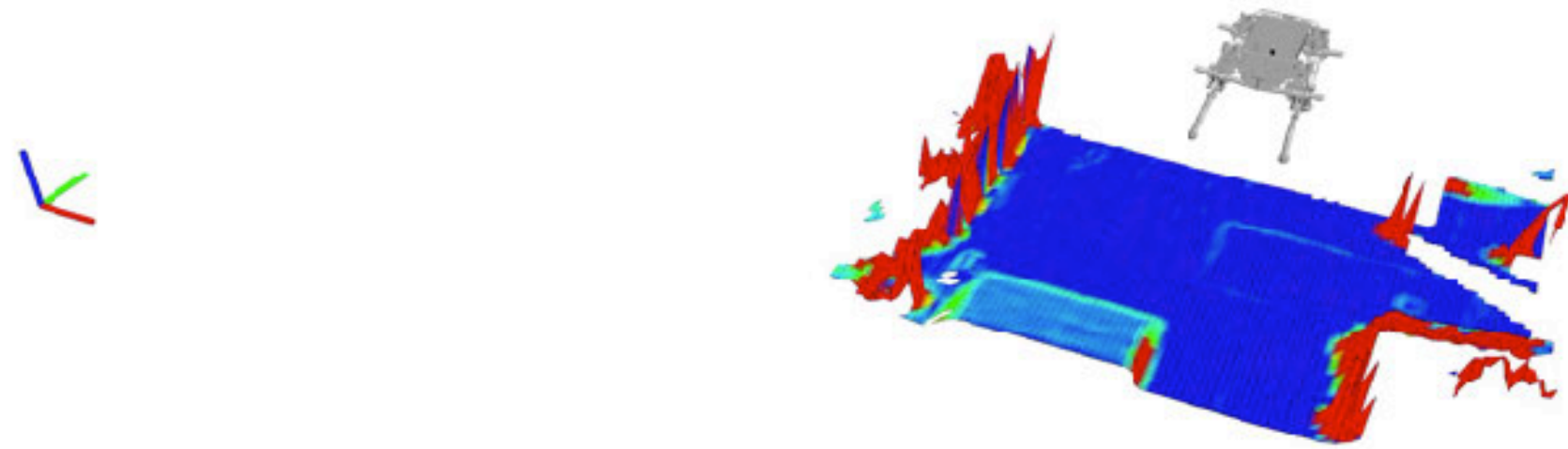
Navigation Traversability Estimation



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

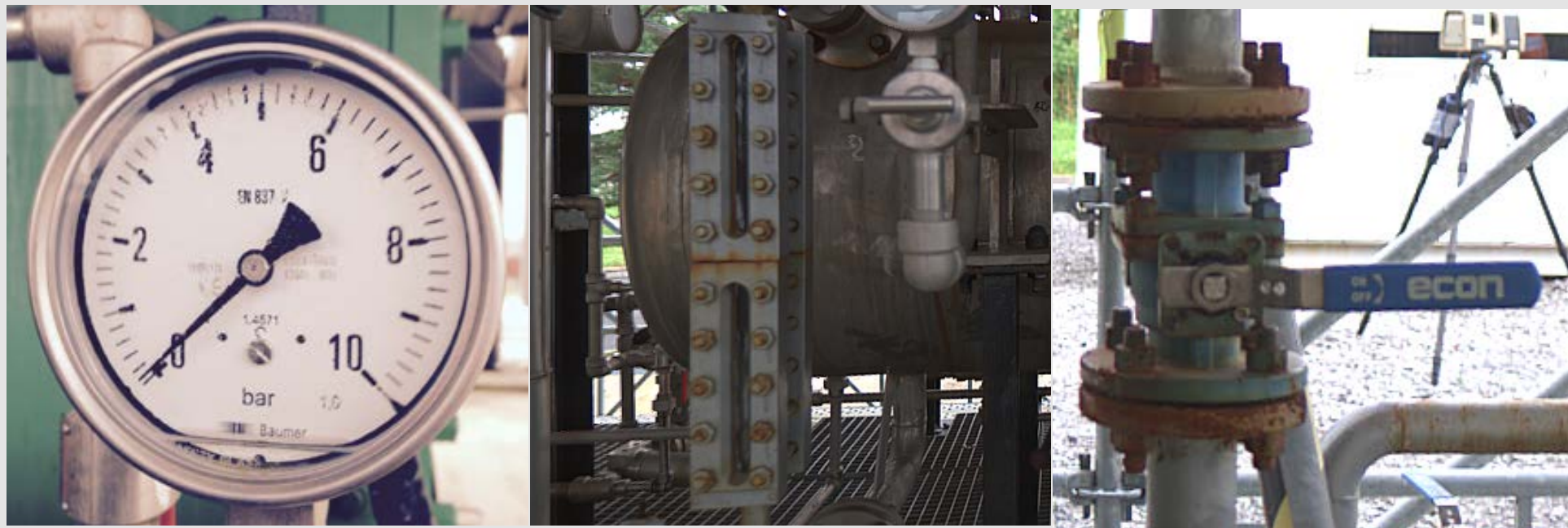


- Online navigation planning based on RRT* (OMPL)
- Works with and without initial map
- Continuous for changing environments

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.

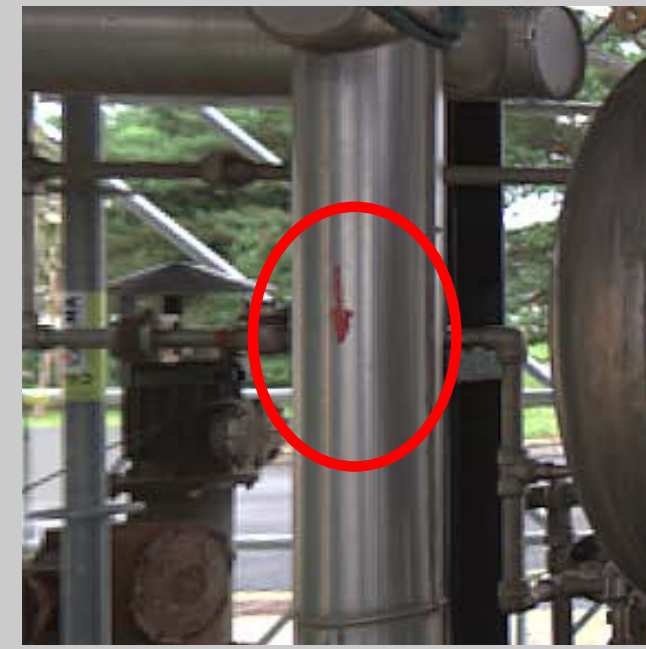
Inspection

Visual inspection



Pressure & Level gauges Valves

Thermal Inspection

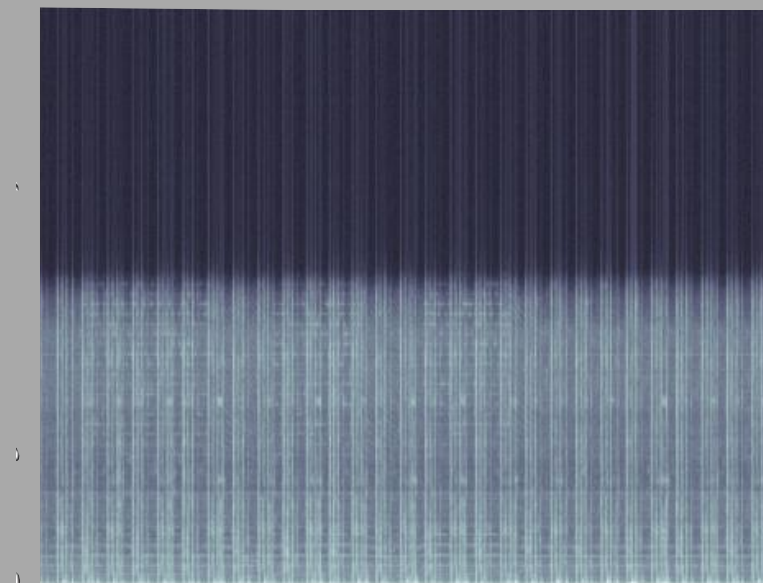
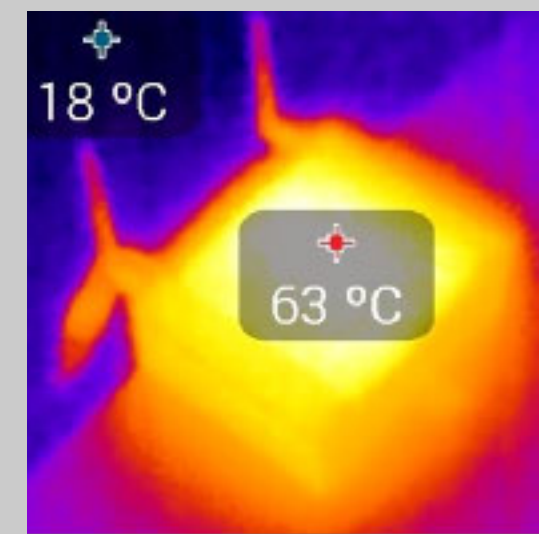
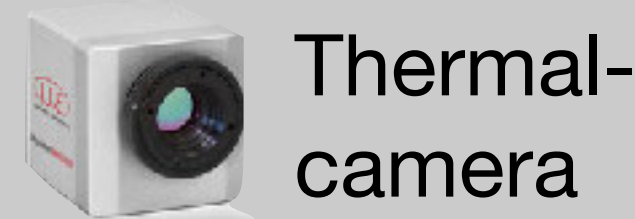
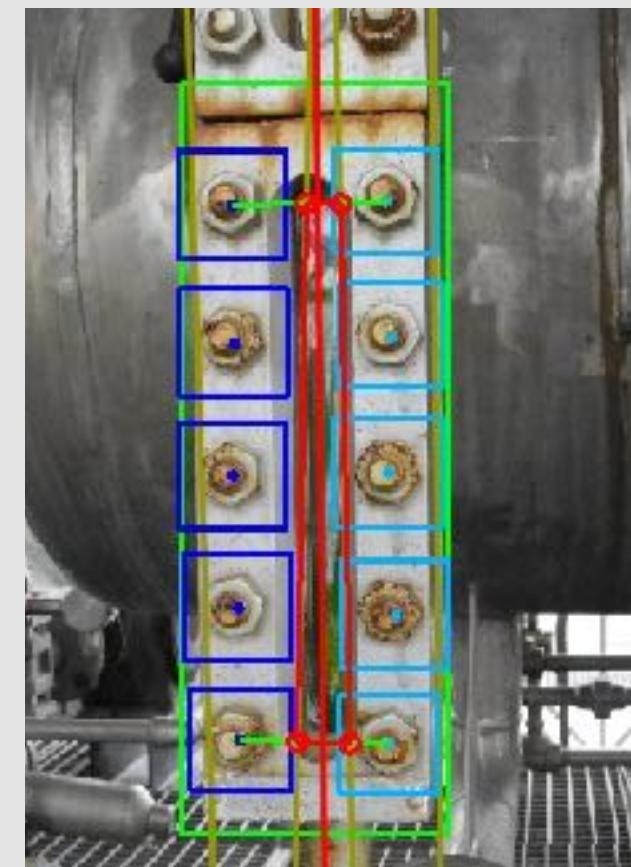


Thermal points

Auditive Inspection



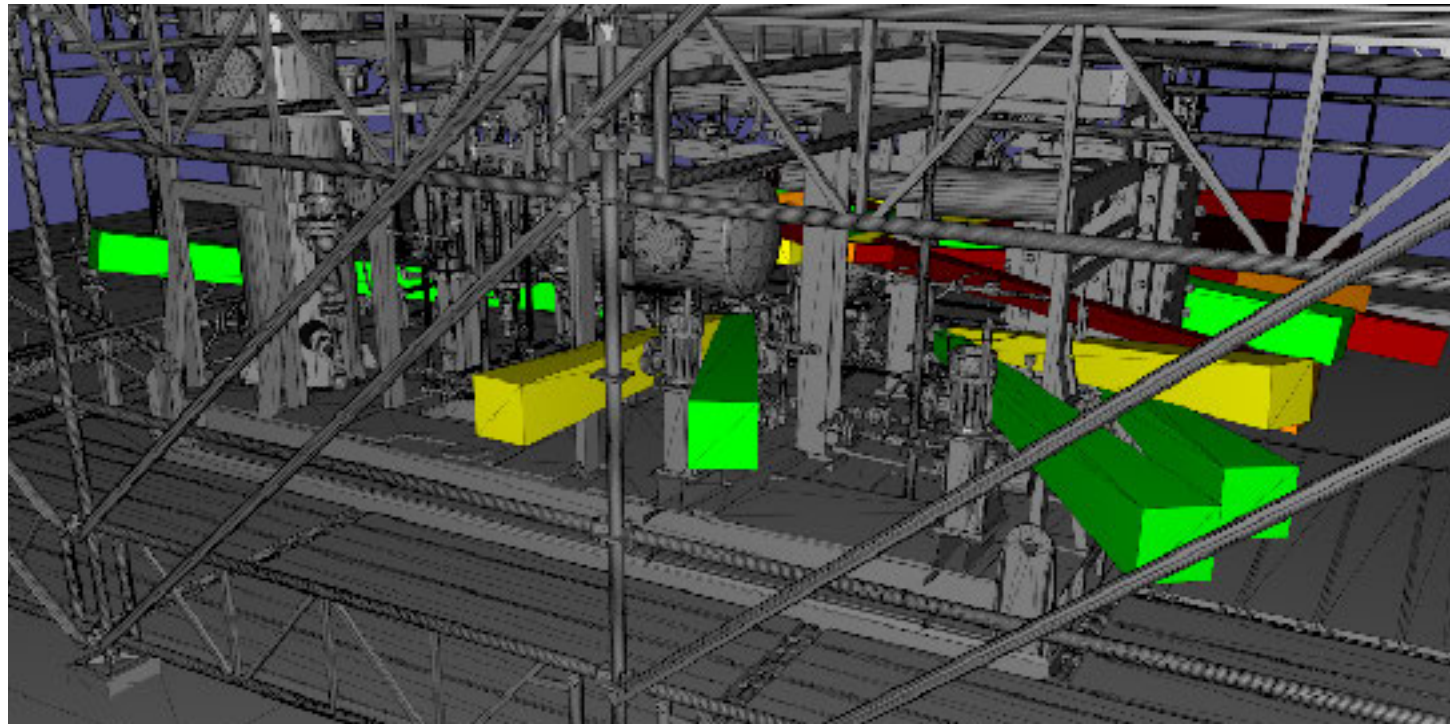
Pumps Gas leaks Platform alarm



Inspection

Visual Inspection of Pressure Gauges

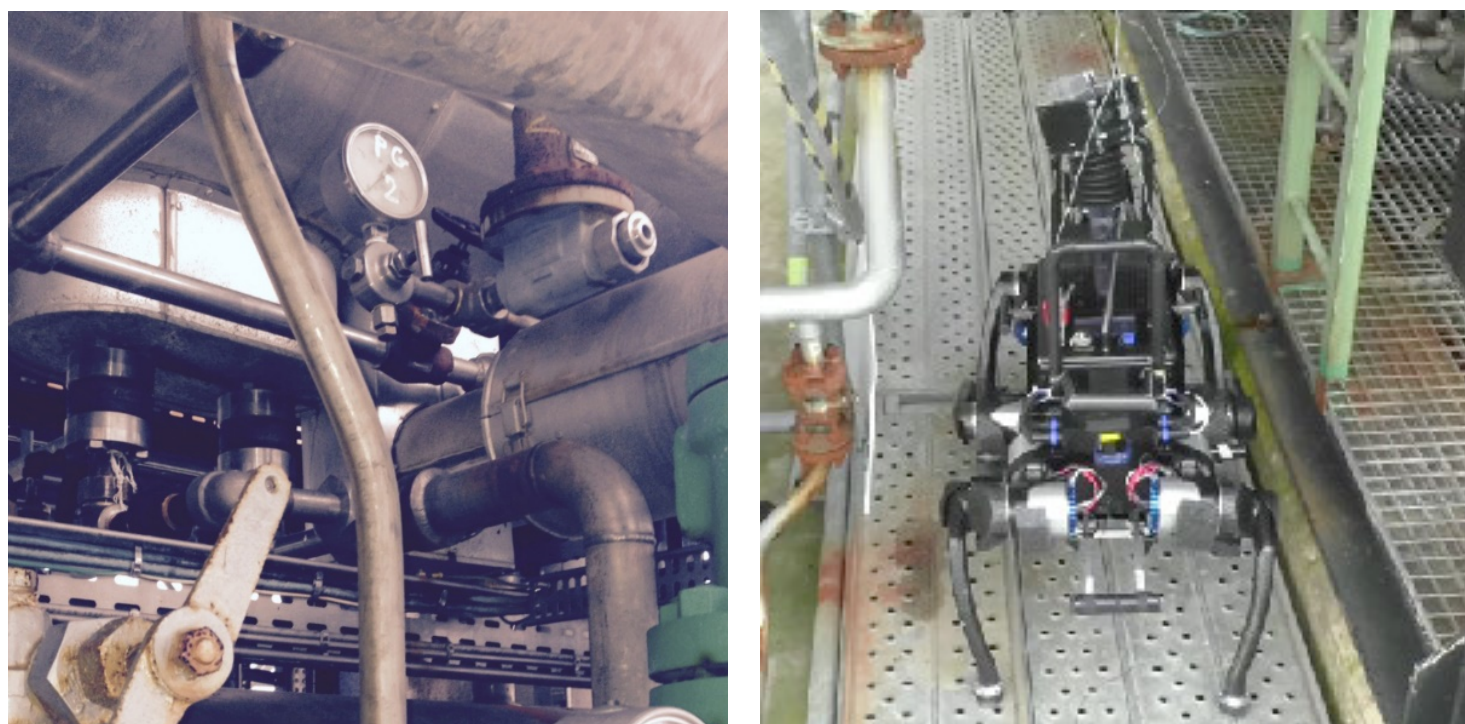
A Automatic view point generation



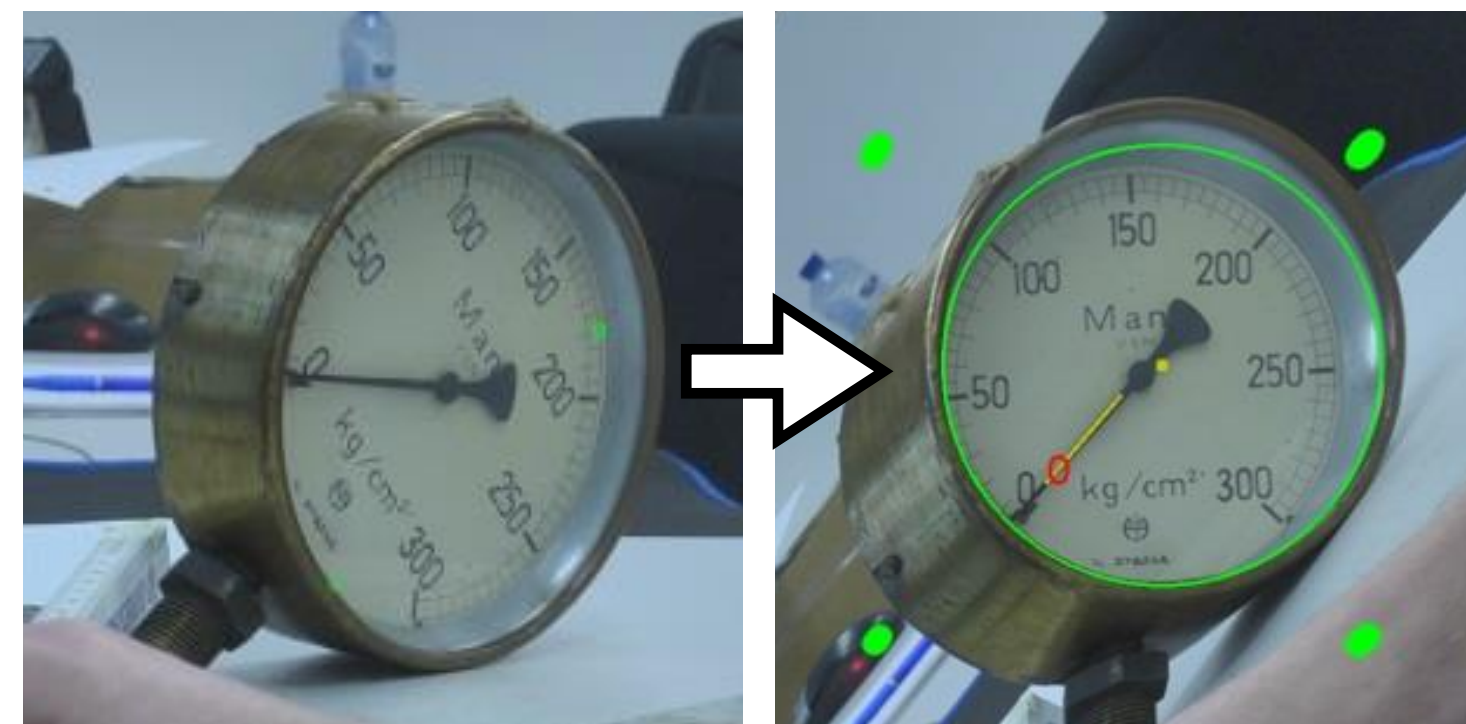
C Visual servoing



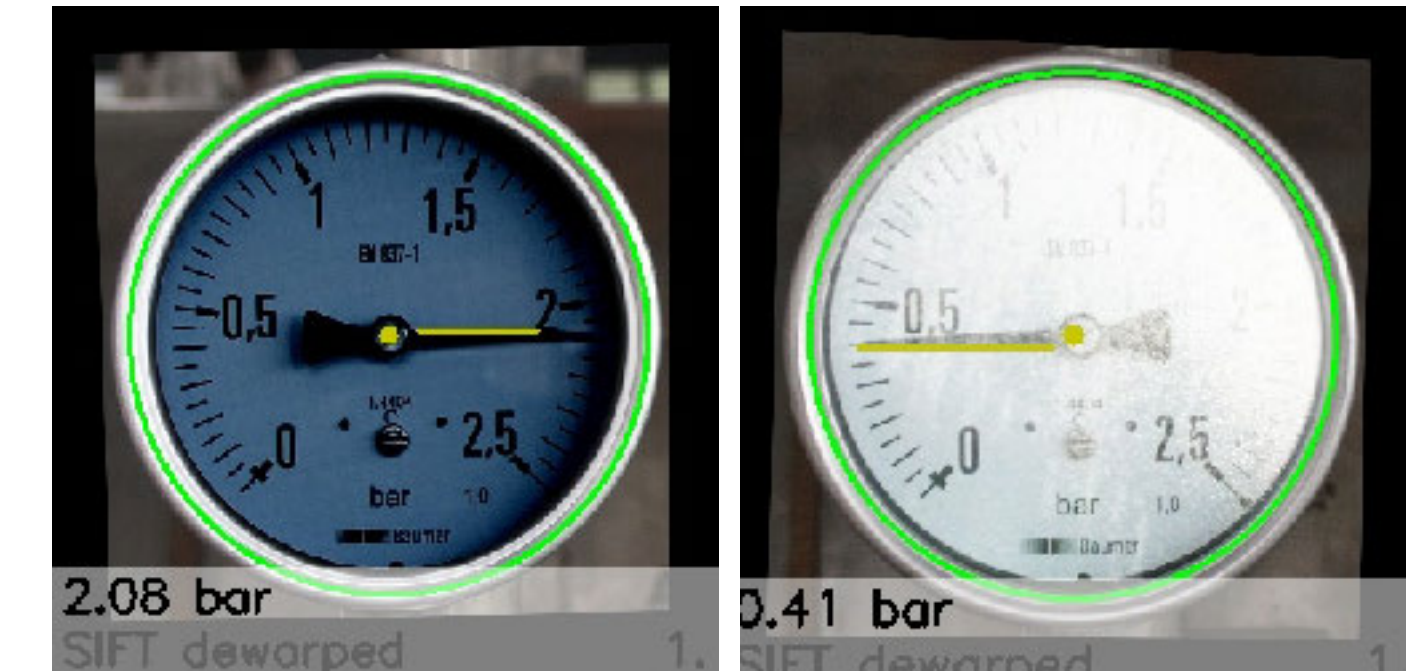
B Whole-body camera positioning



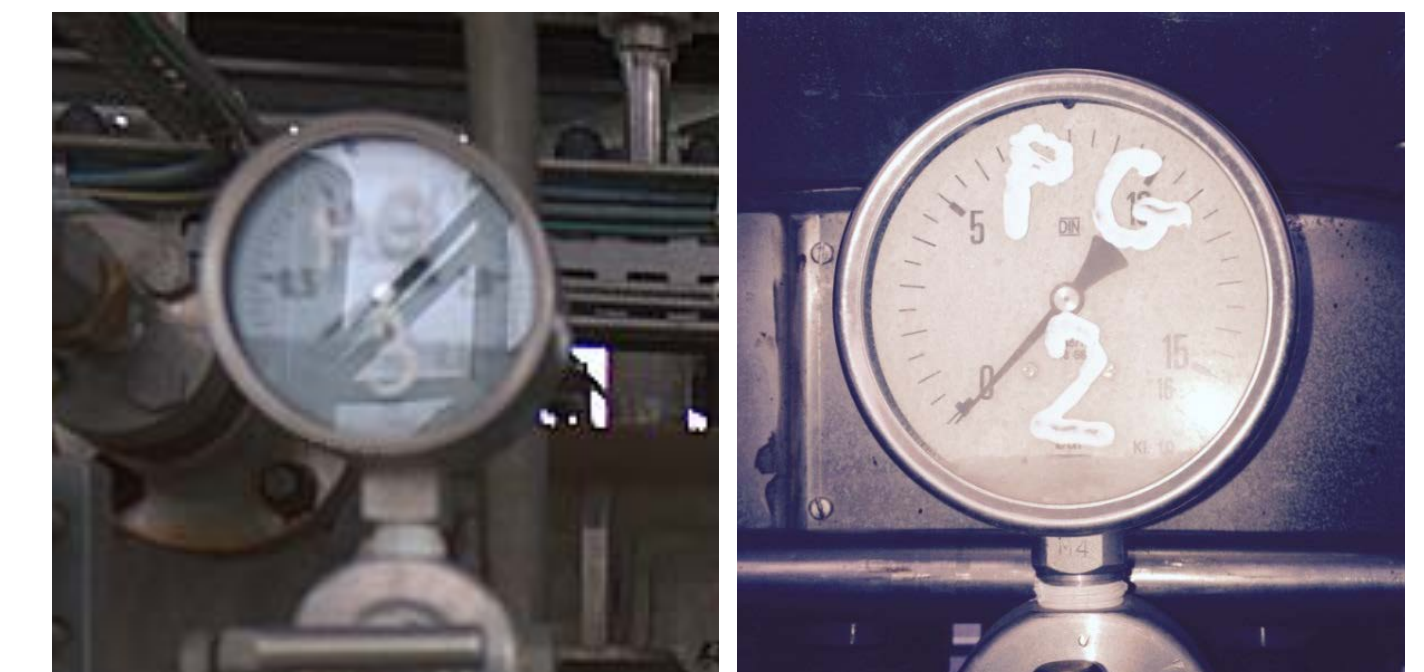
D Image de-warping



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.

The screenshot displays the Argos Mission Control interface. At the top left is the ETH zürich logo. The main window is divided into several sections:

- Top Left:** A camera view labeled "OmnView" showing a real-world scene of a laboratory hallway.
- Top Center:** A 3D simulation of a quadruped robot (JPV) navigating through a white wireframe model of a building's interior. The robot is positioned on a path marked with red and blue lines. A "Reset" button is located below this view.
- Top Right:** A camera view of a blue balloon with a white logo, labeled "Image". Below it are tabs for "Light", "Progress", "Measurements", and "Settings".
- Bottom Left:** Mission control panels including:
 - Start Mission:** A dropdown menu for "ArgosMission" and a "Start" button.
 - Control Mission:** A "Running Mission" dropdown for "ArgosMission", a "Running Task" dropdown for "FollowPath", and a progress bar showing "0.589/5.798 m".
 - Supervisions:** A grid of status indicators for Alarm, Battery, Communication, Dysfunktion, Emergency Stop, Gas Acoustical, Gas Electrochemical, Heat, Localization, and Obstacle.
- Bottom Center:** A "Mission Protocol" table with columns for Stamp, Level, Author, and Designation. The table contains 11 rows of mission events.
- Bottom Right:** A "Quadruped State Visualizer" showing a top-down view of the robot's four legs and joints, labeled "JPV" and "T".

At the bottom of the interface is a navigation bar with tabs for "Argos Mission", "Safety", "ICP", "Pose Controller Manager", "Pose Controller", "Loco", "Robot Monitor", "Console", "VISCA Camera", "Tele-Operation", and "Debug".

Stamp	Level	Author	Designation
1970/01/01 00:00:47.021 UTC	INFO	Mission	The operator started the mission: ArgosMission
1970/01/01 00:00:47.021 UTC	INFO	StartUp	Started.
1970/01/01 00:00:47.021 UTC	INFO	StartUp	Success. (Skipping in simulation.)
1970/01/01 00:00:47.093 UTC	INFO	PlanPathFromHereToInspectionPoint1Aa	Started.
1970/01/01 00:00:47.488 UTC	INFO	PlanPathFromHereToInspectionPoint1Aa	Success.
1970/01/01 00:00:47.495 UTC	INFO	PlanPathFromHereToInspectionPoint1Ab	Started.
1970/01/01 00:00:47.871 UTC	INFO	PlanPathFromHereToInspectionPoint1Ab	Success.
1970/01/01 00:00:47.876 UTC	INFO	PlanPathFromHere	Started.
1970/01/01 00:00:48.116 UTC	INFO	PlanPathFromHere	Success.
1970/01/01 00:00:48.116 UTC	INFO	FollowPath	Started.
1970/01/01 00:00:48.131 UTC	INFO	ObstacleSupervision	Started.

User Interface

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

Situational camera

3D view (RViz)

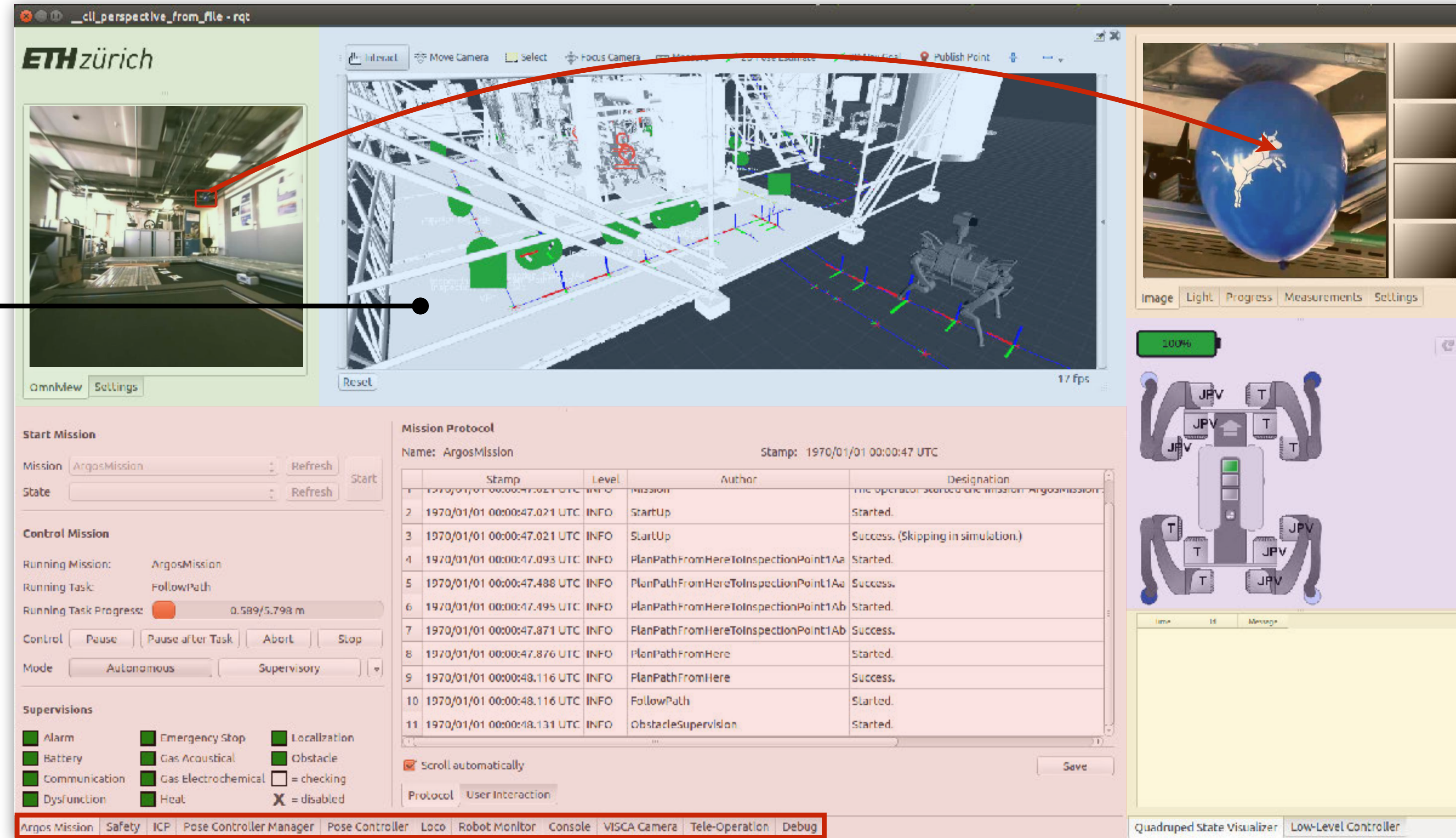
Inspection cameras

Robot actuators & sensors

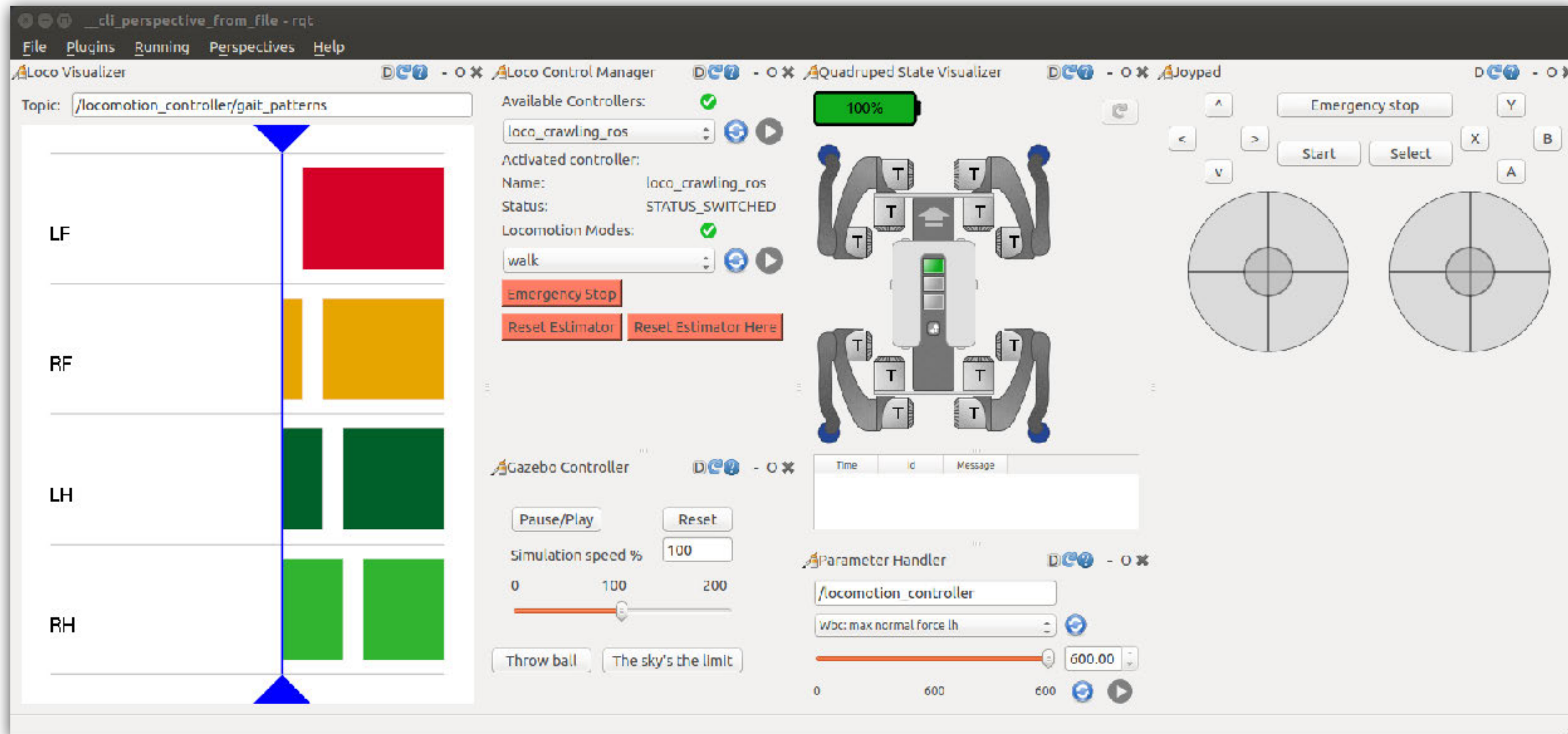
Mission control & protocol

Error protocol

Other modules



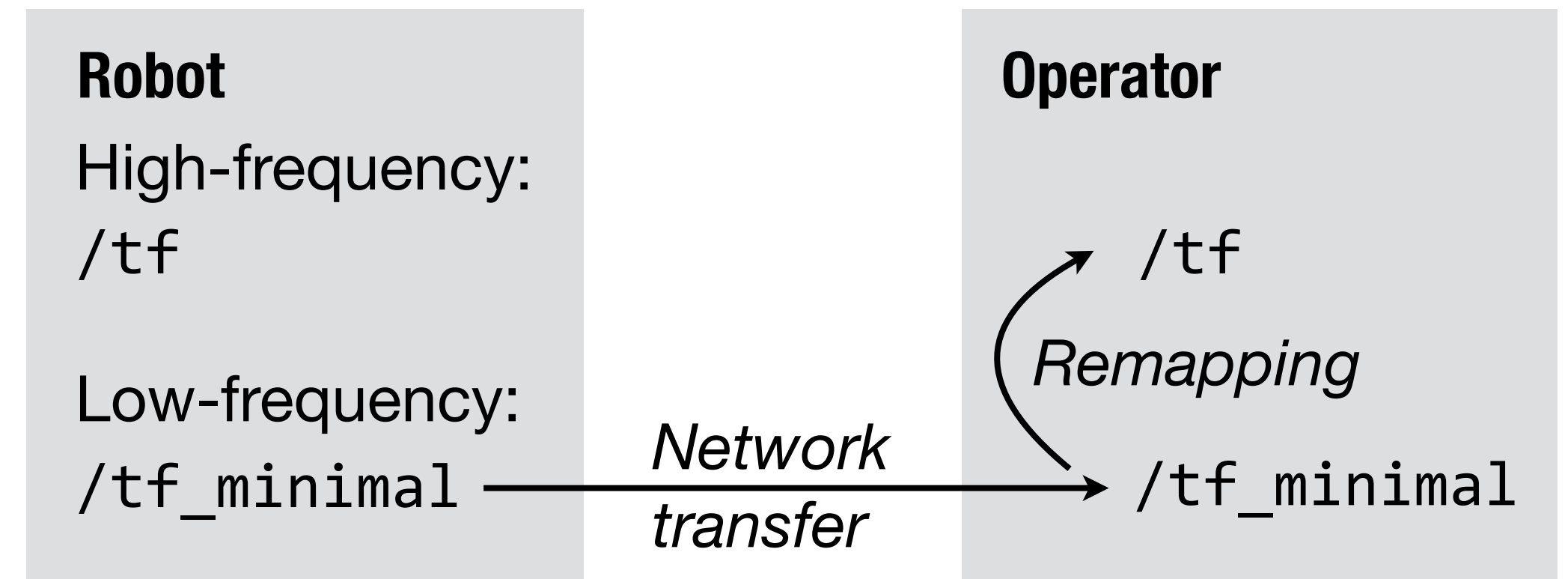
User Interface



User Interface

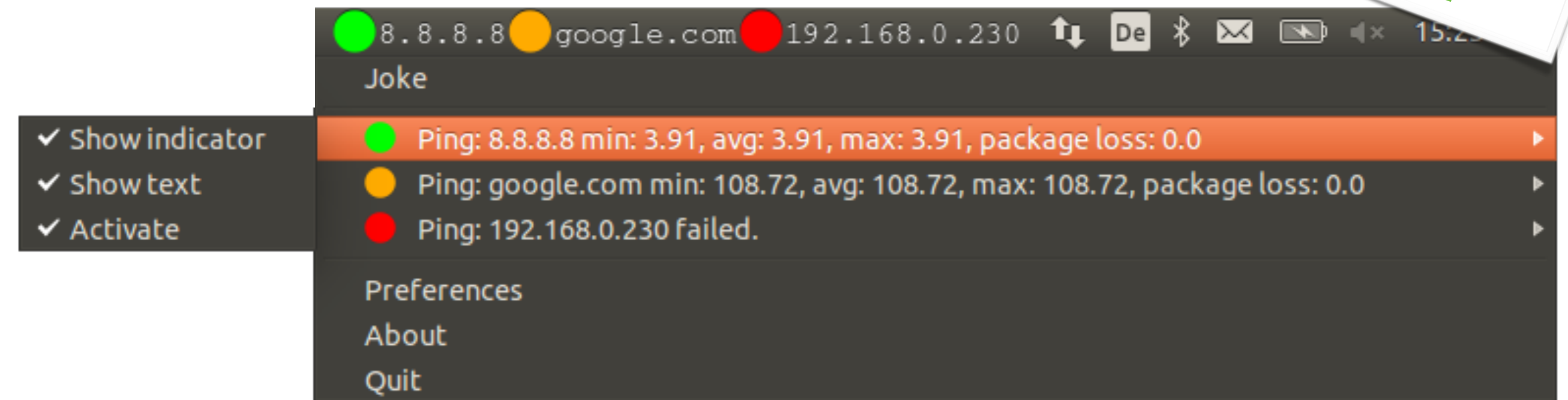
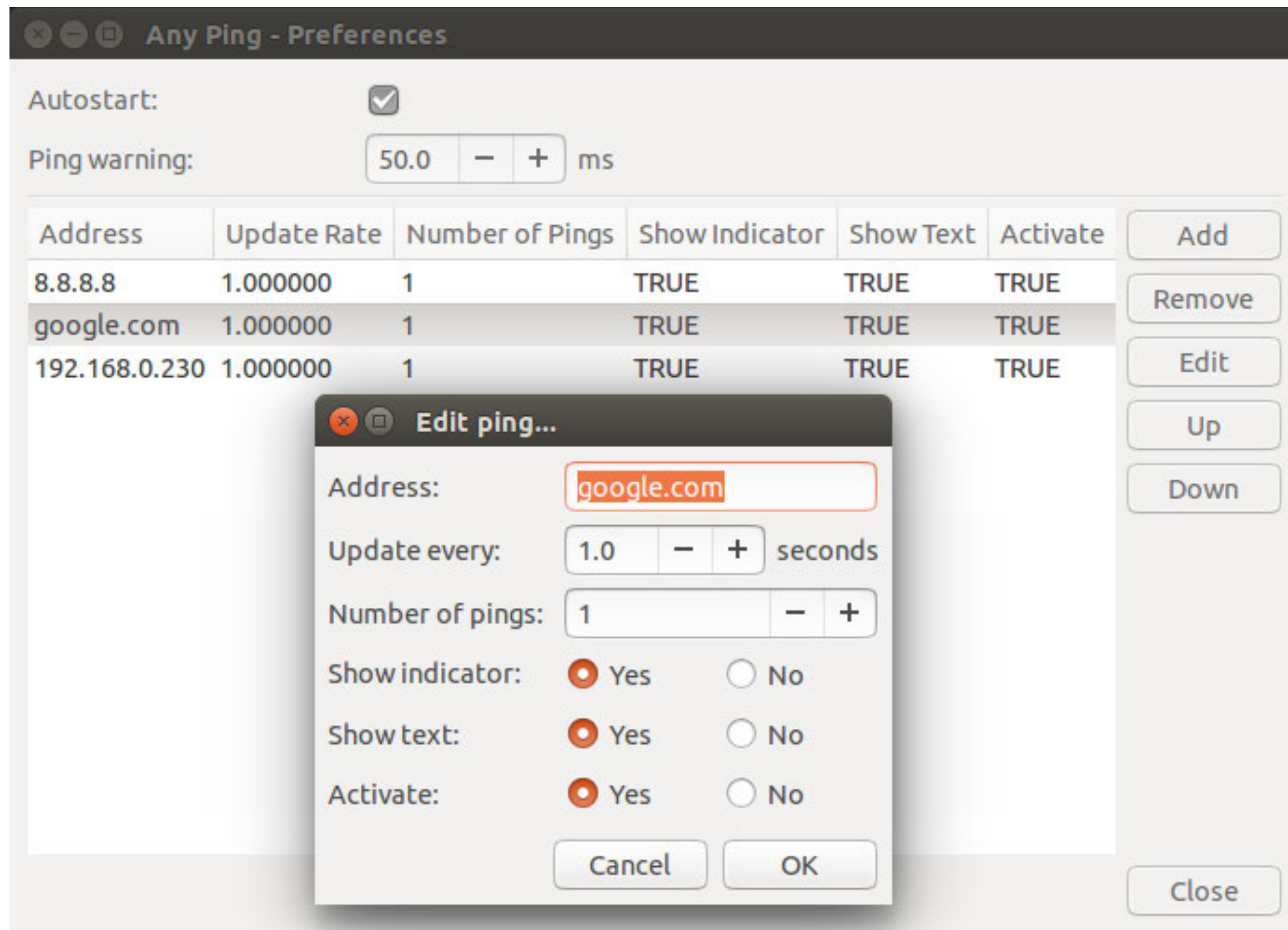
Bandwidth Considerations

- 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

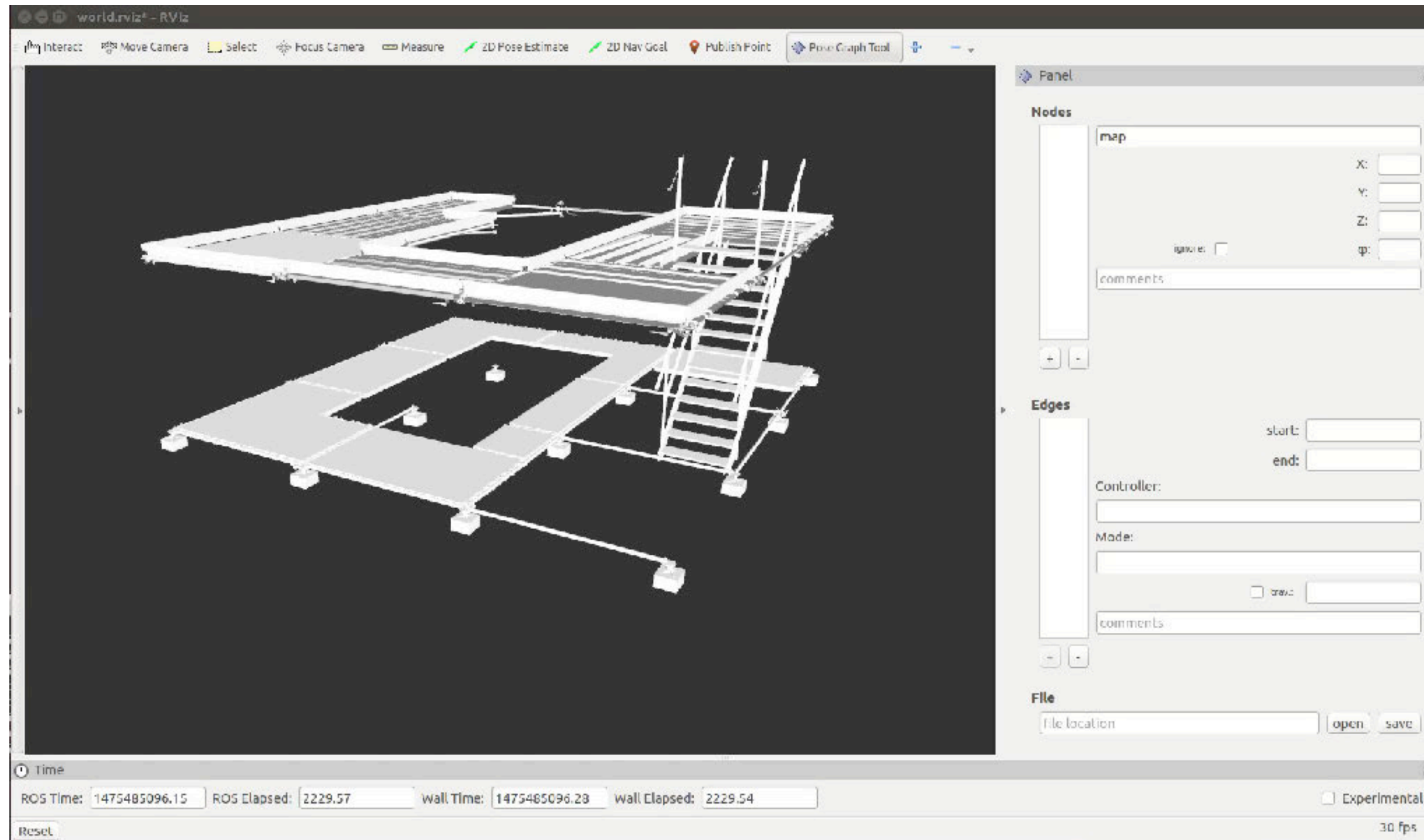
ANYping Indicator



- Indicates PC network availability in Ubuntu menu bar

User Interface

Pose Graph

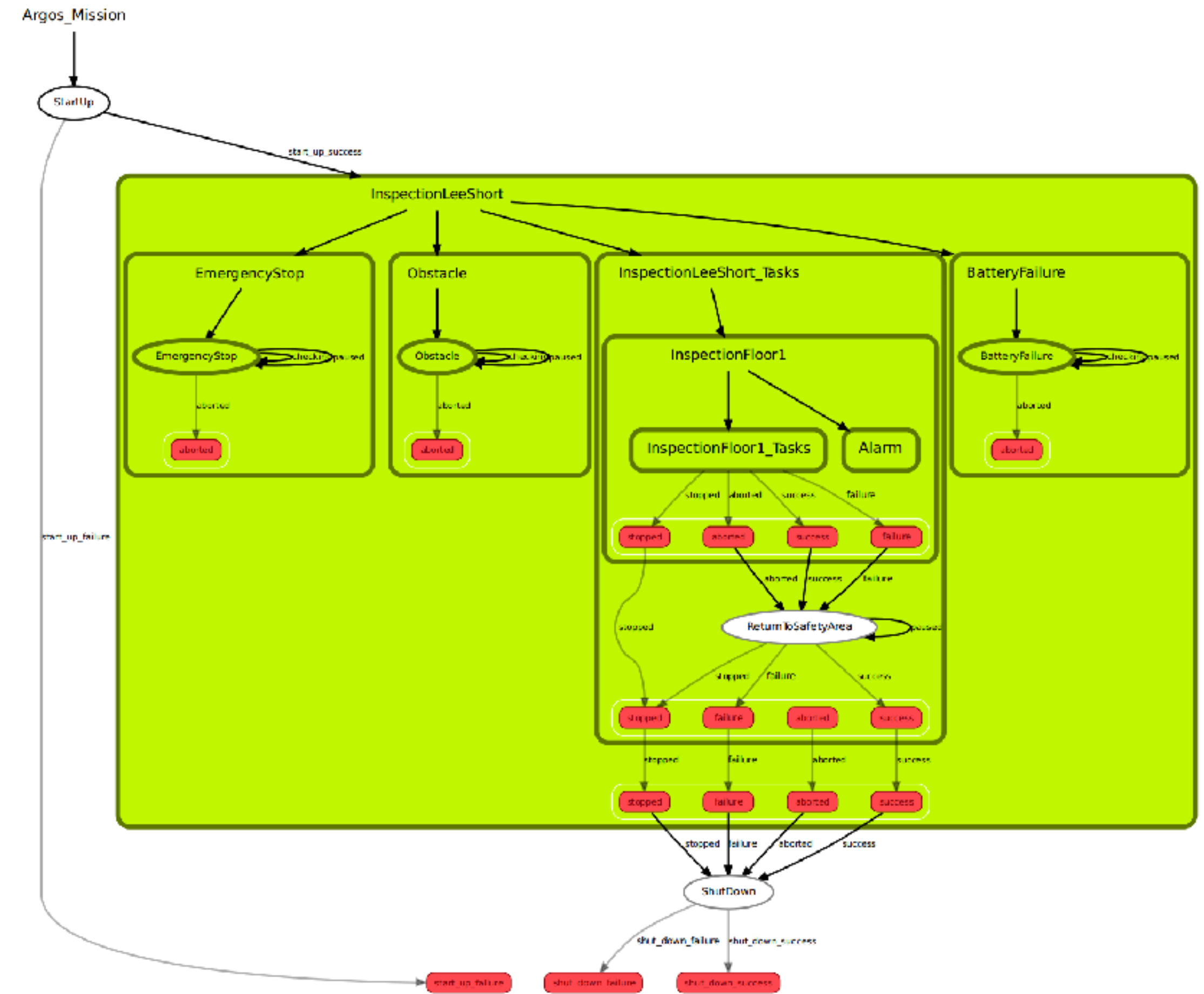


- Pose graph for inspection, special maneuvers (e.g. stairs), docking station etc.
- Visualization and interactive editing of pose graph
- Continuous updating and (re-)planning on pose graph during mission

User Interface

Mission Creation

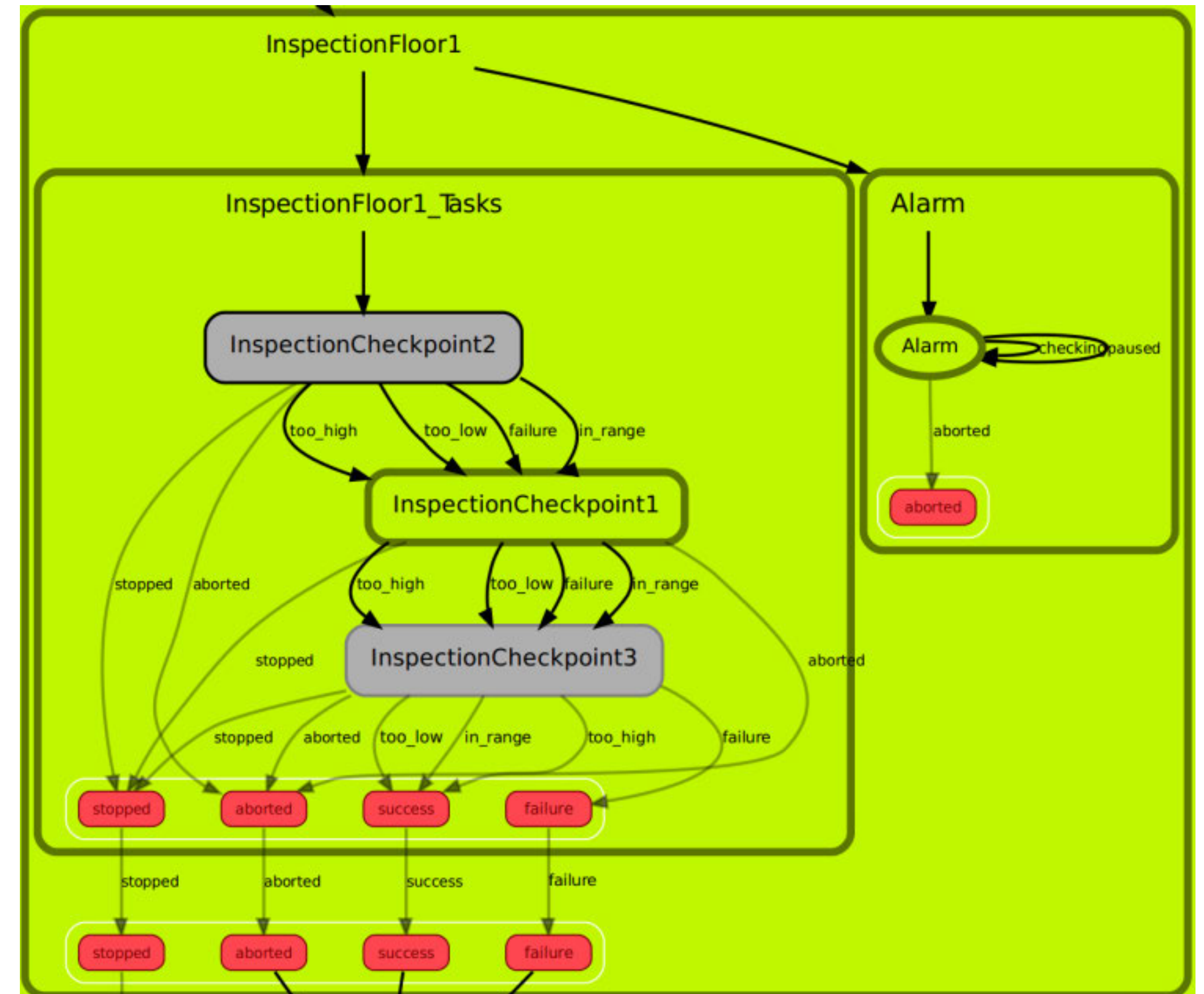
- Task-level state machine (C++ library, similar to SMACH)
- State machine defined in YAML format
- Common building blocks to facilitate construction



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



RQT Multiplot Plugin & Variant Topic Tools



Open Source

[github.com/ethz-asl/
rqt_multiplot_plugin](https://github.com/ethz-asl/rqt_multiplot_plugin)



Open Source

[github.com/ethz-asl/
variant](https://github.com/ethz-asl/variant)



- C++ library (alternative to `rqt_plot`)
- Multiple plots in one window
- Edit, save, and load configurations
- Live plotting or load rosbags

RQT Multiplot Plugin & Variant Topic Tools



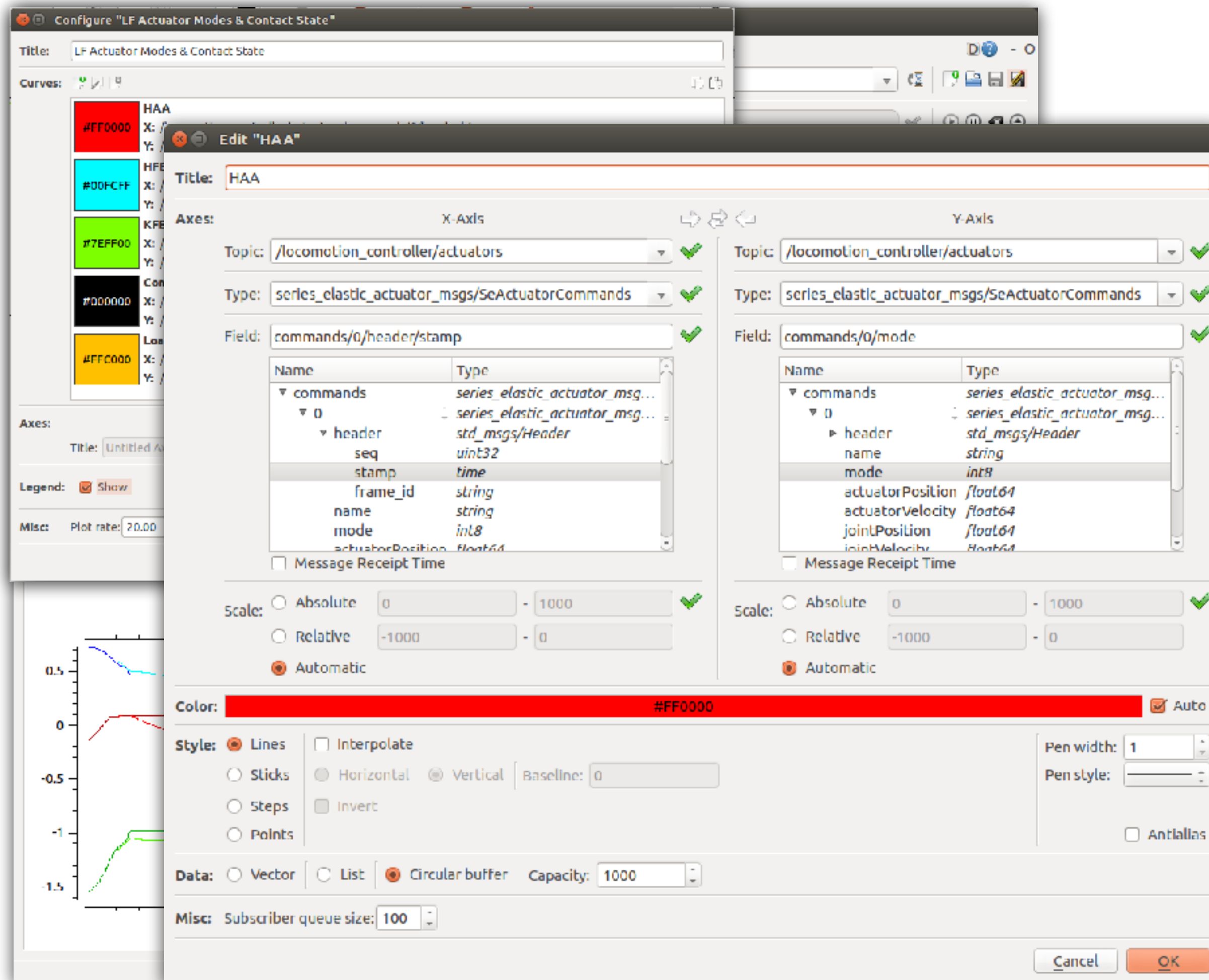
Open Source

[github.com/ethz-asl/
rqt_multiplot_plugin](https://github.com/ethz-asl/rqt_multiplot_plugin)



Open Source

[github.com/ethz-asl/
variant](https://github.com/ethz-asl/variant)



- C++ library (alternative to rqt_plot)
- Multiple plots in one window
- Edit, save, and load configurations
- Live plotting or load rosbags
- Easy to setup configurations

RQT Multiplot Plugin & Variant Topic Tools



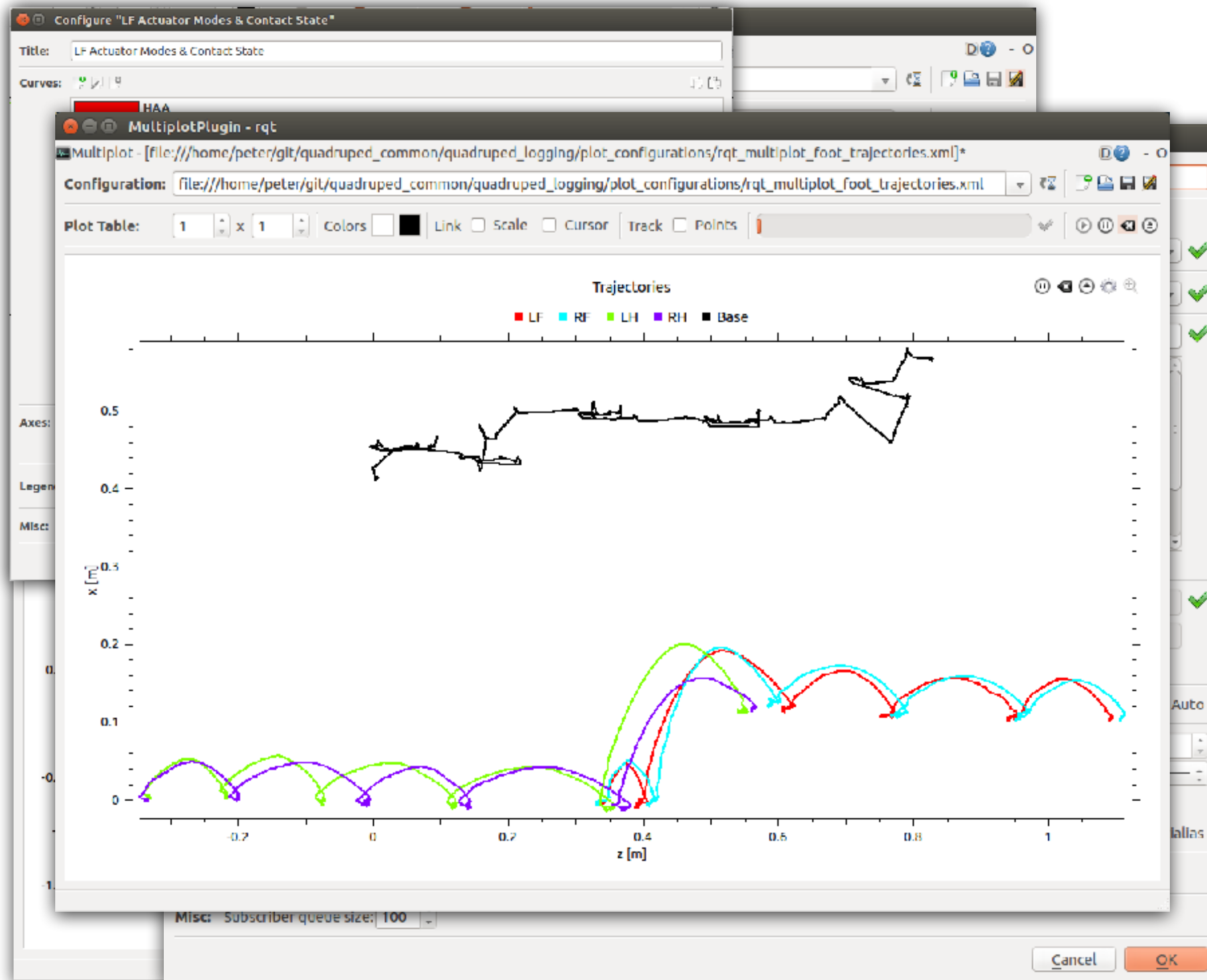
Open Source

[github.com/ethz-asl/
rqt_multiplot_plugin](https://github.com/ethz-asl/rqt_multiplot_plugin)

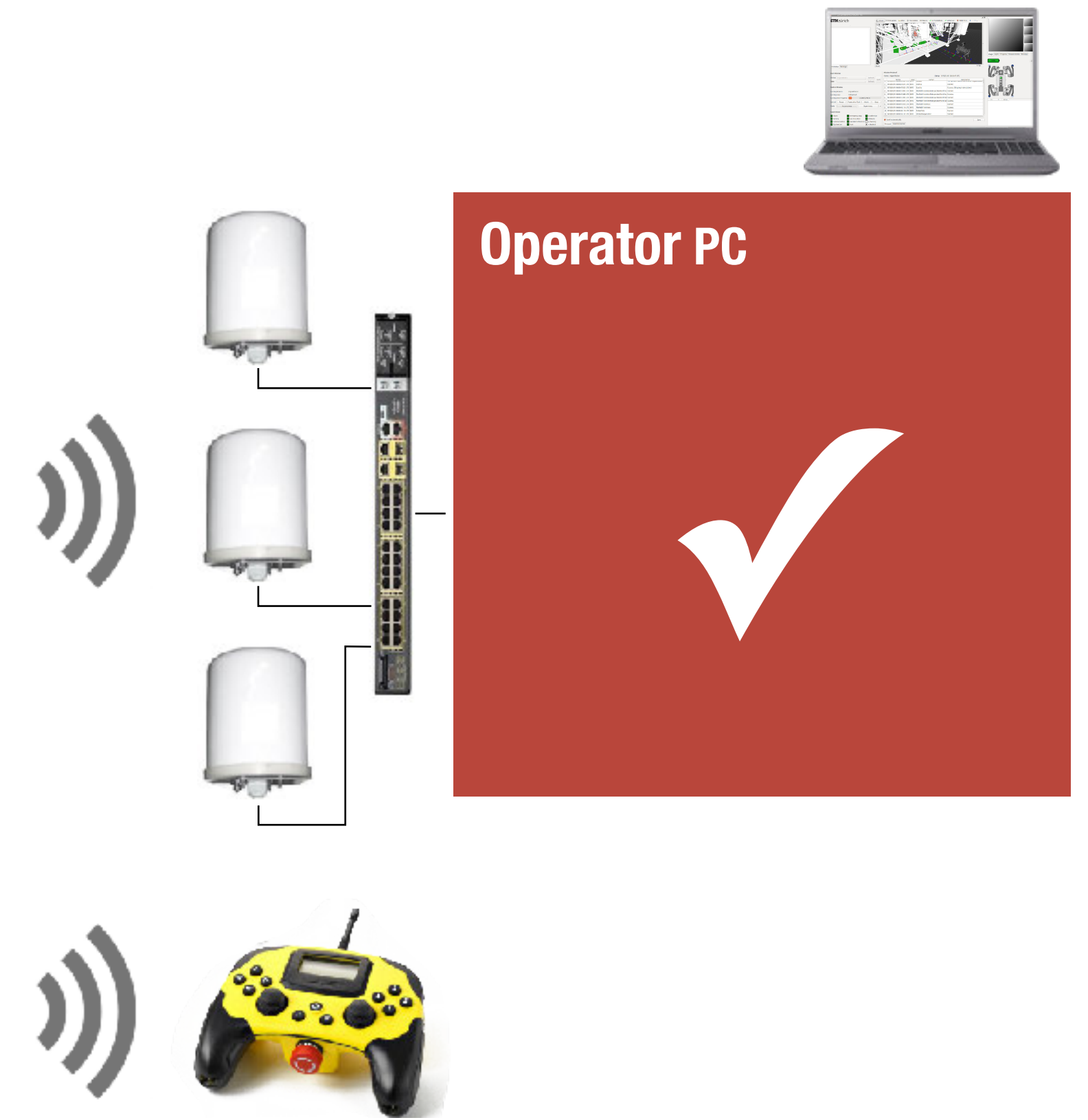
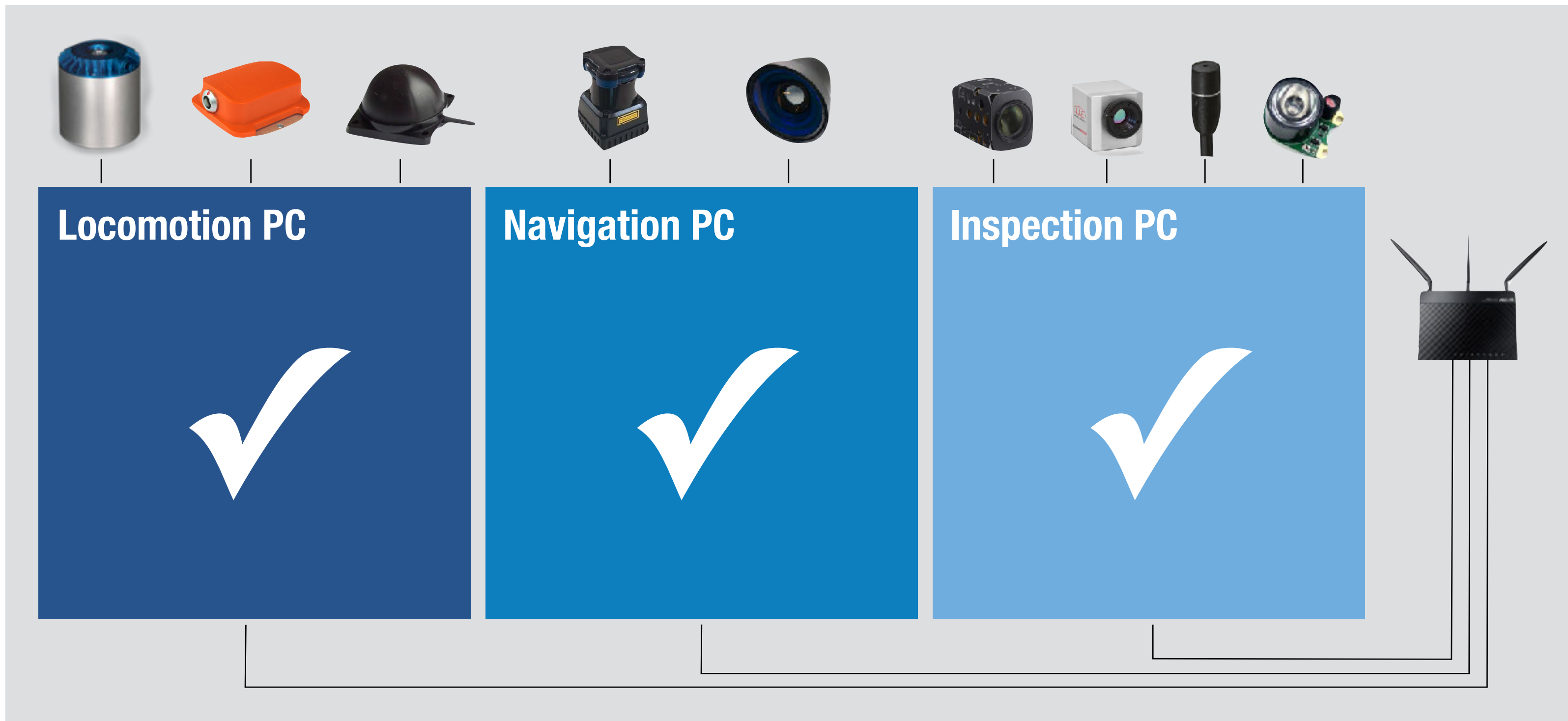


Open Source

[github.com/ethz-asl/
variant](https://github.com/ethz-asl/variant)



- C++ library (alternative to rqt_plot)
- Multiple plots in one window
- Edit, save, and load configurations
- Live plotting or load rosbags
- Easy to setup configurations
- x- and y-axis freely configurable



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

- All developers and robots same setup
 - ➔ Ubuntu 14.04 LTS, ROS Indigo

ubuntu[®]
14.04 LTS



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

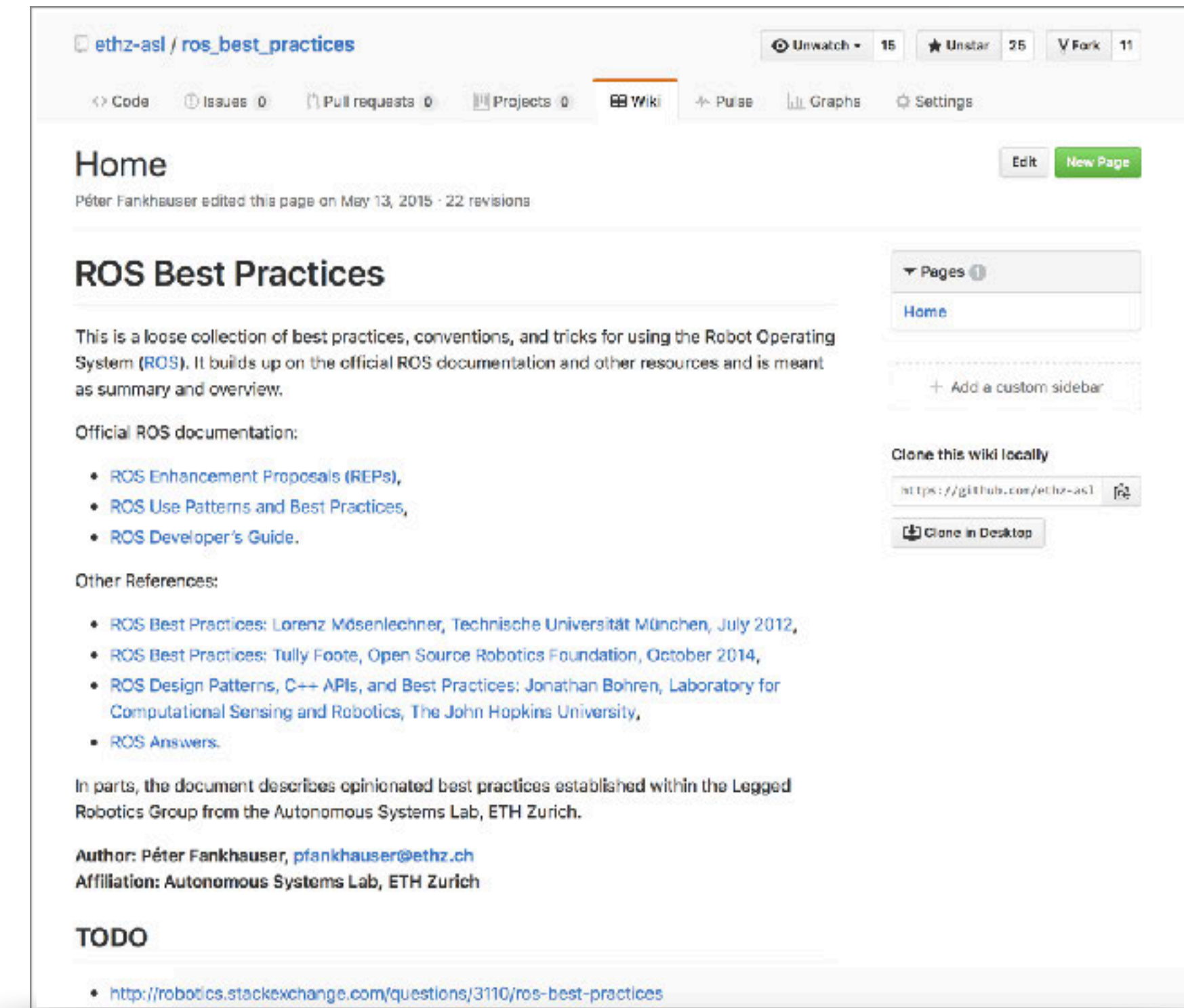
- All developers and robots same setup
 - ➔ Ubuntu 14.04 LTS, ROS Indigo
- Software version control with Git
 - ➔ Bitbucket & GitHub



GitHub

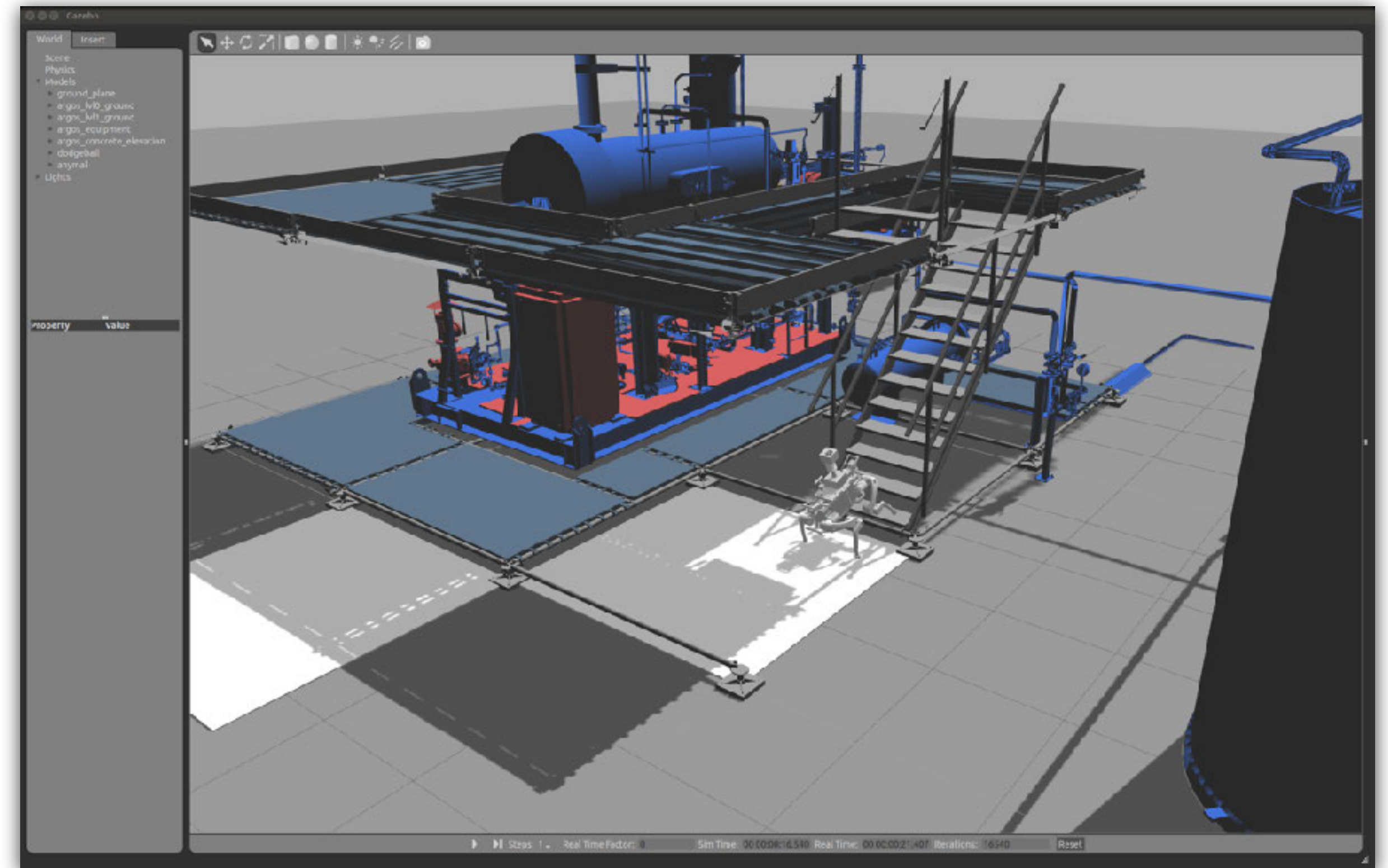
Software Tools – How We (Try) To Keep Things Smooth

- All developers and robots same setup
 - ➔ Ubuntu 14.04 LTS, ROS Indigo
- Software version control with Git
 - ➔ Bitbucket & GitHub
- Conventions for package structure, format, naming, and code style
 - ➔ github.com/ethz-asl/ros_best_practices/wiki



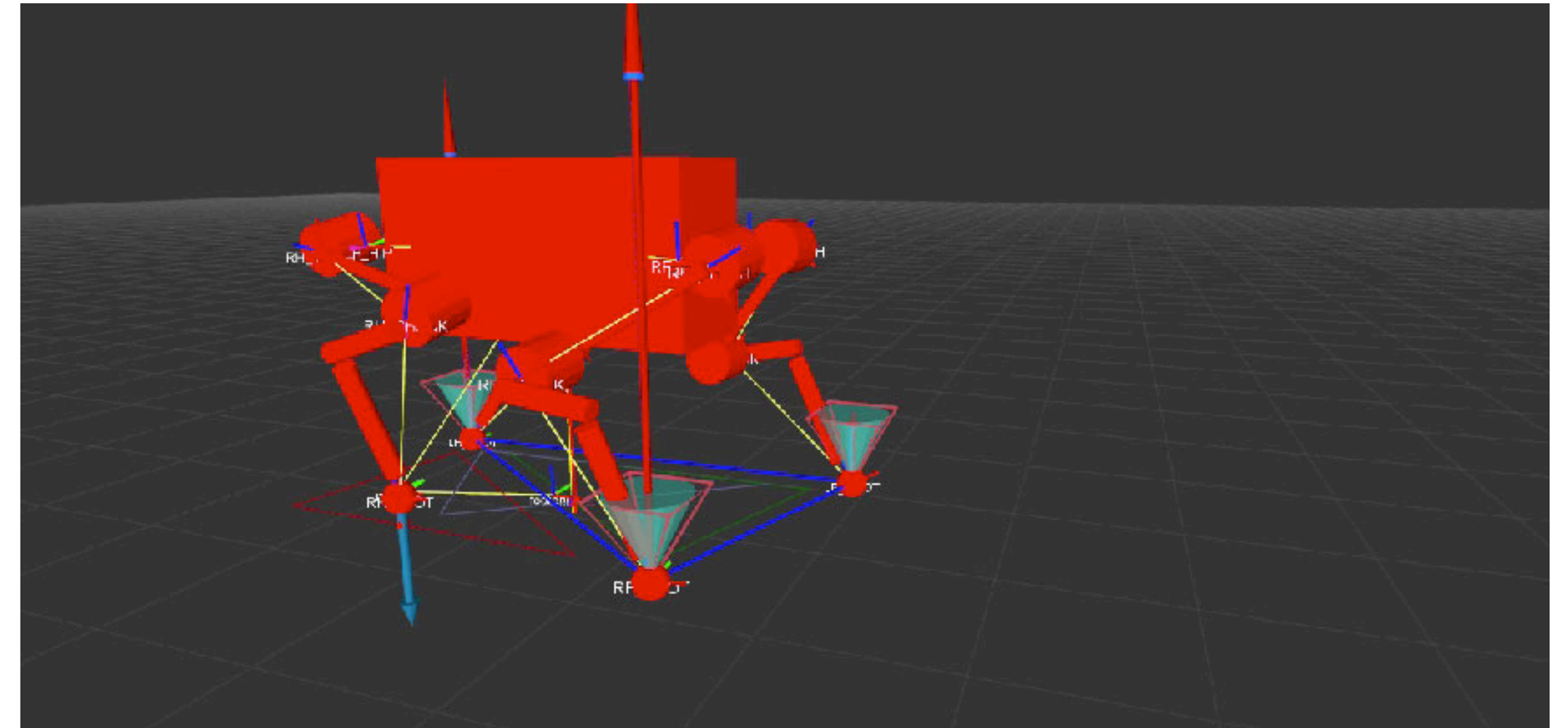
Software Tools – How We (Try) To Keep Things Smooth

- All developers and robots same setup
 - ➔ Ubuntu 14.04 LTS, ROS Indigo
- 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 14.04 LTS, ROS Indigo
- 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



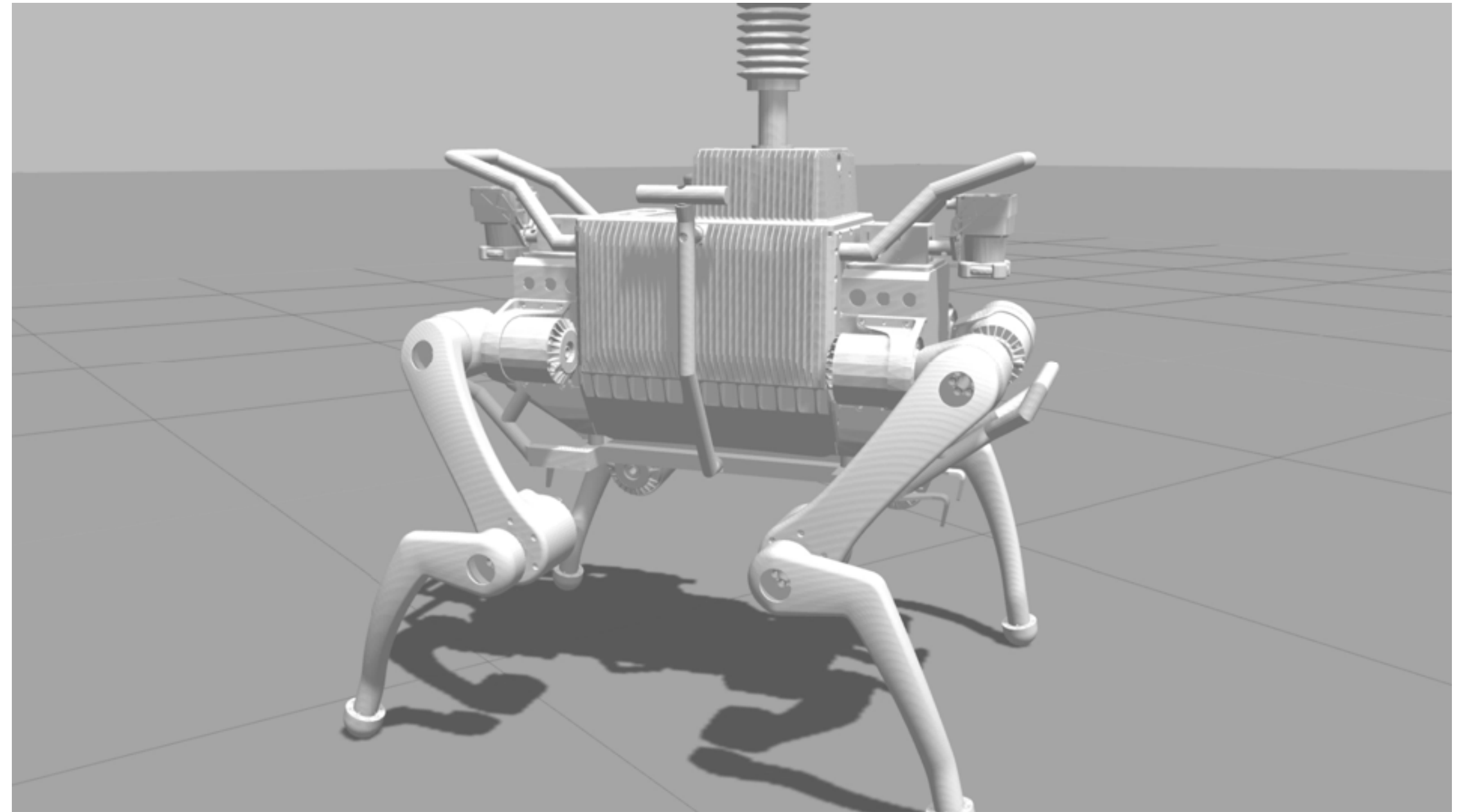
Software Tools – How We (Try) To Keep Things Smooth

- 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)



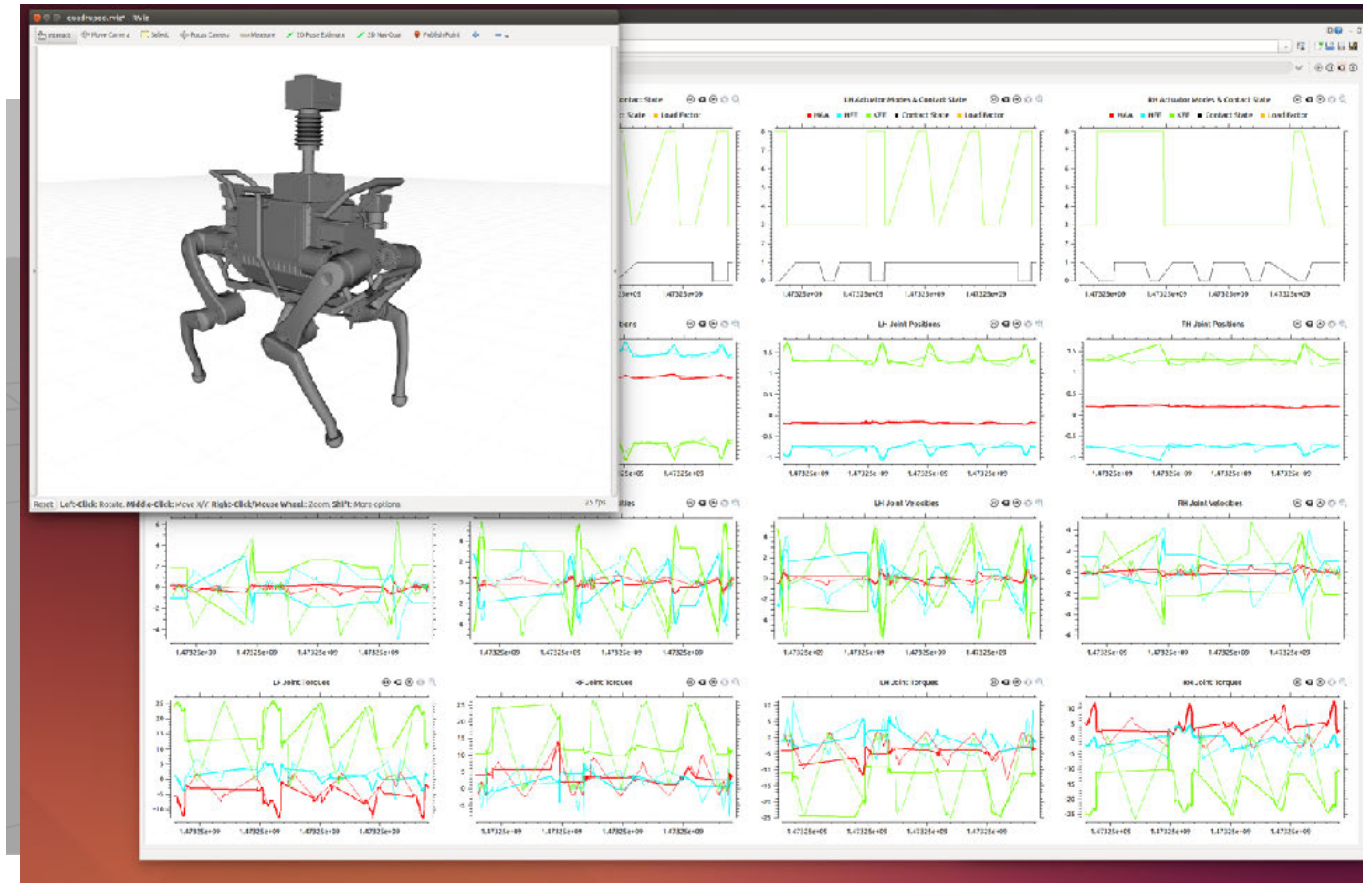
Software Tools – How We (Try) To Keep Things Smooth

- 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 (rosvbag)
 - ➔ All important information is always logged
 - ➔ Review logs with RViz and RQT Multiplot



Conclusion

- Introduced 10 open-source packages, 250+ internal packages
- Coordination of a big team is hard
- Good naming is important
- ROS as “glue”
- WiFi is often problematic
- Reliability is crucial

Thank You



Open-Source Software

github.com/ethz-asl

github.com/leggedrobotics

Péter Fankhauser

pfankhauser@ethz.ch

Samuel Bachmann

Dario Bellicoso

Michael Bloesch

Remo Diethelm

Christian Gehring

Mark Hoepflinger

Gabriel Hottiger

Marco Hutter

Jemin Hwangbo

Linus Isler

Dominic Jud

Ralf Kaestner

Bruno Kaufmann

Philipp Krüsi

Andreas Lauber

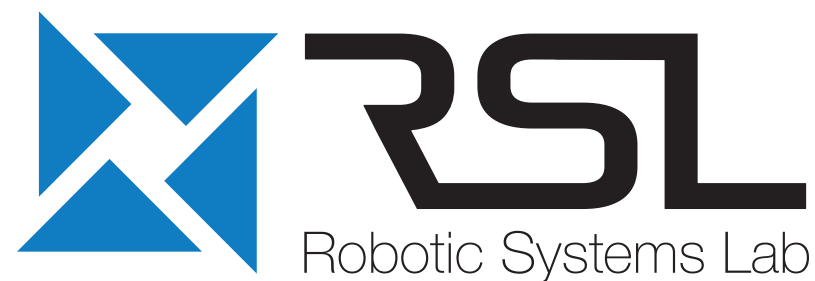
Philipp Leemann

Konrad Meyer

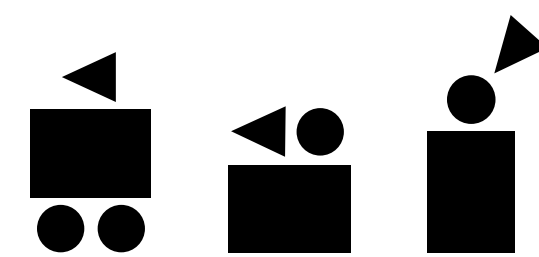
Roland Siegwart

Vassilios Tsounis

Martin Wermelinger



www.rsl.ethz.ch



Autonomous Systems Lab

www.asl.ethz.ch